Si crea, evalúa e impone requisitos de seguridad en su código fuente, puede saber que su aplicación es segura.

Diciembre de 2009

Rational. software

La información es poder

Su software está intentado decirle algo

Ryan Berg
IBM Senior Security Architect
IBM Software Group

Su software está intentado decirle algo. Si escucha con atención, llegará a conocer los riesgos potenciales a los que se enfrenta su empresa. Además de esto, saber es poder:

- **El poder de saber que usted cumple** *normativas como el Estándar de seguridad de datos para la industria de las tarjetas de pago (PCI DSS).*
- **El poder de saber que usted cumple sus promesas** *porque protege los datos personales de sus clientes.*
- **El poder de hacer que sus subcontratistas** *se responsabilicen de los requisitos de seguridad evaluables.*

Su software tiene mucho que decir sobre la privacidad de los datos. Su software es un motor para sus datos. Dentro de él la información se procesa, transforma y transmite. Si sabe interpretar todo lo que le puede decir su software, tendrá un gran poder en sus manos:

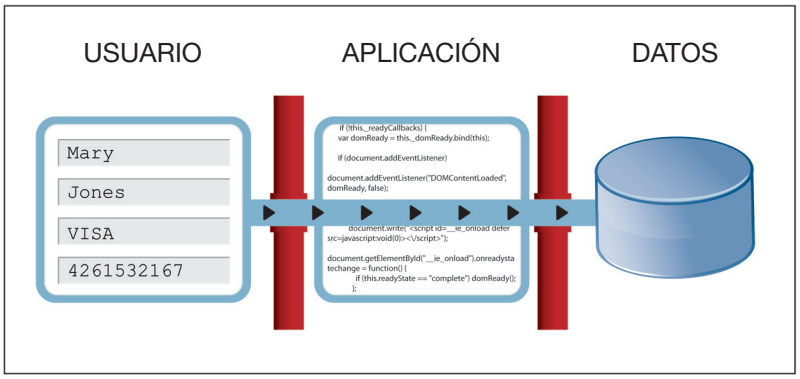
- **El poder de saber que sus datos están lo suficientemente seguros,** *si la implementación de la criptografía, el almacenamiento de datos y al acceso a las bases de datos son seguros.*
- **El poder de distinguir al enemigo del amigo** *para garantizar que solo tengan acceso los autorizados a entrar.*
- **El poder de evitar sorpresas** *al detectar códigos maliciosos.*
- **El poder de imponer las normas** *que usted haya elaborado con el debido cuidado.*

Si escucha a su software podrá saber, no imaginar ni esperar, que su aplicación es lo suficientemente segura. El software se lo dirá.

¿Las aplicaciones protegen sus datos?

Hoy en día, al tomar decisiones sobre las prioridades de seguridad en la tecnología de la información (TI), debe encontrar el justo equilibrio entre el riesgo, el impacto y la probabilidad de incidentes a nivel empresarial y los costes de prevención o limpieza. Tradicionalmente, la variable más analizada de esta ecuación eran los métodos que utilizaban los piratas informáticos para perturbar o invadir el sistema. La seguridad de protección se convirtió en el centro natural de atención, y el nivel de protección se medía calculando la resistencia defensiva frente a ataques reales o simulados. Esta protección ha demostrado ser insuficiente, ya que la intensificación de la frecuencia y del impacto de las intrusiones logradas está demostrando que los activos de TI no están seguros. Es probable que en el contexto siempre cambiante de la capa de infraestructura de la aplicación usted no disponga de la información adecuada sobre dónde y cómo pueden estar expuestos sus datos. Así que, ¿dónde puede dar el siguiente paso para proteger la seguridad de sus activos críticos de información? Como entre el 75% y 90% de todos los ataques en Internet están afectando a la capa de la aplicación, es evidente que ha llegado el momento de que escuche lo que su software intenta decirle sobre la seguridad de los datos.

Las aplicaciones de software son la primera línea de batalla para sus datos. Las aplicaciones de software adquieren, transmiten, almacenan y procesan activos de información. Si usted sabe a qué tiene que prestar atención, sus aplicaciones le pueden ofrecer abundante información sobre sus fortalezas, vulnerabilidades y métodos. Esta es la información que usted proporciona a los reguladores, a los clientes, al jefe y al consejo. El conocimiento procede de la base del código fuente de la aplicación. El conocimiento le da el poder necesario para tomar decisiones fundamentadas en materia de gestión de riesgos.



