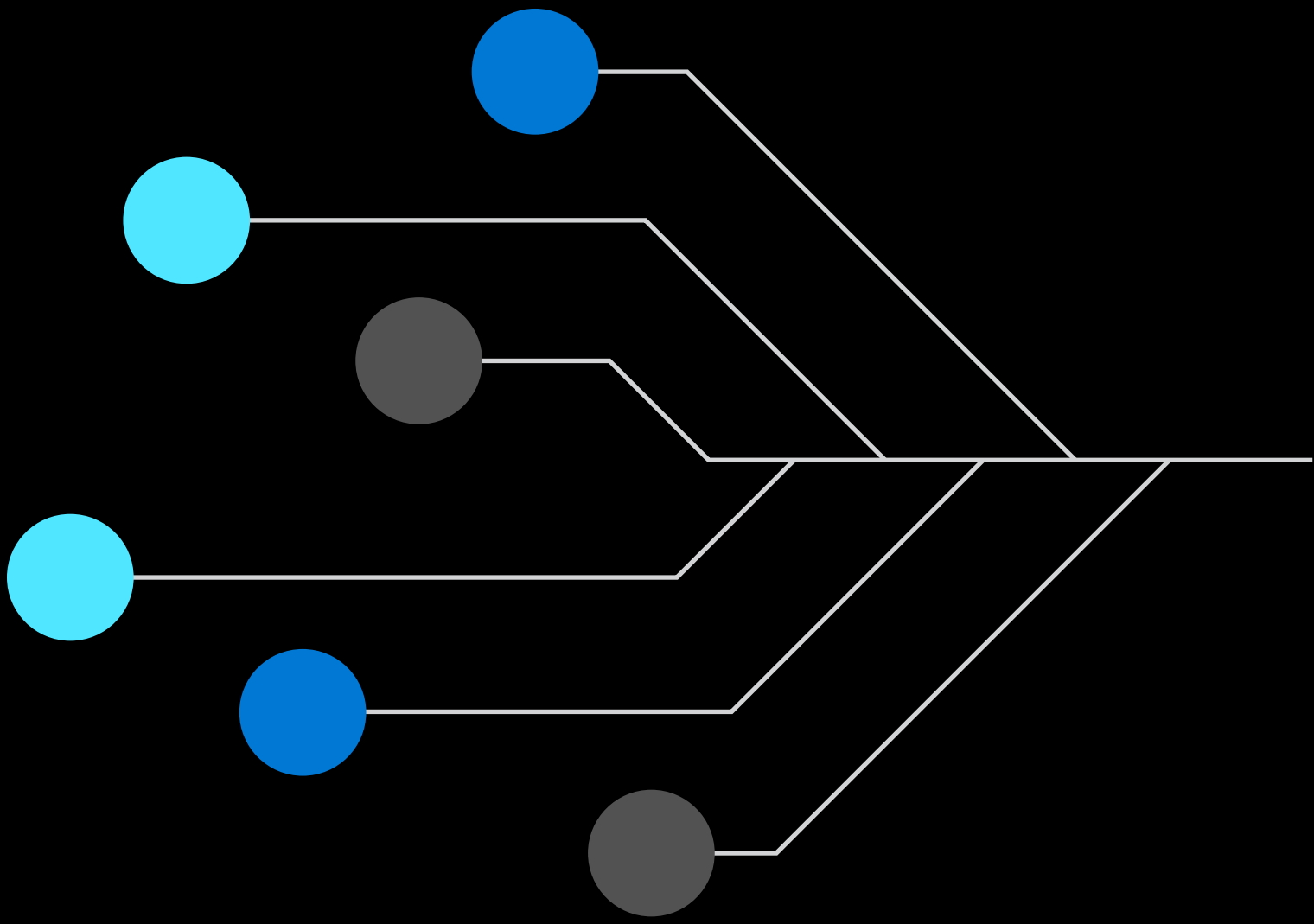
Serie de desarrollo de aplicaciones modernas y Enterprise DevOps

Microsoft sogeti Part of Capgemini

6 consejos para integrar seguridad en DevOps



Resumen

Las prácticas probadas de DevOps ilustran cómo la colaboración entre los equipos de desarrolladores y operaciones conduce a una entrega más rápida de software. El problema que enfrentan los líderes digitales es la seguridad y el cumplimiento de su código, flujos de trabajo e infraestructura. El siguiente paso lógico es integrar su equipo de seguridad con el equipo de DevOps existente, para derribar otro silo organizativo. Lo más desafiante de la adopción de DevSecOps es lograr que la seguridad complemente los procesos empresariales, la cultura y las personas existentes. ¿Cómo pueden los líderes técnicos desarrollar la colaboración entre funciones y unir a los equipos de desarrollo, seguridad y operaciones en torno a la cultura de la seguridad como una responsabilidad compartida?

Los consejos en el interior incluyen cómo:

Desarrollar una cultura empresarial centrada en la seguridad para impulsar la adopción de DevSecOps

Proteger de forma proactiva el código, los flujos de trabajo, la infraestructura y la cadena de suministro de software contra las vulnerabilidades

Proporcionar a sus equipos herramientas compartidas y procedimientos recomendados para habilitar la visibilidad y trazabilidad de extremo a extremo

Aprovechar la información mejorada y la automatización de directivas para lograr un cumplimiento continuo

Lo más desafiante de la adopción de DevSecOps es lograr que la seguridad complemente los procesos empresariales, la cultura y las personas existentes

Índice

6 consejos para integrar seguridad en DevOps

Introducción

A lo largo de los años, las prácticas de desarrollo de software evolucionaron para satisfacer las necesidades y la velocidad del negocio. Recientemente, las metodologías de DevOps brindaron a los ingenieros de software y los equipos de operaciones una forma más rápida y eficaz de desarrollar código. Sin embargo, las prácticas de DevOps eficaces descubrieron un nuevo cuello de botella, lo que puso a la seguridad al final del desarrollo y la administración de aplicaciones. Este cuello de botella es parte del motivo por el que las organizaciones suelen tardar 218 días [1] en descubrir una vulnerabilidad de seguridad, lo que puede ser extremadamente costoso. NIST[2] estimó que el costo de solucionar un defecto de seguridad en la producción puede ser hasta 60 veces mayor que solucionarlo durante el ciclo de desarrollo. Es por eso que la investigación de McKinsey[3] indica que incorporar la seguridad de forma temprana en las etapas de desarrollo y administración de aplicaciones (que se denomina "shifting left" o desplazamiento hacia la izquierda), es un importante enfoque de inversión para los líderes digitales. Estos líderes reconocen que integrar la seguridad en sus canalizaciones y aprovechar las capacidades de plataformas modernas es la siguiente evolución lógica de la metodología de DevOps, DevSecOps.

Ahora, el problema que enfrentan los líderes digitales es la seguridad y el cumplimiento del código, los flujos de trabajo y la infraestructura, todos los cuales se ocupan de la presión externa de los plazos de entrega ajustados. Para cumplir con plazos rígidos, las organizaciones a menudo pasan por alto los procedimientos recomendados de seguridad e implementan código con vulnerabilidades conocidas. El cumplimiento también sigue siendo un problema clave debido a la naturaleza exhaustiva y lenta de las auditorías relacionadas. Forbes informa[4] que "algunos CISO dedican un 30 % o más de su tiempo a lidiar con problemas de cumplimiento". Entonces, ¿cómo puede su empresa mejorar la seguridad y abordar el cumplimiento al mismo tiempo?

Una vez más, la respuesta es la colaboración. Es momento de incluir la seguridad en sus equipos de DevOps. El éxito colaborativo de su equipo de DevSecOps se basa en las herramientas compartidas y la visibilidad del estado de las aplicaciones en cada etapa de la administración y el desarrollo de aplicaciones (ADM). A través de la detección temprana, las organizaciones impulsan correcciones eficaces y rentables de las vulnerabilidades de seguridad. Al mismo tiempo, capturar información de forma obstinada en cada etapa de la ADM permite a las organizaciones lograr un cumplimiento continuo. La tarea más desafiante es lograr que la seguridad complemente los procesos empresariales, la cultura y las personas existentes. Es fundamental para desarrollar la colaboración entre funciones y unir a los equipos de desarrollo, seguridad y operaciones en torno a la cultura de la seguridad como una responsabilidad compartida.

Mejorar la postura frente a la seguridad no se trata solo de migrar la seguridad a una etapa anterior de la ADM. Se trata de adoptar una forma de trabajar diferente, una que enfatice la colaboración entre equipos, la empatía compartida y la responsabilidad compartida. Idealmente, la seguridad se integra en la ADM, por lo que los equipos no lo ven como un paso adicional, sino como un paso integral para la entrega de software. La adopción de DevSecOps requiere que las organizaciones cambien su cultura, evolucionen los procesos existentes, aprovechen las capacidades de plataformas modernas y fortalezcan la gobernanza. Aquí le presentamos seis consejos para usted, un líder técnico, para que integre la seguridad con sus prácticas de DevOps.

El éxito colaborativo de su equipo de DevSecOps se basa en las herramientas compartidas y la visibilidad del estado de las aplicaciones en cada etapa de la administración y el desarrollo de aplicaciones (ADM).

[1]GitHub, Octoverse Security Report, 2020
[2]Security Boulevard, The Importance of Fixing and Finding Vulnerabilities in Development, 2020
[3]Microsoft, Velocidad de desarrollo: Lecciones de los líderes digitales, 2021
[4]Forbes, Awash In Regulations, Companies Struggle With Compliance, 2019

CONSEJO 1

Cree una cultura centrada en la seguridad en toda la empresa

Desarrollar una comunidad de seguridad en su empresa mejora la aceptación en toda la organización y estimula a los empleados

DevSecOps comienza con las personas. Usted no "implementa" DevSecOps; lo acepta. Y para que esto ocurra, su organización tiene que adoptar una cultura de DevSecOps mientras "la vive y la respira", a través de la aceptación de los ejecutivos. Para tener éxito, necesitará a toda la organización, no solo al personal de TI, los equipos de productos y los administradores de proyectos.

La seguridad moderna depende del trabajo en equipo. Un grupo de seguridad por sí solo ya no ofrece suficiente protección para su empresa. Las nuevas amenazas empujan la seguridad a etapas tempranas del ciclo de vida del software para que se acerque más a las aplicaciones. La eficacia depende de la capacidad de los equipos de seguridad, desarrollo y operaciones para trabajar juntos y compartir conocimientos. A menudo, esto comienza con el arquitecto de seguridad trabajando junto a los equipos de DevSecOps, lo que suma principios de seguridad a las primeras etapas de desarrollo. Los desarrolladores también deben expandir su kit de herramientas con conocimiento sobre las operaciones. Las herramientas pueden ayudar, pero la conciencia y la mentalidad son clave. Todo comienza con capacitar a todos a fin de apreciar una verdadera forma de trabajar con DevSecOps.

La cultura es la parte más crítica del proceso de adopción. Por lo tanto, se recomienda que empiece en las personas, migre a los procesos y luego se apoye con la tecnología. Una gran inversión en tecnología fracasa si su personal no tiene interés en la adopción. Requiere un cambio cultural que las personas tengan constantemente una mentalidad enfocada en la seguridad. DevSecOps se basa en un modelo de seguridad compartido, donde los equipos necesitan colaborar. En este modelo, la seguridad no se ve como la responsabilidad de un equipo, sino como un colectivo. Nota: Esto no significa que ya no se necesite personal de seguridad y de infraestructura especializados.

Para comenzar su recorrido de DevSecOps, tendrá que cultivar la idea de que la seguridad es una responsabilidad compartida en toda la organización mediante (1) la capacitación que empodera a sus equipos y (2) una sólida comunidad de InnerSource.

Revitalice sus iniciativas de capacitación

La capacitación es fundamental para que todos en su equipo de DevSecOps comprendan no solo su rol, sino también cómo se cruza con otras responsabilidades en el equipo. A través del intercambio de conocimientos entre los equipos, se espera que todos aumenten su conciencia sobre la seguridad. Recuerde, su programa de DevSecOps demorará en crecer. La adopción generalizada también requiere el respaldo de la administración, para que los equipos no estén excesivamente presionados a medida que se toman tiempo para madurar nuevos procesos y herramientas.

Empiece a crear un equipo con conciencia de seguridad a través de la adopción anticipada del Modelo de campeones de seguridad. El equipo nomina o elige al campeón y se convierte en la voz de seguridad para el equipo. Participan en todas las actividades relacionadas con la seguridad desde el inicio hasta el lanzamiento, pero el campeón de seguridad no es el único responsable de todos los problemas de seguridad en una versión determinada. En cambio, coordinan y rastrean los problemas de seguridad mientras se comunican con las partes interesadas pertinentes. Use el campeón de seguridad para que actúe como un asesor de seguridad in situ que puede prever posibles problemas de seguridad y trabajar en el análisis de riesgos, esbozando los requisitos de seguridad en las primeras etapas de la fase de desarrollo. Estos conocimientos ayudan a crear una base de seguridad para su equipo de DevSecOps. En general, piense en asignar este rol a una de las partes interesadas sénior experimentadas, ya que requiere un nivel óptimo de comunicación, conocimiento práctico de la seguridad y voluntad para expresar con confianza su opinión sobre señales de alerta.

Evalúe la preparación de su organización para un cambio cultural de DevSecOps con los siguientes tipos de preguntas:

¿Las partes interesadas sénior apoyan abierta y explícitamente el uso de métodos Lean, Ágil y de DevOps en el programa?