¿Por qué debería escuchar a su software?

1. Normativas: el poder de saber que cumple

Las violaciones de seguridad generan normativas. Las nuevas normativas, como el PCI DSS, se están erigiendo en estándares de seguridad de la TI de diligencia debida. Las normativas exigen pruebas de que los activos críticos de información estén seguros al nivel de la aplicación. Los intentos previos de normativas impuestas exigían tecnologías o configuraciones; pero los métodos de ataque cambiantes las dejaron atrás rápidamente. El nuevo enfoque centrado en los datos impone la protección de los elementos individuales de la información (como en el caso del registro de la tarjeta de crédito), o de los elementos potencialmente relacionados, que al combinarlos desvelen una identidad. Las normativas se centran en el tratamiento adecuado de estos elementos de datos durante la adquisición, la transferencia, el almacenamiento, el acceso y la destrucción. En consecuencia, el cumplimiento exige un profundo conocimiento del funcionamiento real de la aplicación. Para saber a dónde llegan sus datos, debe conocer con certeza todas las rutas y los puntos terminales. Y esta certeza requiere un análisis del código fuente.

Importante: Respuestas de código fuente a preguntas del PCI

El PCI DSS constituye un excelente estudio de caso del nuevo estándar de diligencia debida en materia de seguridad de la información, ya que se centra fundamentalmente en la seguridad de los datos de las tarjetas de crédito. Para obtener respuestas de cumplimiento fiables se debe inspeccionar el código fuente. El PCI DSS traza exigencias específicas para la protección, la autenticación, la auditoría y el registro de los datos. La tabla incluida a continuación muestra algunos de los requisitos de seguridad del PCI DSS para la información crítica y las aplicaciones:

Sección	Directriz o requisito
3.2	No almacene datos confidenciales de autenticación (ni siquiera si están cifrados).
3.3	Oculte el número de identificación personal (PIN) en el momento de la visualización.
3.4	Haga que los PIN sean ilegibles allí donde se almacenen.
4.1	Utilice una criptografía potente y protocolos de seguridad durante la transmisión en redes abiertas y públicas.
6.3	Desarrolle aplicaciones de software basadas en las mejores prácticas del sector.
6.3.7	Revise el código personalizado antes de permitir la salida a producción.
6.5	Desarrolle todas las aplicaciones web basándose en directrices seguras de codificación.
7.2	Cree un mecanismo que limite el acceso para sistemas con múltiples usuarios.
8.5.16	Autentique todos los accesos a cualquier base de datos que contenga la información sobre titulares de tarjetas.
10.2	Implemente pistas de auditoría automatizadas para todos los componentes del sistema.
10.2.1	Todos los usuarios particulares acceden a los datos de los titulares de tarjetas.
10.3	Registre, como mínimo, las siguientes entradas de pista de auditoría:
10.3.1	Identificación del usuario
10.3.2	Tipo de evento

La información que usted recoge de su código fuente para el cumplimiento se extrae de esta tabla. Si tomamos el requisito de la Sección 3.2 como ejemplo ('No almacene datos confidenciales de autenticación ni siquiera si están cifrados'), la pregunta es evidente. '¿Esta aplicación almacena datos de autenticación?'

- *La única fuente fiable que le facilita esta información es el propio código fuente de la aplicación. Al analizar el código, descubrirá todos los lugares interesantes a los que llega la información de autenticación. A partir de este análisis, puede estar seguro de que la información no se almacena nunca. Lo cierto es que ningún otro lugar puede ofrecer la información necesaria para confirmar el cumplimiento. El software lo sabe y está deseando decírselo.*

El PCI DSS no es el único que aumenta la sofisticación y se centra en el tratamiento seguro de los elementos y servicios de datos. Otras normativas, como la Ley Gramm-Leach-Bliley (GLBA), la Ley de responsabilidad y transferibilidad de seguros médicos (HIPPA) y la Ley de protección de datos del Reino Unido también se centran en la confidencialidad de la información personal identificable. La Ley Sarbanes-Oxley y el Acuerdo de Basilea II reafirman la necesidad de integridad de los datos y los sistemas financieros. Solo las organizaciones y los individuos que han dedicado cierto tiempo a ver qué se hace dentro de la aplicación pueden ofrecer una validación fiable del cumplimiento. Todo lo demás son simples suposiciones.

2. Responsabilidad: el poder de saber que usted cumple sus promesas.

Las declaraciones de privacidad que acompañan a la mayoría de las transacciones en red hacen que los usuarios confíen en que las protecciones implementadas garantizarán la seguridad de su información personal. Sin embargo, en realidad casi nunca se menciona la seguridad al nivel de la aplicación.

Estas declaraciones, creadas para mitigar la inquietud de los usuarios frente a las amenazas en red y los comportamientos empresariales poco honestos, normalmente se interesan por los protocolos de comunicación y las políticas de revelación. En consecuencia, no se mencionan ni se tratan las aplicaciones que son el centro de la experiencia de usuario. Mientras tanto, se aceptan enunciados relativos a la protección y la seguridad de dichos datos. Las promesas de privacidad que usted haga a sus clientes, accionistas y socios sólo se pueden cumplir si el código fuente de su aplicación se evalúa y trata de forma activa.

3. Contratación y adquisición de software: el poder de hacer que sus subcontratistas se responsabilicen.

Es cada vez más frecuente que las empresas desarrollen su actividad utilizando un software que otra empresa ha creado para ellas. Era habitual que una entidad externa automatizara el proceso empresarial sin prestar atención alguna al hecho de que el software recibido fuera o no seguro. Como sucede con todos los requisitos contractuales, los requisitos de seguridad se deben expresar de forma muy clara y el método para evaluar el cumplimiento debe ser preciso.

El código fuente es el único lugar acorde y fiable para recabar esta información. El software habla directamente sobre las cuestiones de los criterios de seguridad suscritos. Esta certeza no se podría lograr mediante una sencilla prueba funcional o una prueba de caja negra. Los requisitos no se detectan mediante pruebas autorizadas de rutinas de validación, ni mediante pruebas de protocolo de seguridad. En consecuencia, el análisis del código fuente constituye el único medio certero para valorar el rendimiento, evaluar el cumplimiento, recuperar costes y calcular daños.

Muchos de los elementos de la aplicación afectan a la seguridad de la información. El análisis del código fuente traduce la gama completa de posibles comportamientos de una aplicación en una presentación que ofrece datos fiables sobre el estado de seguridad de la misma. Si no acceden al código fuente para obtener esta información, las empresas deben confiar o suponer, sin tener pruebas de ello, que su información está segura. Pero ya no se puede vivir con esa incertidumbre.

¿Qué puede decirle su aplicación sobre la seguridad de la información?

1. Privacidad de los datos: el poder de saber que sus datos están lo suficientemente seguros.

A pesar de lo que nos hacen creer, la privacidad de los datos no se logra solamente con una buena seguridad de red y honradez empresarial. Los verdaderos datos sobre la privacidad se encuentran donde se introduce, manipula y almacena la información personal. Éstas son algunas de las áreas de la aplicación que afectan a la privacidad de la información:

Criptografía: Mediante una inspección directa del código fuente, puede saber dónde se está utilizando la criptografía, dónde no se ha utilizado, así como qué tipo y potencia de algoritmo criptográfico se ha aplicado. Si estos datos del código fuente se combinan con el contexto de la función de los datos dentro de la aplicación se obtiene una respuesta directa a las cuestiones de utilización, transmisión y almacenamiento seguro de datos.

Almacenamiento indefinido y temporal de archivos: Cuando las aplicaciones manipulan y transforman datos, esa información se almacena indefinida o temporalmente en el disco o en la memoria. El análisis del código fuente avisa cuando se producen estas actividades potencialmente vulnerables. Si utiliza las mejores prácticas recomendadas, como la limpieza automática y la aleatoriedad de nombre de archivo o de ubicación, puede tomar decisiones fundamentadas sobre la implementación segura de dichas actividades.

Acceso a la base de datos: Las bases de datos suelen ser un punto terminal de datos personales y proporcionar datos seleccionados a los solicitantes. En consecuencia, lo que más afecta a la privacidad de los datos son las metodologías que se utilizan para autenticar, conectar y comunicar con las bases de datos. La coherencia real de estas operaciones (así como los límites impuestos sobre qué es accesible, quién puede acceder y cómo se almacena) se plasma en el código fuente.