¿La organización apoya la colaboración directa entre los equipos de desarrollo, pruebas y operaciones?

¿Los líderes de seguridad son conscientes del nuevo rol que desempeñarán sus equipos en el desarrollo y la administración de aplicaciones modernas?

¿La organización proporcionará los entornos físicos y sociales necesarios para el éxito del equipo?

¿La administración entenderá y abogará por el tiempo y esfuerzo adicional para mejorar las técnicas de DevSecOps?

Use el campeón de seguridad para que actúe como un asesor de seguridad in situ que puede prever posibles problemas de seguridad y trabajar en el análisis de riesgos, esbozando los requisitos de seguridad en las primeras etapas de la fase de desarrollo.

Impulse una comunidad sólida de InnerSource

Mantener viva la mentalidad centrada en la seguridad significa trabajar arduamente para crear una comunidad. Simplemente capacitar al personal y comenzar una cultura no es suficiente. El éxito a largo plazo significa cultivar una comunidad vibrante y enérgica de personas.

Una forma de impulsar su comunidad es mediante la adopción de los procedimientos recomendados de InnerSource, donde los equipos comparten y adoptan arquitecturas de referencia listas para usar, código y componentes comunes para facilitar y optimizar sus flujos de trabajo. Esta forma de pensar colaborativa ofrece una mayor velocidad de entrega, una colaboración más fluida entre grupos, un desarrollo de mayor calidad y una mejor documentación. Haga que la comunidad establezca eventos en torno a los patrones de InnerSource y úselos para propiciar una nueva forma de trabajar.

Nombre a un grupo de personas clave para guiar el proceso de incorporación para aquellos interesados en unirse a la comunidad. Estas guías deben describir los procedimientos recomendados de InnerSource y desarrollar contenido educativo útil. Al mismo tiempo, indique a las guías que creen el repositorio de InnerSource, estableciendo una referencia y estándares para que otros los vean y, con el tiempo, contribuyan. Esta ubicación central permite a otros en la comunidad ayudar, ya sea mediante la revisión o la incorporación de comentarios.

Enfoque las iniciativas de evangelización en los eventos de la comunidad en torno a los temas de la biblioteca de InnerSource o a los problemas y soluciones comunes de DevSecOps. La participación es lo que impulsa a DevSecOps, así que mantenga la puerta abierta a todas las personas de su empresa.

Enfoque las iniciativas de evangelización en los eventos de la comunidad en torno a los temas de la biblioteca de InnerSource o a los problemas y soluciones comunes de DevSecOps.

Para ilustrar cómo se ve una cultura centrada en la seguridad en la práctica, echemos un vistazo al caso ficticio de la empresa minorista en línea "Custom City Clothing".

El caso de Custom City Clothing

Uso de un campeón de seguridad para unir a los equipos

Mark abre su almuerzo con la esperanza de que su ensalada le permita olvidarse de la iniciativa de cultura de la empresa en la que está trabajando. Como director general, Mark lidera Custom City Clothing y está entusiasmado con la próxima transformación de DevSecOps. Pero, al mismo tiempo, Mark es consciente de que los equipos de seguridad, aplicaciones y operaciones no interactúan mucho en sus rutinas diarias y se resistirán al cambio. En busca de una forma de avanzar, Mark contactó a Jodie, una desarrolladora de aplicaciones sénior, para hacer una lluvia de ideas sobre algunas soluciones.

A medida que Jodie y Mark hablan sobre algunas ideas, el Modelo de campeones de seguridad inmediatamente llama su atención. Les gusta la idea de un campeón que actúe como asesor de seguridad in situ. Este campeón debe ser alguien capaz de prever posibles problemas de seguridad y trabajar en el análisis de riesgos. Idealmente, en algún momento, este campeón comienza a esbozar los requisitos de seguridad de forma anticipada en la fase de desarrollo.

Jodie y Mark luego presentan esta idea a los equipos de seguridad, desarrollo y operaciones para las nominaciones. Sin embargo, establecieron algunos requisitos para las nominaciones. El primero es que el rol debe ser una parte interesada sénior experimentada, lista para comunicarse con los equipos de manera eficaz. Más aún, quieren a una persona con conocimientos prácticos de seguridad para que no tema expresar su opinión sobre señales de alerta. Después de revisar las nominaciones, seleccionan a Maddie, una desarrolladora con experiencia en ayudar a las iniciativas de seguridad en el pasado. Antes de adoptar este rol, Maddie se reunió con los líderes de seguridad para un curso acelerado de capacitación en seguridad. Este conocimiento adicional le brinda el contexto para estar consciente de la situación cuando se reúna con los equipos.

Maddie se puso a trabajar de inmediato y comenzó a participar en todas las actividades de seguridad, desde el inicio hasta el lanzamiento. Ahora que Maddie es el nexo entre los equipos, articula las actualizaciones de seguridad, los requisitos y las protecciones de cumplimiento necesarias para los equipos de seguridad en función de los requisitos del equipo de operaciones y desarrollo. También ayudó a encabezar el desarrollo de prácticas de InnerSource dentro de la empresa, con la esperanza de crear una colaboración más fluida entre los equipos y aumentar la documentación de arquitecturas de referencia, código de aplicaciones e infraestructura, y componentes comunes. Tener un único punto de contacto para la seguridad ayuda a Mark a obtener aceptación de todos los equipos. Y ahora que la cooperación entre equipos y el intercambio de conocimientos ha comenzado de lleno, Mark se siente más preparado para presentar los cambios en el proceso y la tecnología que completarían la transformación de DevSecOps de Custom City Clothing.

CONSEJO 2

Integre la seguridad en las primeras etapas del ciclo de vida del desarrollo

Pensar en la seguridad por adelantado e incorporar las prácticas de seguridad de forma temprana le ayudará a evitar las vulnerabilidades y los obstáculos comunes

Elegir una solución de seguridad al final del proceso de desarrollo ofrece una protección limitada para sus cadenas de suministro críticas de código, datos y software. En lugar de empujar la seguridad al final del proceso de desarrollo, la idea es integrarla como parte del desarrollo cotidiano. Garantiza que las comprobaciones de seguridad y los procedimientos recomendados se produzcan de forma temprana y en todo el proceso de desarrollo, lo que reduce la probabilidad de enfrentar vulnerabilidades.

Entonces, ¿cómo se desplaza la seguridad a la izquierda?

Hay mucho que tener en cuenta, incluidas las herramientas modernas, la adopción de procedimientos recomendados y la aceptación en toda la organización. Comienza con los desarrolladores, incluidos los análisis de seguridad, como parte de sus flujos de trabajo de CI/CD. Estos análisis de seguridad continuos posicionan a las empresas para proteger las aplicaciones mediante el diseño y minimizar las vulnerabilidades.

De forma más táctica, su empresa puede impulsar un cambio exitoso hacia la izquierda con dos prácticas específicas: (1) pasar del análisis por lotes a evaluaciones continuas y comprobaciones de cumplimiento, y (2) garantizar la calidad del código, la postura de seguridad y el cumplimiento con un análisis estático y dinámico.

Pase del análisis por lotes a evaluaciones continuas y comprobaciones de cumplimiento

Es momento de evolucionar más allá del enfoque por lotes que muchos equipos de seguridad usan en la actualidad para detectar vulnerabilidades. El análisis por lotes no solo requiere la participación humana, sino que también es propenso a errores y no puede funcionar a petición. Además, estas comprobaciones de seguridad se producen independientemente del equipo de desarrolladores, lo que limita el contexto para los profesionales de seguridad. Pueden ocurrir retrasos entre los ciclos de vida del software y los comentarios a menudo llegan tarde.

El camino más rápido hacia una postura de seguridad óptima es a través de evaluaciones continuas y comprobaciones de cumplimiento. Esto ocurre en varios niveles, desde el análisis de código hasta las pruebas unitarias de las funciones de seguridad. Estos son algunos principios fundamentales para realizar evaluaciones continuas de forma exitosa:

Busque vulnerabilidades y exposiciones comunes (CVE): Con frecuencia, las vulnerabilidades de las aplicaciones más comunes analizadas para su corrección son las vulnerabilidades OWASP Top 10. Una herramienta como el panel de cumplimiento normativo de Security Center de Azure analiza su suscripción de Azure en busca de estas vulnerabilidades.

Amplíe sus análisis para el cumplimiento: Las empresas que no analizan el cumplimiento o las amenazas avanzadas quedan expuestas a las amenazas modernas. La respuesta es analizar en busca de vulnerabilidades más completas, incluida la preparación para el cumplimiento, mediante procedimientos recomendados, como el marco de NIST o la comparativa de CIS.

Es momento de evolucionar más allá del enfoque por lotes que muchos equipos de seguridad usan en la actualidad para detectar vulnerabilidades.

Detecte temprano, corrija temprano: Configure los procesos para analizar imágenes de contenedores e infraestructura como archivos de código para CVE antes de que se inicien para garantizar que no se produzcan vulnerabilidades en la producción y la corrección se realice lo antes posible.

Automatice los procesos: Desarrolle la eficiencia dentro de su empresa mediante el cambio a procesos automatizados que encuentren y corrijan los problemas más rápido al realizar análisis cada vez que se produce un nuevo cambio, todo ello sin necesidad de intervención humana.

Mejore la trazabilidad: Asegúrese de que cada paso de la canalización genere datos que más adelante se puedan usar para auditorías o análisis de tendencias.

Administre en función de las métricas: Genere informes que califiquen a la empresa en el CVE total en la infraestructura, el nivel de olor del código y el código duplicado de su empresa.

Use puertas de calidad para garantizar el cumplimiento: Antes de cada lanzamiento, use una puerta de seguridad para medir la calidad del código según los estándares prescritos. Si el código no cumple con los estándares de calidad, detenga la versión para corregir las vulnerabilidades inmediatamente antes de aprobarlas.

El camino más rápido hacia una postura de seguridad óptima es a través de evaluaciones continuas y comprobaciones de cumplimiento.

Si bien la empresa se beneficia en gran medida de los procesos automatizados, las personas que se utilizan para el análisis por lotes pueden sentir que ya no son necesarias en el equipo. Es fundamental articular esto como un movimiento para alejarse de las tareas repetitivas que gastan su tiempo. Piense en cómo puede mejorar la capacidad de estos empleados para ser más eficaces en generar un valor de alto nivel empresarial. La transición a evaluaciones continuas y comprobaciones de cumplimiento también puede causar fricción con los equipos de desarrolladores que ven cómo se expanden sus roles de seguridad de aplicaciones. Disuada la noción de seguridad como una tarea lenta y mejore el compromiso de los desarrolladores al ofrecerles comentarios y recomendaciones de corrección continuas sobre su código y las bibliotecas que usan (obsolescencia, vulnerabilidades, incompatibilidades de licencias de OSS).

Evalúe la calidad del código y fortalezca la seguridad con análisis estáticos y dinámicos continuos

Obtener un conocimiento holístico de la calidad de la canalización de entrega, el código y las aplicaciones de la empresa se denomina "Cualimetría". El uso conjunto de análisis estáticos y dinámicos ayuda a definir la Cualimetría de las canalizaciones de integración continua de su empresa. Observar constantemente los indicadores de calidad ayuda a reducir el riesgo del desfase y la deuda técnica. Además, permite que el equipo de desarrollo tenga una autonomía aún mayor para abordar la calidad del código. Lleve este concepto aún más lejos e implemente un criterio de entrega que limite los errores funcionales y las calificaciones de calidad de código defectuoso. Estos controles de calidad de la canalización de CI mejoran la seguridad de todo el sistema.

En muchas situaciones, las empresas suelen compilar hallazgos de diferentes herramientas para obtener una imagen más precisa de su Cualimetría. Esta imagen incluye no solo las vulnerabilidades basadas en código, sino también las vulnerabilidades relacionadas con las dependencias, los problemas de infraestructura y la evaluación de imágenes. Estas son algunas de las consideraciones que se tienen al compilar herramientas para un conjunto de herramientas de seguridad integral:

Realice análisis de vulnerabilidades a diario: Implemente una herramienta de vulnerabilidad basada en código, como CodeQL, dentro de la canalización diaria para dar al equipo de desarrollo actualizaciones periódicas de las vulnerabilidades conocidas. CodeQL se integra muy fácilmente con GitHub para notificar errores y vulnerabilidades en lenguajes de desarrollo, como Java y JavaScript. Esto reduce el tiempo de corrección del código cuando se realizan modificaciones para calificar más rápido las evoluciones técnicas.

Incorpore un comprobador de dependencias: También le recomendamos usar un comprobador eficaz de dependencias dentro de su canalización de CI para analizar ambas dependencias que se extraen de la CI y los activos en el repositorio. Además, puede configurar un firewall para bloquear la canalización en función de los objetivos de nivel de seguridad, personalizados para cada aplicación.

Busque sus imágenes de contenedor: Para proteger la implementación segura en contenedores, es necesario analizar el contenedor para detectar vulnerabilidades de imagen.

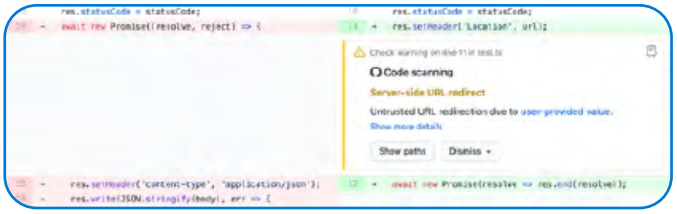
Recuerde analizar la infraestructura: Las pruebas de seguridad de aplicaciones estáticas (SAST) no solo se tratan de identificar las amenazas de seguridad de las aplicaciones, sino también problemas de infraestructura. Si su empresa ya está equipada con infraestructura como código (IaC), entonces se controla fácilmente con una lista conocida de problemas y patrones. Azure Resource Manager y otras herramientas de IaC ofrecen capacidades de "linting" para analizar mejor el código de infraestructura incluido en la CI.

Cree un análisis exhaustivo: Tenga en cuenta que estas herramientas se deben usar en conjunto para generar un informe completo de vulnerabilidades.

Observar constantemente los indicadores de calidad ayuda a reducir el riesgo del desfase y la deuda técnica.

Actualmente, muchas organizaciones solo se basan en el análisis dinámico de código, que proporciona una vista centrada en las aplicaciones mediante técnicas de "caja negra" para evaluar el código. Estos análisis no pueden ofrecer la vista holística que se necesita para entender con precisión el estado actual de las canalizaciones de entrega de la empresa. Es importante complementar el análisis dinámico con el análisis estático del código en las canalizaciones de entrega para que los equipos puedan conocer los problemas potenciales lo antes posible. El análisis de código estático le permite medir la calidad del código a través de indicadores, como errores, vulnerabilidades, deuda técnica, cobertura de pruebas unitarias o duplicación de código. También ayuda a localizar fallas de seguridad a través del reconocimiento de CVE conocidas en dependencias determinadas.

Lamentablemente, el análisis estático es una fuente de "falsos positivos". Por lo tanto, una vez que realice las pruebas de seguridad de aplicaciones estáticas (SAST) en las aplicaciones y la infraestructura, es importante continuar con pruebas de seguridad de aplicaciones dinámicas (DAST) para corregir los falsos positivos. Desde su canalización de CD, implemente un entorno temporal y empiece a comprobar de forma dinámica las infracciones de seguridad en este entorno temporal. A continuación, puede integrar una solución de evaluación de vulnerabilidad diseñada para esta tarea, como Azure Defender. Cuando se realiza en un entorno temporal, disminuye el riesgo de amenazas contra los datos y los sistemas críticos, a la vez que evita una nueva implementación potencialmente peligrosa.



Esta figura muestra el análisis de código de GitHub en contexto donde alerta a los usuarios sobre un redireccionamiento de URL que no es de confianza.

Lo más importante es que los análisis de DAST y SAST no se pueden realizar solos o solo una vez. Deben ejecutarse de forma continua. El análisis regular y constante permite que las actualizaciones se aborden a medida que aparecen, en lugar de hacerlo durante las revisiones extensas que se realizan antes de cada versión de producción. De esta manera, estas evoluciones técnicas pueden beneficiarse de las etapas de calificación funcional para garantizar que no se produzca una regresión técnica. Esta combinación de pruebas continuas, SAST y DAST le permite a su empresa mejorar la calidad del código lo antes posible mientras bloquea las vulnerabilidades en la infraestructura, las imágenes y las aplicaciones.

El análisis regular y constante permite que las actualizaciones se aborden a medida que aparecen, en lugar de hacerlo durante las revisiones extensas que se realizan antes de cada versión de producción.

Para ilustrar cómo se ve en la práctica promover el análisis holístico de amenazas de vulnerabilidad, echemos un vistazo al caso ficticio de la empresa de servicios financieros "King Banking".

El caso de King Banking

Expandir los análisis de seguridad para el cumplimiento

Angela, la directora general de King Banking, se sentó en su escritorio con una taza de té matcha para comenzar el día. Está igual de entusiasmada y nerviosa por el próximo lanzamiento de la aplicación de finanzas personales de King Banking. Por un lado, está muy segura de que a los usuarios les encantará el diseño y la funcionalidad de la aplicación. Pero, al mismo tiempo, no está segura de que la aplicación se lanzará con los controles suficientes de seguridad y cumplimiento. Esta preocupación se debe a que King Banking tiene su sede en Alemania y está sujeto a estrictos requisitos de cumplimiento del RGPD.

Con el lanzamiento en solo 9 meses más, Angela programó una reunión con el director de seguridad de la información de King Banking, Jonas, para desarrollar un plan para calmar sus preocupaciones sobre la seguridad y el cumplimiento. Jonas sabía que tenía que reunir a los líderes de los equipos de desarrollo y seguridad para formar un equipo integral de DevSecOps. Primero invitó a uno de los líderes del equipo de seguridad, Johan, que realiza análisis de seguridad de aplicaciones. Del equipo de aplicaciones, Jonas seleccionó a Mary para que le otorgara una visión general sobre las canalizaciones de entrega actuales.

El recién formado equipo de DevSecOps de King Banking se puso a trabajar rápidamente. Johan sugirió que pasaran de realizar análisis por lotes a llevar a cabo evaluaciones continuas y comprobaciones de cumplimiento. Con el análisis automatizado, Mary sabía que el equipo necesitaba una herramienta para encontrar CVE rápidamente. Sabía que Secure DevOps Kit for Azure analiza automáticamente para buscar CVE, como las vulnerabilidades OWASP Top 10. Jonas también quería obtener información sobre el cumplimiento e insistió en que ampliaran los análisis con los procedimientos recomendados, como el marco de NIST o la comparativa de CIS. Aunque se impresionó con las sugerencias, Johan expresó la necesidad de administrar de manera eficaz los nuevos datos que se generarían al ampliar los análisis. Señaló que debían usar puertas de calidad para garantizar los estándares de cumplimiento antes de cada lanzamiento. Si el código no cumple con los estándares de calidad de King Banking, su equipo puede detener el lanzamiento para corregirlo antes de su aprobación. Satisfecha con los pasos descritos, Angela dispuso que el equipo de DevSecOps ejecutara estas sugerencias y preparara la aplicación de finanzas personales para su lanzamiento.

Han pasado tres meses desde que King Banking lanzó su aplicación de finanzas personales con un tremendo éxito. El enfoque de evaluación continua y comprobaciones de cumplimiento de King Banking evitó que fracasara el lanzamiento de su producto. Y ahora, con cada versión, su equipo de DevSecOps pone en marcha procesos para analizar el cumplimiento y las vulnerabilidades antes de que se apruebe cada versión del código. Con su compromiso con el cumplimiento y la seguridad de los datos, el equipo de DevSecOps dejó contenta tanto a Angela como a los reguladores del RGPD. Angela ahora espera con ansias el lanzamiento de características de nuevos productos en lugar de preocuparse por la seguridad de las aplicaciones y la pérdida de datos de los clientes.

CONSEJO 3

Supervise y observe continuamente con propósito

Planificar los objetivos para realizar una supervisión y observación continuas basadas en el contexto permite que su empresa sea más proactiva contra las amenazas

Con demasiada frecuencia, las empresas aprovechan una solución de supervisión u observabilidad sin adaptarla para que funcione en su organización. Cuando las empresas no planifican por completo su iniciativa de supervisión, se sobrecargan con datos. Puede resultar como buscar una aguja digital en un pajar. Además, sin reunir los datos adecuados de la manera correcta, a menudo no son prácticos para su empresa. El primer paso para permitir una supervisión continua (y el crecimiento del subconjunto anteriormente limitado de inteligencia) es la planificación estratégica.

¿Cómo se ve la supervisión continua con propósito y éxito?

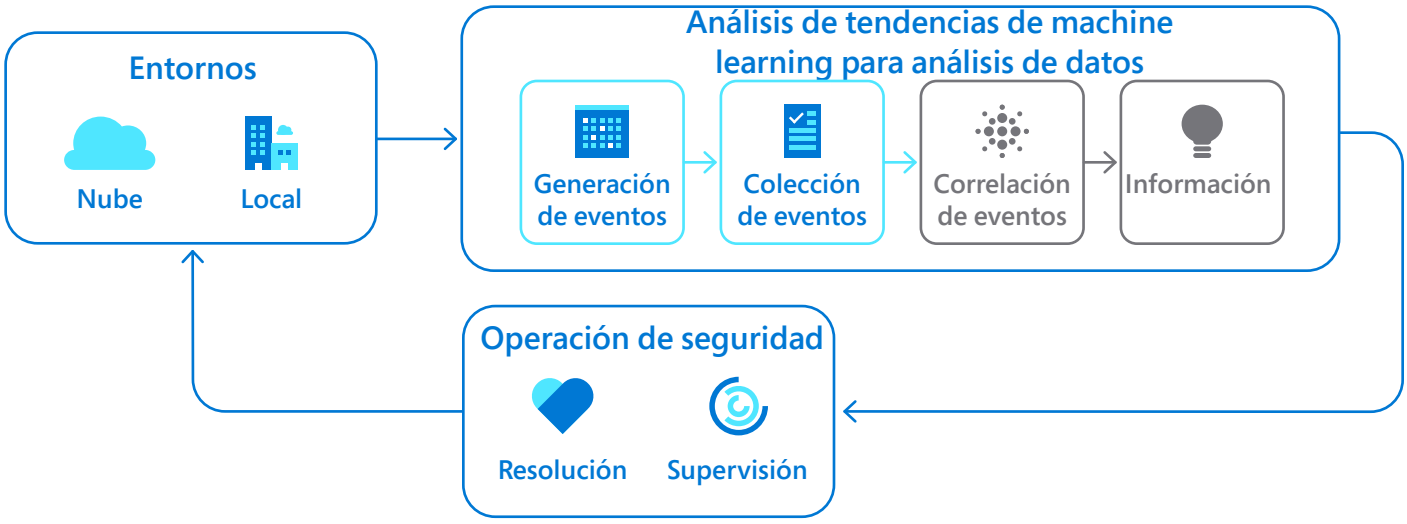
Las prácticas de supervisión exitosas se basan en cuatro aspectos: (1) reunir datos holísticos que ofrecen un panorama completo, (2) estructurar los datos para el análisis, (3) usar alertas prácticas e inteligencia contra amenazas para reaccionar de forma proactiva a las amenazas más rápido e (4) incorporar una cadena sólida de herramientas de supervisión creada para las amenazas modernas.

Cree una imagen completa con datos estructurados

La supervisión proporciona el mayor valor cuando observa todo lo que ocurre dentro de su empresa, ya sea active directory, firewall, syslog, registro de aplicaciones, etc. Recopilar datos de una selección incompleta de fuentes no le permite ver por completo su negocio, así que no olvide recopilar datos de la infraestructura que provienen desde fuera del proceso de administración de cambios.

Las organizaciones exitosas no solo capturan todos sus datos; los organizan con cuidado. Tendrá que decidir qué registrar en función de los objetivos potenciales. Considere la proporción de "falsos positivos" al recopilar datos de diversas fuentes. Por ejemplo, el syslog de un servidor de copias de seguridad podría no albergar información útil, como el syslog del servidor que hospeda la solución de identidades. Use parámetros para determinar la referencia para todas las aplicaciones, teniendo en cuenta elementos como el inicio/cierre de sesión del usuario, la actividad de la red, la actividad del sistema, las transacciones, etc. En general, los datos del registro deben incluir quién (identidad de usuario), cuándo (marca de tiempo de inicio y fin de la actividad), qué (actividad realizada) y dónde (IP de origen). En algunos casos, se trata de una norma de la industria o de seguridad que dicta el requisito de registro, por ejemplo, PCI-DSS, ISO27001.

Las organizaciones exitosas no solo capturan todos sus datos; los organizan con cuidado



En esta imagen, verá que el Análisis de tendencias de machine learning para análisis de datos proporciona telemetría de la operación de seguridad de una organización. Hasta ahora, en este Consejo, analizamos cómo la generación de eventos y la recopilación de eventos son procesos importantes para llevar a cabo un análisis eficaz. Más adelante, exploraremos cómo usarlos de forma paralela con la correlación de eventos para proporcionar información significativa a su equipo de DevSecOps.

Aproveche la supervisión basada en máquinas para el análisis y la generación de informes

Los sistemas actuales generan más datos y eventos de los que los humanos pueden interpretar por sí solos. Muy a menudo, los datos sin procesar son inútiles. Además, los eventos recopilados deben correlacionarse entre sí para proporcionar un panorama más amplio. La forma de superar estos desafíos es usar la supervisión basada en máquinas. Idealmente, debe recopilar y agregar sus registros y datos de eventos de diversas fuentes, e identificar, analizar, estandarizar y ayudar a encontrar correlaciones. Al correlacionar eventos, puede reducir el número de falsos positivos y descubrir amenazas que, de lo contrario, no se habrían detectado. Por ejemplo, supongamos que sus sistemas presenciaron tres eventos que, por sí solos, no activarían alarmas. Pero, después de correlacionar estos eventos en conjunto, el incidente se revelaría claramente como un problema más grande.

La supervisión basada en máquinas comienza con la configuración de la referencia para el comportamiento "normal" de la aplicación para poder detectar una desviación significativa. La incorporación de análisis avanzados y machine learning ayuda a indicar cualquier comportamiento inusual que señale una infracción. El siguiente paso consiste en modelar el comportamiento normal del sistema mediante datos de capacitación y vigilando cualquier desviación. Mientras que los datos de registros evaluados de forma genérica y los datos de otras fuentes a menudo ofrecen resultados falsos positivos, las soluciones de machine learning maduras realizan predicciones correctas sistemáticamente.