Registro y gestión de errores: para los piratas informáticos los fallos dicen mucho. El registro y la gestión de errores son áreas en las que las aplicaciones pueden ser demasiado reveladoras. Por ejemplo, un registro de transacción contiene información que nunca se debería considerar fuera de peligro, sino como parte de la transacción. Un volcado de memoria en un error de aplicación contiene información personal que no se ha borrado del todo de la memoria de la aplicación que ha dado el error. Las rutinas que gestionan estas dos funciones y la coherencia con la que se aplican se pueden identificar en la aplicación para que se puedan reseñar, examinar y validar.

2. Autenticación y autorización: el poder de distinguir al enemigo del amigo.

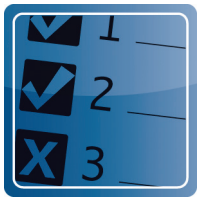
Las aplicaciones se basan en una variedad de métodos de control de acceso y autenticación para garantizar que se da acceso a las personas autorizadas. Una autorización insuficiente y restricciones poco estrictas para el control de acceso permiten accesos no autorizados o la manipulación de datos personales. Por ejemplo, cuando un pirata informático utiliza credenciales que permiten un tipo de acceso con servicios o datos que no verifican lo suficiente si las credenciales son correctas. Como sucedía con el viejo y defectuoso perímetro de seguridad, el exterior es duro y crujiente (autorización para entrar en la aplicación) y el interior es blando y masticable (ahora se puede entrar en cualquier sitio sin comprobaciones adicionales).

Los mecanismos de autenticación, los modelos de control de acceso, así como las diferentes comprobaciones que detectan a los usuarios autenticados y las acciones contra las restricciones del control de acceso se encuentran en el código. Es ahí donde debe buscar la información para proteger sus activos críticos.

3. Código malicioso: el poder de evitar sorpresas desagradables.
El código malicioso, cuando está bien escrito, se parece a cualquier otra sección del código. Una comunicación de red o una rutina de sincronización son componentes habituales de una aplicación.

Solo las condiciones o el contexto de dicha funcionalidad dan la voz de alerta. Hay que saber dónde surgen estos casos potencialmente maliciosos para verificar que no son ni exposiciones accidentales de la funcionalidad (depuración del desarrollador, funcionalidad diagnóstica o funcionalidad vencida de una versión anterior todavía en uso), ni componentes realmente maliciosos insertados para ofrecer una puerta trasera tras el despliegue.

El método más efectivo para identificar funcionalidades inesperadas consiste en examinar el código fuente de una aplicación y en enumerar todas las rutas potenciales. Aunque una prueba funcional activa no detecte un evento malicioso temporal o excepcional, los intentos subyacentes de establecer una transferencia de datos, dañar los datos o crear conexiones inapropiadas se ponen claramente en evidencia. Se compara lo que está haciendo la aplicación con su capacidad esperada, se identifica el código malicioso y se elimina antes de su emisión.