Si bien la supervisión basada en máquinas libera información para proteger mejor a la empresa, el siguiente paso complementario es generar informes útiles. Los informes útiles significa que sean lo suficientemente flexibles como para proporcionar información con diferentes niveles de datos, a menudo diseñados para diferentes públicos. Pueden localizar y observar de cerca un dominio concreto, a la vez que ofrecen un panorama más amplio de la empresa.

Otro principio de los informes útiles es la comprensión humana. Debido a que la mente humana está diseñada para notar patrones en formas y colores, la visualización de los datos ayuda a comprender y analizar la correlación. Es mucho más rápido detectar eventos fuera del rango de actividades habituales cuando se muestran como gráficos, en lugar de tablas de datos. También es una forma de mejorar la participación de los miembros del equipo, ya que a menudo los gráficos son más fáciles de comprender. Úselos como el primer enfoque al presentar los asuntos de seguridad a los desarrolladores antes de pasar a problemas más complejos.

Si bien la supervisión basada en máquinas libera información para proteger mejor a la empresa, el siguiente paso complementario es generar informes útiles.

Combine alertas prácticas con inteligencia contra amenazas para habilitar la seguridad proactiva

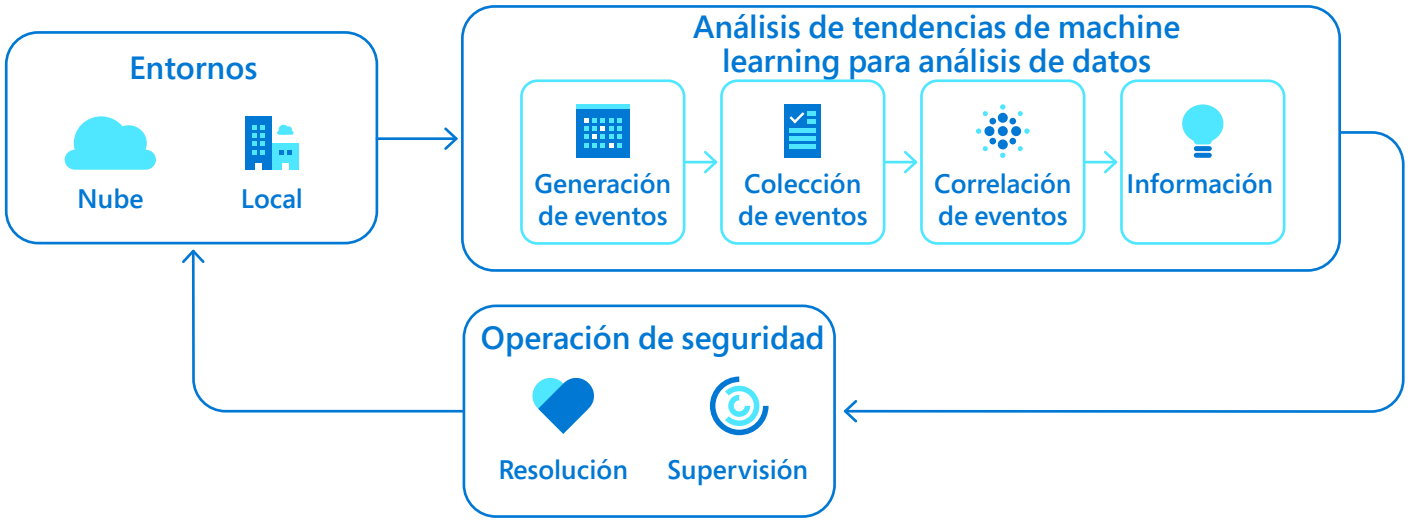
Las empresas informadas usan la inteligencia contra amenazas para evaluar las posibles amenazas en comparación con las registradas. La inteligencia contra amenazas recopilada de varias fuentes sobre amenazas emergentes y existentes proporciona una mayor comprensión de la capacidad de amenaza, los IOC (indicadores de ataque), las tácticas, técnicas y procedimientos (TTP) y los controles de mitigación que se utilizarán en su contra.

Compartir datos de la inteligencia contra amenazas es una situación en la que todos ganan. Las organizaciones pueden compartir experiencias con diferentes amenazas según su propia diversidad única de servicios, ubicación geográfica, tecnología, etc. Esto dota a todas las organizaciones con el conocimiento para detectar estos agentes de amenazas. Pueden ubicar a los agentes de amenazas en función de los patrones de explotación, las herramientas y técnicas repetidas que se usan, las ubicaciones comunes o los objetivos verticales. También es importante que integre la administración de vulnerabilidades con la inteligencia contra amenazas para determinar qué vulnerabilidades representan los mayores riesgos en función del panorama de amenazas.

Sin embargo, la supervisión por sí sola no resolverá el problema si no hay ninguna respuesta. La respuesta oportuna a los incidentes de seguridad es fundamental. Al mismo tiempo, responder requiere de puntos de datos abundantes para el análisis, lo que a su vez lleva tiempo y está sujeto a la disponibilidad de un analista de seguridad. Ayude a resolver el incidente mediante la recopilación de datos, como la ubicación del usuario, el nombre del dispositivo comprometido, la dirección IP, el tipo de dispositivo y la última fecha de revisión en varios sistemas, como Active Directory y la base de datos de administración de configuración (CMDB).

La automatización de las respuestas puede ser primordial para mitigar los incidentes de manera oportuna y evitar un incidente grave. Deberá integrar una herramienta de coordinación con varios otros sistemas para analizar los datos recopilados para generar una respuesta y corrección automatizadas. Estas son un par de tácticas de corrección rápida: 1. bloquear a los usuarios a través de herramientas de identidad al detectar comportamientos malintencionados y 2. bloquear los puertos del firewall en respuesta a un ataque DDoS.

La inteligencia contra amenazas recopilada de varias fuentes sobre amenazas emergentes y existentes proporciona una mayor comprensión de la capacidad de amenaza, los IOC (indicadores de ataque), las tácticas, técnicas y procedimientos (TTP) y los controles de mitigación que se utilizarán en su contra.



Con los pasos que hemos analizado, su empresa ahora está preparada con un poderoso bucle de Análisis de tendencias de machine learning para análisis de datos para informar mejor a sus equipos de seguridad.

Para una supervisión proactiva, también debe integrar las herramientas de administración de cambios, lo que impide la implementación de cambios no autorizados en la producción. Asegúrese de ejecutar todas las actualizaciones a través del sistema de administración de cambios para validarlos antes de su lanzamiento. Si la corrección es necesaria, el sistema de administración de cambios ejecutará la actualización correcta y emitirá los vales de servicio de alta prioridad relacionados para una investigación adicional.

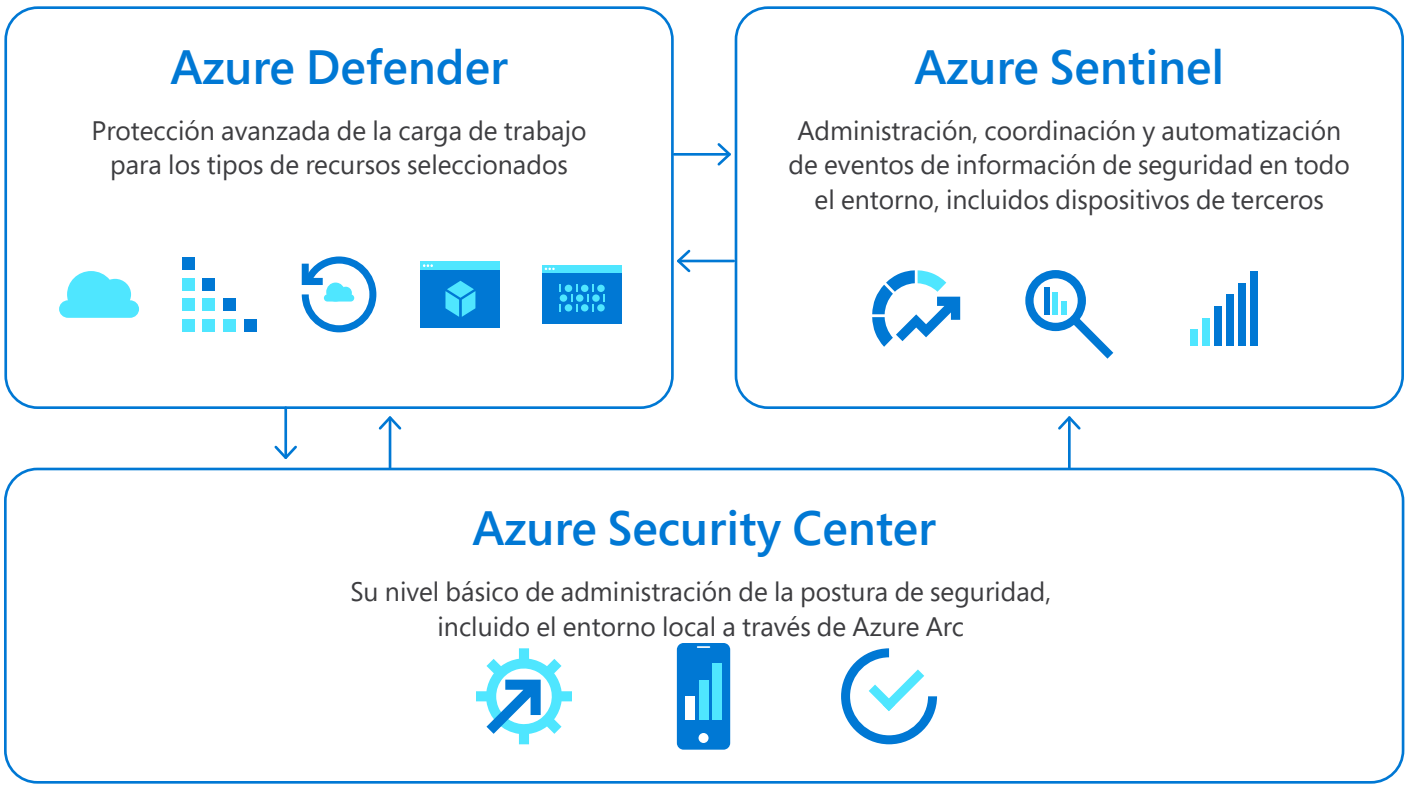
Para ilustrar qué herramientas ayudan a llevar a cabo una supervisión y observabilidad coherentes, revisemos algunos productos de Microsoft asignados al ciclo de vida del desarrollo de software.

Incorpore una cadena de herramientas sólida creada para las amenazas modernas

A menudo, la tarea más difícil es encontrar una cadena de herramientas de seguridad que funcione para su empresa. Las soluciones fragmentadas también complican las cosas. Es fundamental que los componentes de la cadena de herramientas trabajen en armonía y protejan a su empresa desde todos los ángulos. Azure está diseñado para proveer a su organización una cadena de herramientas que protege todo el ciclo de vida del desarrollo de software (SDLC), sin problemas. En la siguiente imagen, se muestra una vista general de qué herramienta se relaciona con cada función del marco de ciberseguridad de NIST:

Identificación	Protección	Detección	Respuesta	Recuperación
Azure Security Center	Azure Defender	Azure Sentinel		

Echemos un vistazo más de cerca para ver cómo funciona cada componente en conjunto:



Azure Security Center: Azure Security Center ofrece administración de la postura de seguridad y protección contra amenazas para las cargas de trabajo de la nube híbrida al supervisar constantemente los recursos para las configuraciones de seguridad con errores. Al trabajar en conjunto con Azure Security Center, Azure Arc ayuda a estandarizar la visibilidad, las operaciones y el cumplimiento en una amplia gama de recursos y ubicaciones al extender el plano de control de Azure. Los procedimientos recomendados o las normas pertinentes para una organización (PCI, HIPAA, ISO, RGPD, etc.) pueden impulsar la orientación de la configuración. El estado de seguridad de todos los recursos se visualiza a través de Azure Secure Score, que proporciona información sobre la postura de seguridad actual de su organización. A partir de esto, Azure Security Center realiza recomendaciones para mejorar la puntuación, de modo que su empresa tenga un camino para avanzar.

Azure Defender: Azure Defender es uno de los componentes principales de Azure Security Center y brinda protección a las cargas de trabajo, los recursos y los servicios estratégicos en los entornos de nube. Azure Defender ofrece seguridad para los servidores, las instancias de App Service, el almacenamiento, SQL y SQL Databases, Key Vault, el administrador de recursos, DNS, los clústeres de Kubernetes y los registros de contenedores. No se preocupe, Azure defender también proporciona protección a los servidores que no son de Azure, como los que se hospedan en los centros de datos locales u otros proveedores de nube. Azure Defender es una herramienta centralizada con una amplia cobertura y profundidad de las capacidades de protección de seguridad en toda la carga de trabajo nativa de la nube.

Para empezar a proteger las cargas de trabajo, analice las VM, los servidores SQL y los registros de contenedores para detectar vulnerabilidades (analizadas por el servicio de nube de Qualys). A continuación, debe configurar Azure Defender para proporcionar los resultados analizados con prioridad según lo crítico que sean y emparejados con la última revisión disponible. Luego, aproveche el control de aplicaciones adaptable de Azure Defender para hacer referencia a todas las aplicaciones y VM seguras conocidas. Cualquier desviación de esta referencia a su vez activará una alerta de seguridad. Esta es solo una forma de aprovechar Azure Defender. Si habilita Azure Defender para los servidores, puede usar el acceso a VM just in time para bloquear el tráfico entrante a sus VM, lo que reduce la exposición a ataques mientras facilita el acceso para conectarse a las VM cuando sea necesario. Otros casos prácticos incluyen comprobaciones de integridad de archivos, protección de red adaptable y mapas de red.

Azure Sentinel: Azure Sentinel ofrece funcionalidades de Información de seguridad y administración de eventos (SIEM) y Coordinación, automatización y respuestas de seguridad (SOAR) a Azure para recursos nativos de la nube y recursos locales. También está listo para integrarse con las soluciones que no son de Microsoft mediante las API. Azure Sentinel analiza los datos/registros recopilados en busca de amenazas y correlaciones de eventos. Cualquier amenaza detectada se notifica como un incidente para su corrección. Y para la coordinación, Azure Sentinel incluye un manual que detalla más de 200 conectores con diferentes soluciones. El manual ayuda a suavizar el proceso de integración con otras herramientas y describe los pasos de automatización propuestos para el incidente, como enviar un vale en ServiceNow.

Azure Security Center, Azure Defender y Azure Sentinel trabajan en conjunto mediante la automatización del flujo de trabajo a través de Azure Logic Apps, que puede desencadenar respuestas (aplicaciones) en eventos predefinidos. Por ejemplo, Azure Security Center puede desencadenar flujos de trabajo para responder a las amenazas con Azure Defender, como enviar una notificación por correo electrónico a un equipo de seguridad sobre alertas de alta gravedad para que puedan investigarlas. Además, puede crear un Grupo de seguridad de red para contrarrestar un ataque de fuerza bruta. Del mismo modo, Azure Sentinel puede desencadenar un bloqueo en los usuarios de Azure Active Directory en caso de que se comprometa cualquier identidad. Estos servicios no solo proporcionan información significativa para sus recursos, sino que también pueden configurarse para responder de forma automática a cualquier infracción de seguridad.

Para ilustrar cómo se ve en la práctica la supervisión coherente de amenazas, echemos un vistazo al caso ficticio de la empresa de servicios financieros, Pension Experts.

El caso de Pension Experts

Remediar de forma proactiva con alertas de amenazas automatizadas

Matilda es la directora general de Pension Experts, una empresa de servicios financieros en Ámsterdam, Países Bajos. Antes se sentía segura con la postura de seguridad de su empresa, pero han pasado años desde la última actualización de seguridad. Pension Experts ofrece administración de pensiones de servicio completo y recopila información de identificación personal (PII). Matilda sabe que la PII, como la asignación de pensiones, la información de cuentas y el registro de pagos, exige un enfoque exhaustivo hacia la seguridad, en especial en los Países Bajos, donde deben cumplir con los requisitos del Banco Central de los Países Bajos (DNB) y la Autoridad holandesa para los mercados financieros (AFM).

Debido a que necesitaba entender más sobre los procesos de seguridad actuales, Matilda llamó a la directora de seguridad de la información, Stephanie, y a un líder del equipo de seguridad, Gregory, para obtener más información. Ansioso por ayudar, Gregory describió la postura de seguridad actual y la forma en que protegen las diferentes cargas de trabajo. Señaló que, si bien era funcional, su sistema actual no podía encontrar y corregir las amenazas de forma rápida y proactiva. Stephanie estuvo de acuerdo y poco después comenzó a investigar soluciones que ayudan a administrar la postura de seguridad. El equipo acordó volver a reunirse en una semana y revisar algunas de las mejores opciones de Stephanie.

Después de que el equipo de DevSecOps de Pension Experts debatió con vehemencia las principales ofertas de seguridad que presentó Stephanie, finalmente seleccionaron una solución compuesta por Azure Security Center, Azure Defender y Azure Sentinel. Con Azure Security Center, el equipo pudo supervisar continuamente todos sus entornos mientras evaluaba de forma eficaz el estado actual de su organización. Sinceramente, lo que más entusiasmaba al equipo era usar las capacidades de Azure Defender y Azure Sentinel para localizar y corregir rápidamente las amenazas. Este sistema no solo pudo llevar a cabo el análisis de vulnerabilidades, sino que les brindó la oportunidad de configurar respuestas automatizadas a las amenazas.

Seis meses después, el equipo de DevSecOps había implementado completamente la solución. Gregory y Stephanie están muy satisfechos con el mayor tiempo de respuesta a las amenazas y la telemetría que ofrece la solución. En lugar de esperar a que sus análisis de seguridad programados aborden las amenazas, Gregory recibe un recordatorio automatizado cada vez que aparece una amenaza. Ahora no se preocupa por una amenaza que existe durante días antes de que se realice un análisis. Incluso configuró el sistema nuevo para analizar las vulnerabilidades con cada versión del código para que pueda detectar las amenazas a medida que ocurren. Lo mejor de todo es que Matilda, la directora general, ya no se preocupa por la capacidad de la empresa para eludir amenazas antes de una infracción desastrosa.

CONSEJO 4

Adopte todo como código

Adoptar un enfoque de "todo como código" ayuda a la confiabilidad de la implementación de su empresa, el control de versiones y la eficacia de las pruebas

Cuando sus equipos realizan tareas tediosas de forma manual, como el aprovisionamiento de infraestructura o la administración de implementaciones de aplicaciones, no pueden desarrollar código nuevo e innovador. Un enfoque de todo como código optimiza el desarrollo, la entrega y la administración de software, lo que libera a los equipos de desarrolladores para centrarse en el desarrollo.

Al igual que DevOps, un enfoque de todo como código permite operaciones más eficaces al estandarizar los mecanismos del desarrollo de software. Un enfoque de todo como código codifica los aspectos del desarrollo, como la infraestructura, el esquema y las canalizaciones, lo que le permite administrar la gobernanza de los archivos de directivas en lugar de los procesos manuales. Piense en ello como la aplicación ideológica de aplicar un enfoque de desarrollo de aplicaciones a otros componentes de TI (incluido DevOps) para garantizar que se definan y sigan con el mínimo esfuerzo los procedimientos recomendados.

Uno de los mayores beneficios de un enfoque de todo como código es el menor riesgo de error humano. Con los flujos de trabajo definidos como código, hay menos probabilidades de que un ingeniero que sigue una lista de comprobación manual se olvide de un paso o haga clic en un botón incorrecto por error. Es más fácil aprobar auditorías con configuraciones de todo como código que registran automáticamente el historial de actualizaciones del sistema a través de cambios de Git.

Un enfoque eficaz de todo como código abarca una variedad de elementos: (1) infraestructura como código, (2) infraestructura inmutable, (3) un sistema de control de versiones seguro, (4) configuración como código, (5) canalización como código y (6) directiva como código.

Empiece su transformación con la infraestructura como código

Muchas empresas inician su transición a todo como código mediante la adopción de la infraestructura como código (IaC), que les permite administrar su infraestructura de TI mediante archivos de configuración. Una herramienta de IaC es Terraform, una herramienta desarrollada por Hashicorp que le permite crear, cambiar y mejorar de forma previsible infraestructura segura. Profundizaremos en algunas aplicaciones prácticas de Terraform más adelante en este consejo.

La mayoría de las empresas adoptan la IaC para quitar la carga a sus equipos de administrar infraestructura de forma manual. El trabajo manual y tedioso ralentiza su empresa y no puede proporcionar la eficiencia, la escalabilidad y las pruebas necesarias para la seguridad moderna.

A continuación, se muestran algunos ejemplos de cómo una implementación exitosa de infraestructura como código alivia los puntos débiles de la administración manual de TI:

Desafíos de la administración manual de la infraestructura

Problemas de escalabilidad e incoherencia

Dado que la configuración manual es lenta, las aplicaciones no pueden escalar de forma automática. Para complicar esto, los administradores del sistema suelen administrar la carga mediante la creación manual de servidores. Estos retrasos también pueden afectar la disponibilidad general del sistema ya que los trabajadores pierden el acceso a los sistemas.

Además de la escalabilidad, la incoherencia también es una de las principales preocupaciones. Muchas implementaciones o configuraciones no son repetibles, lo que genera entornos heterogéneos y discrepancias entre los entornos de desarrollo y los entornos de producción.

Mejoras de la infraestructura como código

Repetibilidad de las implementaciones

Con la IaC, habilita las implementaciones y configuraciones escalables en el entorno de desarrollo, entorno de ensayo y entorno de producción. También minimiza el desfase entre los entornos. Por ejemplo, cada entorno se compila con la misma versión de cada componente en los sistemas de TI: entornos de desarrollo, ensayo y producción. Además, puede asegurarse de que el middleware, el sistema operativo y los parches de seguridad sean coherentes en todos los entornos.

Desafíos de la administración manual de la infraestructura (continuación)

Alto costo

La administración manual de la infraestructura es un esfuerzo costoso. La creación de servidores y redes y la configuración de máquinas cuesta dinero y mano de obra. Esto también significa que su empresa debe emplear a especialistas en todos los dominios de TI para crear y mantener la infraestructura. Más aún, también necesitará especialistas en seguridad para analizar y corregir las vulnerabilidades, las configuraciones erróneas y los intentos de hackeo en su infraestructura. Ejecutar entornos todo el día todos los días no es rentable. A menudo, los equipos no destruirán ni cerrarán los entornos porque les resulta difícil recrear o reconfigurarlos fácilmente.

Capacidades de supervisión insuficientes

Sin la IaC, debe asegurarse manualmente de que el sistema funcione de manera óptima sin ningún cuello de botella. Los problemas pueden ser difíciles de identificar y deberse a numerosas causas: configuraciones incorrectas, tamaños de servidor incorrectos, problemas de red o incluso un diseño de aplicaciones deficiente.

Mejoras de la infraestructura como código (continuación)

Ahorros en eficiencia ambiental

Con la IaC, ahorra a través de la administración eficaz del entorno. Sus equipos podrán destruir los entornos que no usan sin preocuparse de pérdidas. Con todo lo almacenado y en versiones, puede recrear rápidamente sus entornos al ejecutar sus canalizaciones de CI/CD. Por ejemplo, puede crear entornos efímeros para fines de prueba en su canalización de CI/CD mediante la ejecución de pruebas, como pruebas de integración, pruebas de humo, pruebas de aceptación de usuarios, etc.

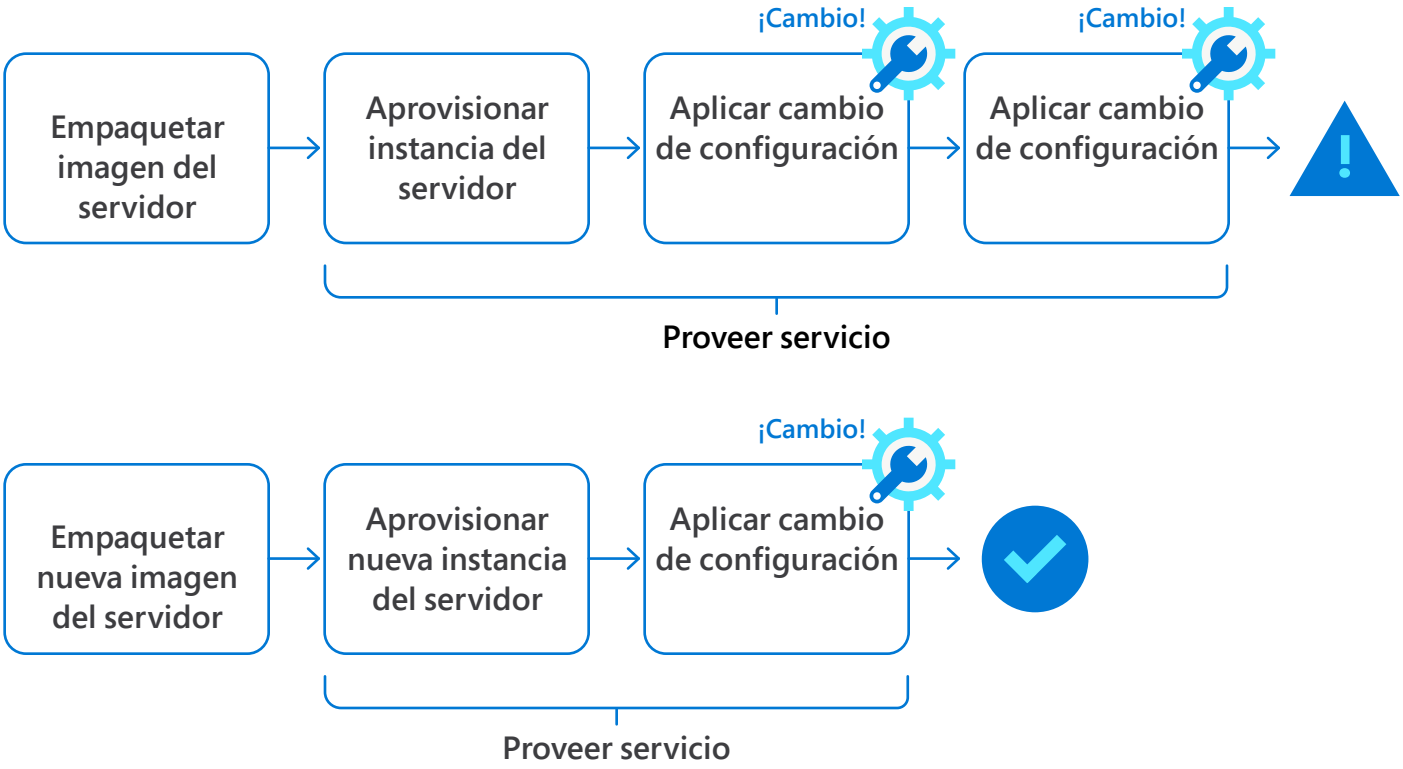
Mejores versiones y pruebas

Con la IaC, su empresa ahora puede aprovechar los paradigmas de desarrollo para la infraestructura. Por ejemplo, puede almacenar su infraestructura en Git, crear una versión y ver el historial de todos los cambios de los desarrolladores. Luego puede rastrear fácilmente cuándo aparece un error y cuándo se produce la corrección. También puede aplicar pruebas directas en el código, denominadas "infraestructura basada en pruebas", donde puede probar un entorno antes de implementarlo. Esto ayuda a que la infraestructura mantenga la conformidad con lo que se requiere.