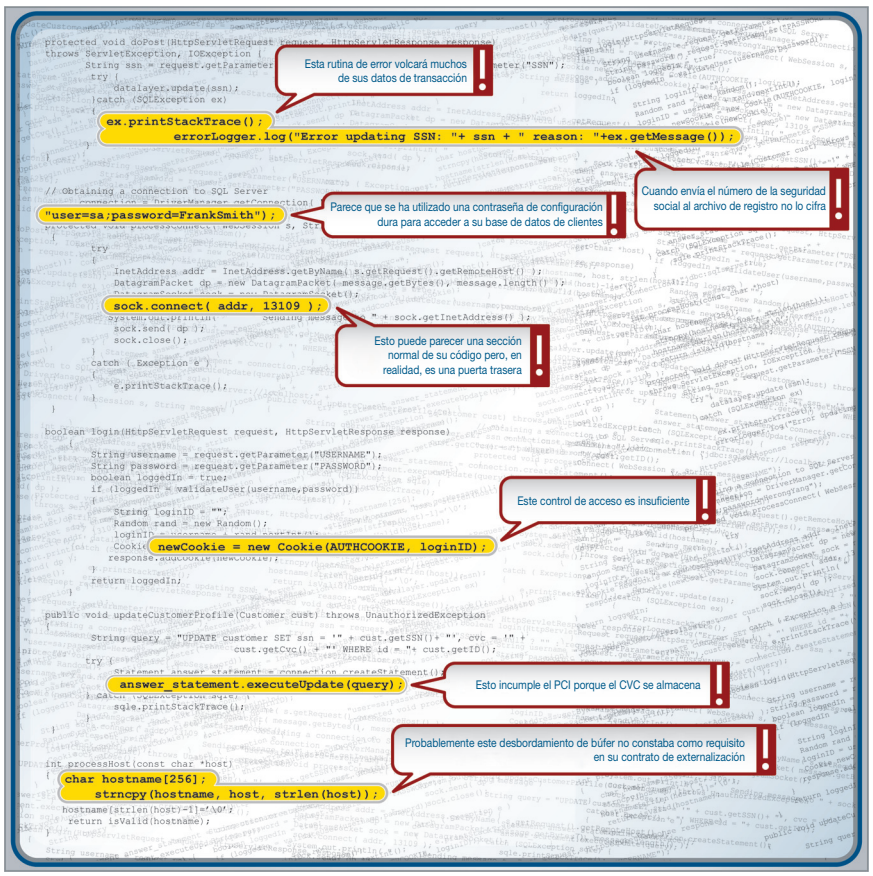
4. Directrices y políticas: el poder de imponer las normas.
Las mejores prácticas del mundo no sirven si no se implementa un proceso para garantizar que se cumplen las directrices. Si impone el uso exclusivo de criptografía potente, prohíbe la validación en línea, o exige el uso de un acceso centralizado a las bases de datos, verifique que estas normas estén en el código. Ya que es ahí donde se puede comprobar e imponer el cumplimiento de prácticas seguras de codificación. Imaginemos que el análisis buscó toda criptografía y encontró versiones más débiles y no autorizadas. Validó o impidió el uso de una funcionalidad específica, como los habituales intentos de registro seguro o las rutinas de acceso a bases de datos. Un análisis del código fuente ofreció una visión de la aplicación completa, así como de todos los intentos y operaciones, que permitió evaluar el cumplimiento de las mejores prácticas internas.

Si una empresa dedica tiempo a definir políticas de seguridad, la capacidad de medir y evaluar el cumplimiento permite multiplicar varias veces la rentabilidad de la inversión (ROI). En la fuente encontrará la información sobre si la aplicación se ha adherido a una política o la prueba de que se necesita más vigilancia.

Estas son algunas de las áreas en las que, mediante un análisis del código fuente, se puede ver claramente la diferencia entre la seguridad de datos deseada y real. Representan los puntos clave del cumplimiento, la responsabilidad y las condiciones contractuales. La fiabilidad del sistema, la posibilidad de auditar la actividad de la aplicación y la capacidad de recurso frente a los subcontratistas que no cumplan los requisitos de seguridad contractuales se interpretan mediante un análisis del código fuente de las aplicaciones públicas de carácter crítico. Si crea, evalúa e impone requisitos de seguridad en la aplicación, podrá saber, no suponer ni esperar, que la aplicación es segura.

Escuche lo que su software le quiere decir:

Las vulnerabilidades que ponen en riesgo sus datos y los mecanismos de seguridad que los protegen están integradas en los millones de líneas de los códigos fuente que dan poder a su organización. Si dispone de la herramienta adecuada para analizar el código fuente, un analista de seguridad ya no tendrá que hacer conjeturas sobre los lugares del software que afectan a la privacidad, a las operaciones y a la integridad de la información. Llegado el caso, su software hablará alto y claro. Este conocimiento adquirido le da poder para tomar las decisiones más efectivas para su empresa en materia de gestión de riesgos.



Para obtener información adicional

Para obtener más información acerca del IBM Rational AppScan Source Edition, póngase en contacto con su representante de marketing o distribuidor comercial de IBM, o bien visite la siguiente página Web: **ibm.com**/software/rational/products/appscan/source/

Ryan Berg es arquitecto de seguridad sénior en IBM. Ryan es un orador, instructor y autor de gran popularidad en el campo de la seguridad, la gestión de riesgos y los procesos de desarrollo seguro. Tiene patentes registradas y algunas otras pendientes en evaluación de seguridad multilingüe, seguridad del kernel, lenguaje intermediario de evaluación de seguridad y protocolos seguros de comunicación remota.

IBM España
Santa Hortensia 26-28
28002
Madrid

TAKE BACK CONTROL WITH Rational.

Reciclable, reciclar por favor

RAW14202-ESES-00