La IaC es solo una forma eficaz de aumentar la repetibilidad, mejorar el control de versiones y reducir los costos, pero la IaC por sí sola tiene limitaciones. En la próxima sección, exploraremos cómo la adopción de infraestructura inmutable agrega mayores controles de seguridad a su enfoque de todo como código.

Adopte infraestructura inmutable

Incluso con la infraestructura como código instaurada, los servidores de larga ejecución corren el riesgo de desfase de la configuración, un proceso en el que la infraestructura que aloja los datos o la producción de un centro de datos con el tiempo se aleja de las configuraciones de copias de seguridad y recuperación. Los servidores administrados manualmente son en particular propensos al desfase de la configuración. No es posible administrar completamente la configuración de un servidor, lo que significa que existen muchas oportunidades para que se produzca el desfase de la configuración u otros cambios inesperados en el servidor. Lamentablemente, cada vez que aparece un desfase o un servidor deja de funcionar, debe volver a compilar desde cero y aplicar una configuración.



La imagen anterior describe la diferencia entre la administración de cambios eficaz y con errores. La línea superior muestra un sistema donde se aplican demasiados cambios, lo que conduce al desfase de la configuración, mientras que la segunda línea presenta un nuevo sistema de implementación con menos sucesos en la actualización.
En última instancia, limitar las oportunidades de actualizaciones dentro de un sistema disminuye la posibilidad de desfases de la configuración.

La práctica de infraestructura inmutable representa un nuevo enfoque para actualizar la infraestructura. En el pasado, cada vez que se requería una actualización, se insertaba una actualización posteriormente en la VM en cuestión. Esto expone su organización al desfase de la configuración. La diferencia con la infraestructura inmutable es que ahora, en lugar de actualizar las VM existentes, se crea una VM completamente nueva que es una copia actualizada del servidor antiguo. Esto le permite instalar y analizar automáticamente las imágenes antes de implementarlas en la nube. En un enfoque de DevSecOps, las imágenes se prueban intensamente para detectar cualquier CVE presente antes de su implementación a escala, por lo tanto, es importante no usar ninguna herramienta de administración de configuración que cree oportunidades para cambios no probados. Un procedimiento recomendado es aplicar los cambios necesarios en la imagen base, probarla y, a continuación, implementarla. Asegúrese de eliminar y reemplazar los servidores que aún no reciban actualizaciones.

Si desea ir un poco más allá, puede crear una imagen inmutable sin ningún acceso remoto o acceso secure shell (SSH). Esto protege aún más los servidores contra los ataques, especialmente si se exponen en Internet. Y si un servidor se ve comprometido, puede separarlo del grupo de clústeres para realizarle un estudio forense. Después, vuelva a crear un nuevo servidor con una configuración válida y conocida aplicada a la imagen base inmutable. Para acelerar el tiempo de arranque del servidor, las herramientas como Azure Image Builder le permiten automatizar la creación de imágenes base configuradas con todo lo que necesita, incluida la aplicación.

La infraestructura inmutable es muy útil para aprobar auditorías de seguridad, ya que sus equipos pueden realizar un seguimiento de la creación y las correcciones de vulnerabilidades en tiempo real. Si se detecta una nueva CVE, puede volver a compilar una nueva imagen base con la solución de seguridad propuesta e implementarla automáticamente en su clúster, lo que soluciona la vulnerabilidad sin ninguna interrupción del servicio. Al final, la infraestructura inmutable adopta el mismo paradigma que la creación de imágenes de Docker: las implementa en un clúster y las actualiza cada vez que se crea una nueva imagen.

La capacidad de controlar la versión de las imágenes es uno de los mayores beneficios de la infraestructura inmutable. Con la infraestructura inmutable, puede crear un archivo, la "receta" para crear la imagen, y almacenarla en el control de versiones. Luego, se generan las versiones del archivo, en donde puede ver cada cambio que los equipos realicen posteriormente. Por supuesto, es importante asegurarse de que la imagen se administre de forma segura.

La infraestructura inmutable es muy útil para aprobar auditorías de seguridad, ya que sus equipos pueden realizar un seguimiento de la creación y las correcciones de vulnerabilidades en tiempo real.

Almacene el código en un sistema de control de versiones seguro

La mejor manera de almacenar cualquier archivo de control de versiones que cree a través de una infraestructura inmutable es dentro de un sistema de control de versiones seguro. Estos sistemas permiten a su empresa definir un acceso detallado a cualquier repositorio que contenga la código de la empresa. Hoy en día, el formato más utilizado es Git.

Conectar el directorio de su empresa a su Git protege el acceso tanto a los repositorios de aplicaciones como de infraestructura. Por ejemplo, puede conectar GitHub a su Active Directory para ayudar a proteger el acceso a través del inicio de sesión único (SSO). Además, puede ser aún más estricto con el control de acceso al aprovechar la autenticación multifactor (MFA) junto a un Active Directory.

Es fundamental que defina un acceso minucioso a su repositorio. Empiece por definir diferentes roles (colaborador, propietario, lector) con diferentes directivas de acceso al código. Asegúrese de que su rama maestra (o principal) también esté protegida. Para ello permita que solo unos pocos roles selectos contribuyan a la rama maestra o principal. Por último, para proteger aún más sus sistemas de producción, limite las contribuciones a las solicitudes de incorporación. Estas directivas vigentes de acceso y control de versiones no solo hacen que su empresa sea más segura, sino que también ayudan a prepararse mejor para la auditoría de seguridad.

La mejor manera de almacenar cualquier archivo de control de versiones que cree a través de una infraestructura inmutable es dentro de un sistema de control de versiones seguro.

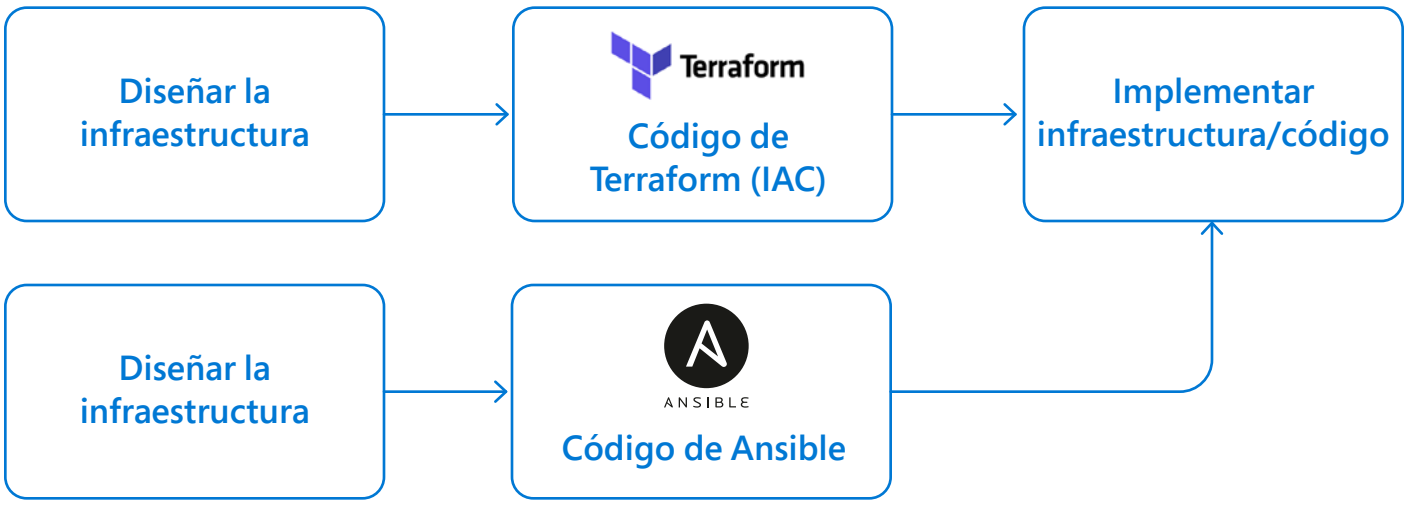
Establezca la configuración como código

En este punto, solo hemos analizado la infraestructura como código, pero un enfoque de todo como código va más allá de la infraestructura. La configuración como código (CaC) lo lleva un paso más allá para administrar también los recursos de configuración. En última instancia, la CaC permite replicar las configuraciones del servidor en los entornos, todo sin intervención humana. Ansible es un excelente ejemplo de una herramienta de automatización de open source para la administración de la configuración y la implementación de aplicaciones que puede ayudarlo a lograr la adopción total de la CaC.

La poderosa automatización de Ansible simplifica las tareas largas y complejas, lo que permite a sus equipos concentrarse en el desarrollo de valor agregado. Funciona en un nivel superior a los lenguajes de desarrollo y se ejecuta en archivos YAML para implementar configuraciones en tipos de figura específicos. Esto significa que no necesita instalar ningún otro software en el servidor.

Con Ansible, puede implementar rápidamente las aplicaciones y dividir sus configuraciones en módulos individuales. Ansible ejecuta cada módulo mediante el uso de manuales que actúan como un manual de instrucciones para controlar automáticamente los servicios, los paquetes y los archivos.

Las herramientas de CaC, como Ansible, trabajan en conjunto con otras herramientas de infraestructura como código, como Terraform. Terraform ayuda a definir y crear la infraestructura de su sistema. Por otra parte, Ansible configura e implementa las aplicaciones mediante la ejecución de sus manuales en las instancias de servidor proporcionadas. Utilice las integraciones de Terraform para ejecutar un script de Ansible específico a fin de usar los programas en conjunto. Una vez unidos, estos programas forman una plataforma sólida para la adopción eficaz de la configuración como código.



Esta figura muestra un entorno de IaC completo. En este ejemplo, la canalización implementa la infraestructura después de diseñarla con Terraform. Después de este paso, la infraestructura se configura con Ansible.

Implemente la canalización como código

Para configurar la canalización como código, adopte el enfoque del estilo de desarrollo de aplicaciones utilizado para la infraestructura como código o la configuración como código. En un enfoque de canalización como código, almacena todas las canalizaciones de CI/CD dentro del sistema de control de versiones como archivos, lo que permite un control de versiones más estricto para las revisiones de seguridad.

La adopción correcta de la canalización como código en la canalización de aplicaciones mejora la seguridad en todas las etapas de la implementación. Por ejemplo, ahora puede analizar su código de aplicación holístico mediante los métodos de prueba de seguridad de aplicaciones estáticos (SAST), dinámicos (DAST) e interactivos (IAST). También puede integrar la autoprotección de aplicaciones en tiempo de ejecución (RASP) en los servidores para proteger aún más las aplicaciones.

Piense en la canalización como código como una pauta de implementación de aplicaciones para su equipo. Esta pauta ofrece una vista general de todas las etapas necesarias para compilar e implementar aplicaciones seguras. Con todos estos aspectos ahora definidos "como código", se puede aplicar e implementar aplicaciones seguras a gran escala para todos sus equipos.

**Piense en la canalización como código como una pauta de implementación de aplicaciones para su equipo.
Esta pauta ofrece una vista general de todas las etapas necesarias para compilar e implementar aplicaciones seguras.**

En el siguiente consejo, veremos otro elemento del enfoque de todo como código, la directiva como código y sus aplicaciones eficaces cuando se usan junto con los otros elementos "como código".

CONSEJO 5

Alcance el cumplimiento con la automatización de directivas

Tanto el panorama normativo como el software que rigen están cambiando constantemente, lo que exige un enfoque automatizado para el cumplimiento de las directivas

Los regímenes normativos son cada vez más rigurosos. Eso es positivo. Sin embargo, garantizar el cumplimiento en un entorno emergente de aplicaciones no es fácil. El Informe de Enterprise DevOps 2020–2021[1] indicó que casi la mitad de los ejecutivos encuestados dijeron que no estaban seguros de qué normas de cumplimiento de datos debían cumplir. Además, el informe señaló que ya no basta con simplemente comprobar la seguridad de una aplicación o un entorno cuando se implementan por primera vez. Claramente, la solución lógica es garantizar el cumplimiento continuo en cada paso del desarrollo y la administración de aplicaciones (ADM).

La mejor manera de lograrlo es con la automatización de directivas.

No obstante, en primer lugar es necesario establecer directivas que se relacionen con los requisitos normativos y de cumplimiento de la empresa, los estándares de la industria y los objetivos organizacionales, de modo que cuando se realice una auditoría, el cumplimiento de la seguridad nunca esté en duda. Cuando se crean correctamente, estas directivas ofrecen un conjunto de controles a sus equipos DevSecOps, lo que garantiza que las vulnerabilidades de seguridad se supervisen y corrijan en cada etapa del ADM.

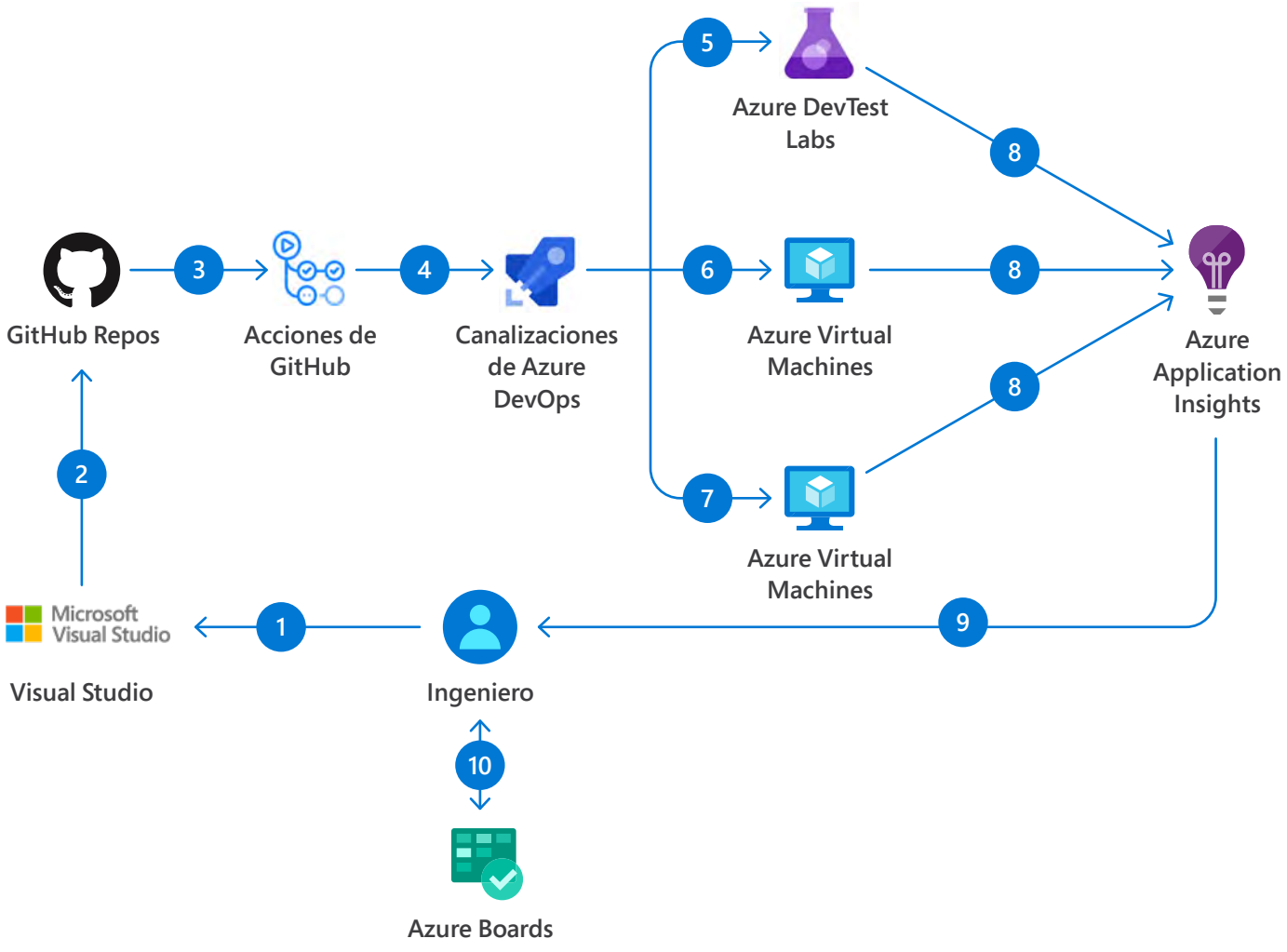
Para lograr un cumplimiento continuo, su organización debe seguir seis pasos distintos hacia la automatización de políticas. La implementación de estos seis pasos, a su vez, también ayudará a instaurar un bucle cerrado de la automatización de directivas.

[1]Microsoft, Informe de Enterprise DevOps 2020–2021, 2020

Seis pasos para la automatización correcta de directivas

¿Cómo aplica de forma continua los estándares de cumplimiento en todo el ciclo de DevOps? No olvidemos que, en algunos casos, la vida útil del sistema puede extenderse durante muchos años. Esta cantidad de tiempo hace que sea inevitable el no cumplimiento de la directiva, a menudo, debido a que el panorama normativo evoluciona. Sin embargo, puede asegurarse de que todas las cargas de trabajo y aplicaciones sigan cumpliendo con las directivas al aprovechar que los procedimientos recomendados para directivas de los proveedores de nube actúen como protecciones. El equipo de seguridad o fundamental suele administrar estas directivas y deben automatizarse.

En el ejemplo que se muestra aquí, las máquinas virtuales (VM) se implementan en Azure y un bucle de retroalimentación basado en telemetría se enlaza con el equipo de desarrollo. En este caso, se definió que las VM de Azure no podían tener una IP pública. Sin embargo, este requisito pasaría desapercibido hasta que se produzca la implementación en los pasos 5, 6 y 7. Antes de esos pasos, el equipo de ingeniería no recibió actualizaciones sobre estas directivas. Esto expone a la empresa a las vulnerabilidades a través de una directiva obsoleta. Para cualquier actualización, el equipo necesitaría revisar las directivas actualizadas de la plataforma en la nube y garantizar que todo su código cumplió con las directivas.



La automatización de políticas se basa en el argumento del Consejo 2 (Integre la seguridad en las primeras etapas del ciclo de vida del desarrollo) para ilustrar por qué cambiar la seguridad a la izquierda es fundamental para todas las empresas. Si tomamos Microsoft Azure como referencia, estos son los seis pasos para que su empresa alcance el cumplimiento continuo con la automatización de directivas:

1. **Determine su conjunto de directivas**
2. **Adopte un modelo de directiva como código**
3. **Actualice las directivas en el código e insértelo en Azure**
4. **Cierre el bucle con el análisis de cumplimiento**
5. **Cambie a la izquierda con una puerta de calidad**
6. **Use Azure Security Center para supervisar y observar**

Paso 1: Determine su conjunto de directivas

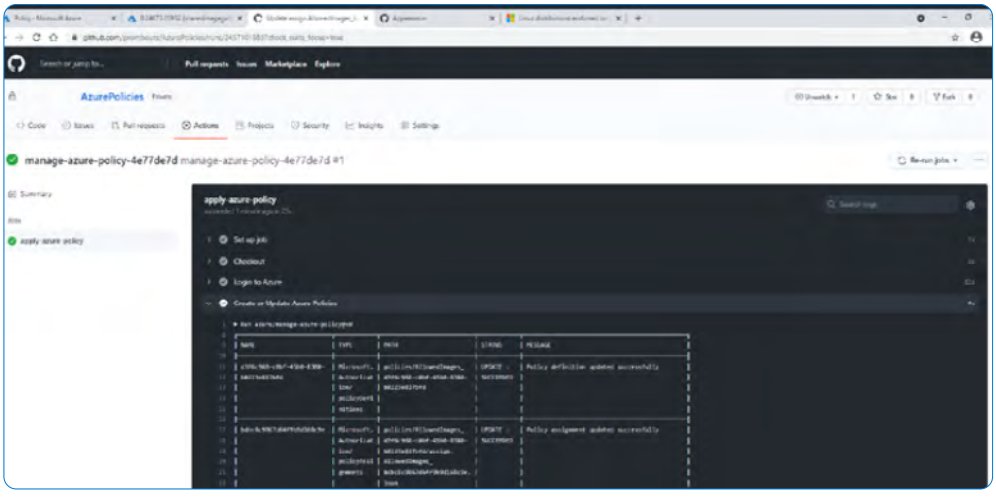
El primer paso en el proceso es determinar qué directivas desea usar. Su sector de la industria (por ejemplo, banca, atención de salud, etc.) tendrá requisitos de cumplimiento específicos a los que atenerse. En Azure, puede seleccionar una plantilla de ejemplo para comenzar, como CIS, ISO27001 y PCI-DSS. Es fácil configurar estas referencias en Azure muy rápido con Azure Blueprints.

Paso 2: Adopte un modelo de directiva como código

El segundo paso es asegurarse de que las directivas se almacenen en un repositorio de código. Usar un repositorio, como GitHub, le permite exportar estas directivas y asegurarse de que se controlen con la versión. Esto también establece el estado operativo para realizar futuras actualizaciones de directivas e insertar cambios en Azure a través del código.

Paso 3: Actualice las directivas en el código e insértelo en Azure

Después de que las directivas se controlen con la versión y se actualicen en GitHub, puede insertar cambios en Azure. Puede configurar esta acción para que se active automáticamente cuando se actualicen las directivas. Esto hace que el seguimiento de los cambios sea sencillo con el control de versiones de GitHub.



Paso 4: Cierre el bucle con el análisis de cumplimiento

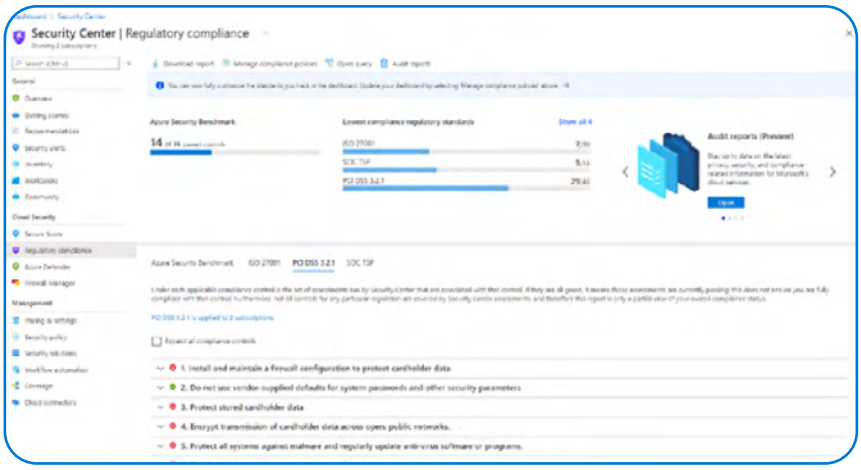
Con tantos equipos en toda la empresa, se generan constantemente servicios en la nube, lo que aumenta la posibilidad de que la empresa no cumpla con las directivas. Con sus directivas como código, ahora puede usar GitHub para ejecutar análisis de evaluación a petición en sus entornos de nube de Azure. Esto cierra el bucle al definir el estado actual mientras empodera a sus equipos con pasos prácticos para restaurar el cumplimiento según sea necesario.

Paso 5: Cambie a la izquierda con una puerta de calidad

Ahora que las directivas se crean y mantienen a través del código, los equipos de toda la empresa pueden entender, validar y probar las directivas respecto del código incluso antes en el ciclo de vida del software. Algunas herramientas útiles, como Open Policy Agent, actúan como una puerta de calidad en el sistema de implementación automatizada de Kubernetes (k8s) y prueban el código de Terraform en el repositorio antes de implementarlo en cualquier entorno. Esto significa que sus equipos pueden comprobar las directivas incluso antes de que se cierre una solicitud de incorporación y, posteriormente, corregirlos en la etapa más temprana.

Paso 6: Use Azure Security Center para supervisar y observar

El último paso es supervisar y observar el estado de cumplimiento de su empresa con Azure Security Center. En la siguiente figura, se muestra un resumen de comprobación de cumplimiento de Azure Security Center, que proporciona telemetría de cumplimiento en tiempo real. Esto resulta extremadamente útil en caso de que un auditor desee revisar el estado actual de una suscripción. Este resumen también ofrece tareas de corrección de Azure Security Center para resolver cualquier problema.



El objetivo final de estos seis pasos es habilitar un bucle cerrado de la automatización de directivas. Cuando se configura, el cumplimiento se supervisa de forma continua y se alinea con todos los cambios, en cualquier momento.

CONSEJO 6

Proteja y visualice la cadena de suministro de software

Comprender las dependencias de los sistemas de software que derivan de plataformas, marcos y componentes de terceros y open source le permite proteger aún más su cadena de suministro de software

El código propio no es la única fuente de vulnerabilidades de seguridad modernas. A menudo, el mayor peligro radica en las dependencias: cualquier código al que se hace referencia y se agrupa para hacer que un paquete de software funcione. En el software moderno, el 80 % o más[1] del código de la mayoría de las aplicaciones proviene de dependencias. Estas dependencias se basan en dependencias, lo que resulta en un complejo diagrama de relaciones. ¿El resultado? Un árbol de dependencias de relaciones complejas y dinámicas dentro del software que plantea una importante preocupación sobre la seguridad para las empresas. No entender el complejo árbol de dependencias completo de un sistema ofrece una vía para que los agentes malintencionados ataquen sus sistemas.

¿Cómo puede proteger su cadena de suministro de software?

El primer paso es establecer la transparencia en el historial de actualizaciones de cada componente, que incluyen los lanzamientos, los controles de calidad completados, las versiones y la documentación. Esto ayuda a establecer algo similar a una cadena de custodia en el código, los componentes y las dependencias posteriores. Pero esto es solo el comienzo.

Para proteger realmente su cadena de suministro de software, la empresa tendrá que comenzar (1) a actuar en función de la información recopilada por la visualización del árbol de dependencias y (2) aumentar la transparencia con una lista de materiales de software.

[1]GitHub, Octoverse Security Report, 2020

Actúe en función de los conocimientos recopilados por la visualización del árbol de dependencias

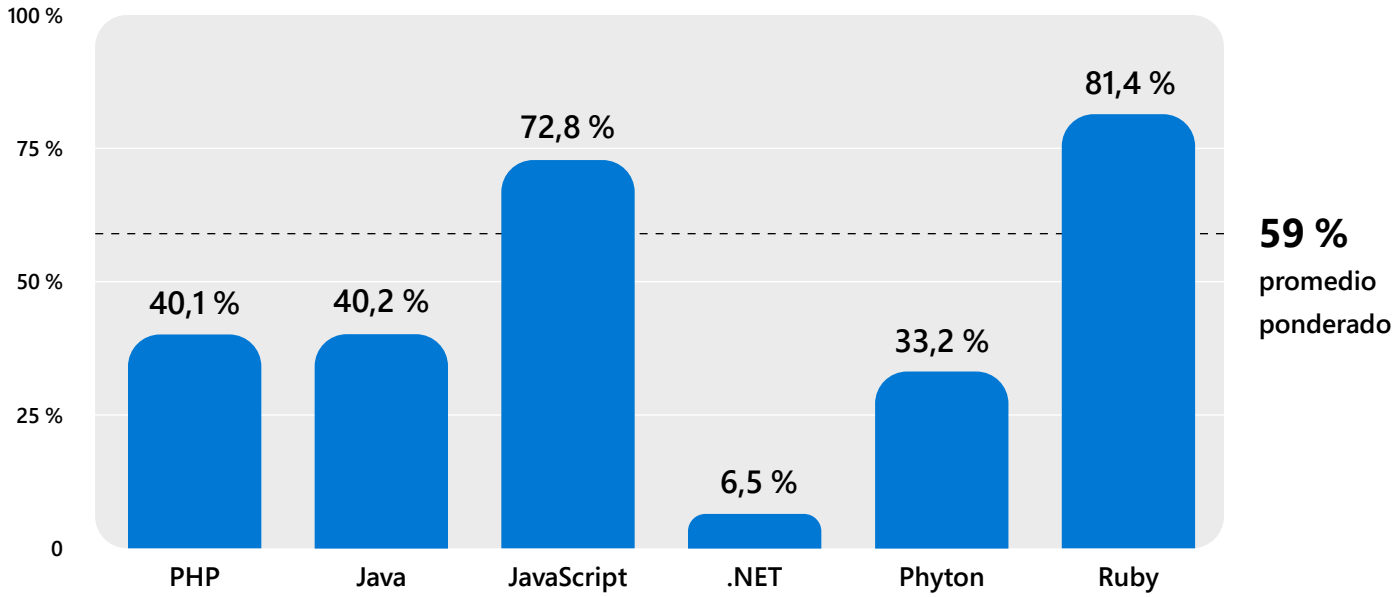
Solo visualizar las dependencias por sí solas no ofrece a su empresa una mayor seguridad. Si bien puede ofrecer excelente información a su organización, el paso clave es actuar en función de esa información para mantener sus sistemas fuera del infierno de las dependencias. Usar estos conocimientos para ejercer una administración adecuada de las dependencias es el siguiente paso lógico.

Al ejercer la administración de dependencias, considere actividades como analizar las dependencias para buscar vulnerabilidades, actualizar las dependencias y tomar el control del uso de las dependencias, lo que limita una lista potencialmente larga de componentes interdependientes. A menudo se denominan dependencias conflictivas, circulares y de rombo, y todo resulta en sistemas difíciles de actualizar e inseguros.

Dependabot de GitHub ayuda a impulsar la administración de dependencias adecuada mediante la visualización de las dependencias de cualquier componente dentro de un sistema, incluidas todas las dependencias conectadas. Pero la mejor parte es lo proactiva que puede ser la herramienta. Dependabot notificará a su equipo cuando localice una vulnerabilidad conocida o cuando estén disponibles actualizaciones para una dependencia específica. Incluso ayuda a las prácticas de corrección al preparar y sugerir los cambios necesarios para las actualizaciones en el código base.

Aunque herramientas poderosas, como Dependabot, ayudan a resolver la administración de dependencias, sigue siendo primordial mantener un conjunto claro de directivas. Es imprescindible que los equipos desarrollen una directiva clara para adoptar las dependencias, administrar las actualizaciones y localizar posibles vulnerabilidades. Sin la adopción coherente de directivas entre equipos, la administración de dependencias de la organización se convierte en caos y problemas de seguridad.

Porcentaje de repositorios activos que recibieron alertas de Dependabot[1]



[1]GitHub, Octoverse Security Report, 2020

Aumente la transparencia con una lista de materiales de software (SBOM)

La transparencia aporta confianza en los sistemas. Los equipos que están íntimamente familiarizados con los módulos de software en los que confían desarrollan procedimientos recomendados para las actualizaciones y entienden el impacto que un módulo puede tener en su sistema y en todo el ciclo de vida de la entrega.

Una práctica importante que debe considerar cuando desarrolla la transparencia es crear una SBOM. De manera similar a cómo una lista de materiales de fabricación detalla la construcción de un producto, lo mismo ocurre con una SBOM. Una SBOM describe la construcción del módulo de software y proporciona información para la administración de la seguridad. Además de reforzar la seguridad, una SBOM también puede reducir los riesgos de licencia y cumplimiento.

La estandarización de las SBOM está en marcha. Diversas iniciativas en el sector de la seguridad están intentando solidificar el modelo de lo que es una SBOM estandarizada que pueda leer una máquina. Por ahora, un buen punto de partida es mantener una lista actualizada de componentes, todas las versiones complementarias y las estrategias de actualización, junto con las vulnerabilidades y los mantenedores conocidos. Además, una SBOM completa se considera como un indicador de calidad para un producto de software.

De manera similar a cómo una lista de materiales de fabricación detalla la construcción de un producto, lo mismo ocurre con una SBOM. Una SBOM describe la construcción del módulo de software y proporciona información para la administración de la seguridad.

Consideraciones finales

Al integrar la seguridad en sus prácticas de DevOps, se compromete con la creación y la administración de un equipo de DevSecOps. Al igual que DevOps en las operaciones combinadas y los equipos y prácticas de los desarrolladores anteriores, introducir la seguridad a estos equipos requiere paciencia. Los cambios a gran escala a menudo afectan las capacidades de los equipos para cumplir con los plazos, por lo que los empleados se resistirán a la adopción de DevSecOps si la preparación no es metódica. La adopción correcta de DevSecOps requiere una combinación reflexiva e intencional de colaboración entre equipos, una cultura centrada en la seguridad y tecnología de vanguardia.

Es tentador adoptar una tecnología eficaz sin considerar primero la cultura de la empresa o los procesos existentes. Pero es un error. Crear una cultura centrada en la seguridad es primordial para que DevSecOps tenga repercusión en sus equipos. Considere la posibilidad de crear una comunidad de InnerSource o adoptar un modelo de campeón de seguridad para estimular la colaboración entre equipos y difundir los conocimientos entre equipos.

La adopción correcta de DevSecOps requiere una combinación reflexiva e intencional de colaboración entre equipos, una cultura centrada en la seguridad y tecnología de vanguardia.

Después de preparar a su organización para un cambio cultural, el siguiente desafío es aprovechar la tecnología de vanguardia de manera eficaz. Con una gran cantidad de opciones, es difícil seleccionar la solución perfecta. Empiece por seleccionar herramientas y tecnología que se adopten fácilmente. Céntrese en la tecnología que ofrece valor, ya sea mediante la mejora de la observabilidad, el fortalecimiento de la seguridad en las primeras etapas del desarrollo de aplicaciones o la provisión proactiva de correcciones.

Crear y administrar un equipo de DevSecOps siempre es distinto para cada organización. No todas las recomendaciones se aplican directamente. Está bien explorar formas únicas para complementar con seguridad los procesos empresariales, la cultura y los trabajadores actuales. Use estos consejos como una guía para equipar al equipo DevSecOps recién formado con toda la información y los recursos necesarios para transitar por el cambio.

Adoptar DevSecOps es una ventaja para la entrega de software. Elimine los cuellos de botella que obstruyen su canalización de distribución y proporcione los controles necesarios para el cumplimiento y la seguridad. Al descubrir las vulnerabilidades antes, sus equipos ahorran tiempo para corregir los problemas y lograr el cumplimiento, a la vez que minimizan los costos asociados. Vuelva a lo importante: impulsar la innovación con una entrega de software eficaz y segura.

Al descubrir las vulnerabilidades antes, sus equipos ahorran tiempo para corregir los problemas y lograr el cumplimiento, a la vez que minimizan los costos asociados.

Cómo pueden ayudar Microsoft y Sogeti

Los equipos de DevSecOps tienen éxito con la colaboración entre equipos, el enfoque en la velocidad del desarrollador y las herramientas de vanguardia. Microsoft ofrece recursos de aprendizaje, productos y servicios para orientar a todos los equipos de DevSecOps para la innovación, independientemente del lenguaje, el marco o la nube.

El proveedor de servicios administrados de expertos de Azure, Sogeti, mejora continuamente la entrega digital centrada en la empresa para los equipos de DevSecOps que usan centros de nube y DevOps en todo el mundo. Sogeti usa su Marco de adopción DevSecOps y la biblioteca de CloudBoost para impulsar la mejora continua y la adopción de InnerSource en los equipos de DevSecOps, para así aprovechar las capacidades completas de la plataforma de Azure para la gobernanza, la seguridad y el cumplimiento como una base nativa de la nube.

Comparta esto

Inspire a otros líderes técnicos a integrar la seguridad en sus prácticas de DevOps al compartir este informe técnico en las redes sociales o por correo electrónico.

Recursos

Solución de DevSecOps de Microsoft: Integre la seguridad en todos los aspectos del ciclo de vida de entrega del software. Obtenga información sobre cómo Microsoft ofrece una solución completa para habilitar DevSecOps, o proteger DevOps, para las aplicaciones en la nube (y en cualquier lugar) con Azure y GitHub.

GitHub: Seguridad en cada paso. GitHub ayuda a las empresas a anticiparse a los problemas de seguridad, aprovechar la experiencia de la comunidad de seguridad y usar open source de forma segura.

Microsoft Azure: La seguridad está integrada en todos los aspectos de Azure. Fortalezca su posición de seguridad con Azure. Azure le ofrece ventajas de seguridad únicas derivadas de la inteligencia de seguridad global, los sofisticados controles orientados al cliente y una mayor infraestructura protegida.

Documentación: Aprenda cómo la seguridad de Azure ayuda a proteger sus aplicaciones y datos, apoyar sus esfuerzos de cumplimiento y ofrecer seguridad rentable para organizaciones de todos los tamaños.

Hable con nuestro equipo de ventas: Hable con nuestros especialistas para ver cómo Microsoft puede ayudar a su equipo de DevSecOps.

Sogeti: Estamos trabajando con Microsoft para garantizar que nuestros clientes creen valor a partir de las nuevas herramientas, enfoques y capacidades de la nube de DevOps.

Marco de adopción de DevSecOps de Sogeti: Lleve a cabo lanzamientos más rápido con una arquitectura de referencia empresarial de DevSecOps, descripciones de productos y DevSecOps Blueprints.

Biblioteca de CloudBoost de Sogeti: Automatice el aprovisionamiento del entorno en la plataforma en la nube de Azure con nuestro repositorio de plantillas y scripts reutilizables.

Servicios de OneNative de Sogeti: Habilite el desarrollo dinámico de la nube pública, desde el diseño hasta la compilación y la ejecución.

Autores

Samit Jhaveri es el director de Marketing de Productos de Microsoft Azure que se concentra en el desarrollo de aplicaciones en la nube y DevOps con GitHub. Se desempeña como líder empresarial trabajando en la administración de productos, liderazgo de ventas y finanzas con la responsabilidad de definir y ejecutar la estrategia de comercialización de E2E, que incluye precios y ofertas y planes de ejecución, como campañas e iniciativas de campo y socios para el crecimiento del negocio. Antes de su rol actual, Samit lideró un equipo de ingeniería en la División de servidor y herramientas de Microsoft y fue responsable de enviar varias soluciones B2B para diferentes industrias verticales. Samit obtuvo su MBA de la Universidad de Washington y maestrías en Sistemas de información gerencial de la Universidad de Arizona.

Clemens Reijnen es CTO global de Servicios en la Nube y líder de DevOps de Sogeti. Ha sido galardonado con el premio Microsoft Most Professional durante 10 años consecutivos y es miembro técnico de SogetiLabs. Es coautor del libro "Informe de Enterprise DevOps 2020–2021" con Microsoft y escribe con frecuencia sobre la nube y DevOps en Sogeti.com. Como líder global de DevOps, trabaja estrechamente con los grandes clientes empresariales de Sogeti para garantizar que su adopción de la nube y los programas de transformación de DevOps empresarial creen valor para el negocio.

Coautores de Sogeti:
André Andersen, Gwendal Jabot, Laurent Grangeau, Matt Braafhart, Olivier Dupré, Peter Rombouts, Rahul Sharma, Sandra Parlant y Tony Jarriault.