Alerta de seguridad desde el diseño: eliminación de vulnerabilidades de inyección de comandos del OS

Los agentes malintencionados utilizan vulnerabilidades de inyección de comandos de OS para comprometer los sistemas

A pesar del amplio conocimiento y la documentación de las vulnerabilidades de inyección de comandos del OS durante las últimas dos décadas, junto con la disponibilidad de medidas de mitigación efectivas, los fabricantes de software han seguido desarrollando productos con este defecto, que pone a los clientes en riesgo.

Las vulnerabilidades de inyección de comandos del sistema operativo (OS, por sus siglas en inglés) son una clase de vulnerabilidad prevenible en los productos de software. Los fabricantes de software pueden eliminarlas desde la fuente mediante un enfoque de seguridad desde el diseño. A pesar de este hecho, siguen apareciendo vulnerabilidades de inyección de comandos en el sistema operativo, lo que permite a los adversarios explotarlas para causar daños. La CISA y la FBI publican esta Alerta de seguridad desde el diseño en respuesta a recientes campañas de agentes de amenazas muy publicitadas que explotaron defectos de inyección de comandos en el sistema operativo en dispositivos de borde de red (CVE- 2024-20399, CVE-2024-3400, CVE-2024-21887) para atacar y comprometer a los usuarios. Estas vulnerabilidades permitieron que agentes maliciosos no autenticados ejecutaran código de forma remota en dispositivos de borde de la red.

Las vulnerabilidades de inyección de comandos del OS surgen cuando los fabricantes no validan ni desinfectan de forma adecuada la entrada del usuario al construir comandos para ejecutar en el OS subyacente. El diseño y desarrollo de software que confía en la información ingresada por el usuario sin una validación o desinfección adecuadas puede permitir que los agentes de amenazas ejecuten comandos maliciosos, poniendo en riesgo a los clientes.

La CISA y la FBI instan a los directores ejecutivos y otros líderes empresariales de fabricantes de tecnología a que soliciten a sus líderes técnicos que analicen las ocurrencias pasadas de esta clase de defectos y desarrollen un plan para eliminarlos en el futuro.

Para prevenir aún más estas vulnerabilidades, los líderes técnicos deben hacer lo siguiente:

- Asegurarse de que el software utilice funciones que generen comandos de formas más seguras al preservar la sintaxis prevista del comando y sus argumentos.[1]
- Revisar sus modelos de amenazas.
- Utilizar bibliotecas de componentes modernas.
- Realizar revisiones de código.
- Implementar pruebas de producto agresivas y adversas para garantizar la calidad y seguridad de su código durante todo el ciclo de vida del desarrollo.[2]

Lecciones por aprender de Seguridad desde el diseño

Los productos que son seguros desde el diseño protegen de forma razonable contra agentes cibernéticos malintencionados que explotan las clases más comunes y peligrosas de defectos del producto[3]. Incorporar seguridad desde el principio (comenzando en la fase de diseño y continuando durante el desarrollo, el lanzamiento y las actualizaciones) reduce la carga para los clientes y el riesgo para el público. Las vulnerabilidades de inyección de comandos del OS se pueden prevenir desde hace mucho tiempo gracias a la separación clara de la entrada del usuario del contenido de un comando. A pesar de este hallazgo, las vulnerabilidades de inyección de comandos del OS (muchas de las cuales son resultado de CWE-78) siguen siendo una clase predominante de vulnerabilidad. La CISA ha agregado recientemente CVE-2024-20399, CVE-2024-3400, y CVE-2024-21887 al Catálogo de KEV, que documenta vulnerabilidades explotadas en el terreno. **Nota:** CWE-78 es la debilidad infantil de CWE-77 y está relacionada con varias otras debilidades.

[1] Hoja de trucos para la validación de entrada. Serie de hojas de trucos de OWASP. https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html
[2] División de Seguridad Informática, Laboratorio de Tecnología de la Información, Instituto Nacional de Estándares y Tecnología, Departamento de Comercio de EE. UU. (sin fecha). Marco de desarrollo de software seguro | CSRC | CSRC. https://csrc.nist.gov/Projects/ssdf
[3] CWE - 2023 CWE Top 25 most dangerous software weaknesses. (sin fecha). https://cwe.mitre.org/top25/archive/2023/2023_top25_list.html

¿Cómo pueden los fabricantes de software prevenir las vulnerabilidades de inyección de comandos de OS?

Durante el diseño y desarrollo de un producto de software, los desarrolladores deben tomar medidas para evitar vulnerabilidades de inyección de comandos del OS a gran escala, incluidas, entre otras las siguientes:[4]

- Siempre que sea posible, utilice funciones de biblioteca integradas que separen los comandos de sus argumentos en lugar de construir cadenas sin formato que se introducen en un comando de sistema de propósito general.
- Utilice la parametrización de entrada para mantener los datos separados de los comandos; valide y desinfecte toda la entrada proporcionada por el usuario.
- Limite las partes de los comandos construidas por la entrada del usuario únicamente a lo que sea necesario.

Ejemplo: en Python, un desarrollador que quiera crear una carpeta dada la entrada de un usuario debe utilizar la función `os.mkdir()` en lugar de invocar un comando.[5] Si dicha función de biblioteca incorporada no está disponible, el desarrollador debe invocar el comando de una manera que separe los argumentos del comando del comando del OS en sí, por ejemplo, desinfectando la entrada del usuario y luego invocando `subprocess.run(["mkdir", user_input])`. Los fabricantes de software pueden hacer cumplir este proceso mediante la escritura de reglas al momento de las solicitudes de extracción para no permitir la invocación de comandos de riesgo, como prohibir `os.system`, `subprocess.run` cuando se suministra con una cadena (en lugar de una matriz) y `subprocess.run` cuando se suministra con el argumento `shell=True`.[6]

Principios a seguir de Seguridad desde el diseño

La CISA y la FBI animan a los fabricantes a aprender cómo proteger sus productos para que no sean víctimas de vulnerabilidades de inyección de comandos de OS y otras actividades maliciosas evitables mediante la revisión de los tres principios establecidos en la guía conjunta Cambiar el equilibrio del riesgo de la ciberseguridad: principios y enfoques para un software seguro desde el diseño.

Principio 1: Asumir los resultados del cliente en materia de seguridad

Los fabricantes de software deberían responsabilizarse de los resultados de seguridad de sus clientes con la eliminación de las vulnerabilidades de inyección de comandos del OS de sus productos. Hay áreas de seguridad clave en las que los fabricantes deberían invertir para proteger a sus clientes y al público. Esto incluye proporcionar bloques de construcción seguros para sus desarrolladores para garantizar que un solo error no comprometa los datos de millones de usuarios. El ciclo de detección, mitigación e implementación de parches para vulnerabilidades que se conocen desde hace años no es un enfoque duradero de la seguridad. Existen mecanismos eficaces para prevenir clases de vulnerabilidades a gran escala y los fabricantes de software deberían implementarlos lo antes posible en el ciclo de desarrollo. La adopción de prácticas recomendadas estándar, como las pautas mencionadas anteriormente, puede ayudar a los fabricantes a erradicar las vulnerabilidades de inyección de comando de OS en la fuente, en lugar de depender de los clientes para aplicar las correcciones. Los fabricantes también deberían implementar mecanismos automatizados que eviten que su software utilice funciones inseguras.

Además, los altos ejecutivos de los fabricantes de software deben asumir la responsabilidad de la seguridad de sus clientes, empezando por realizar pruebas periódicas y revisiones de código para determinar la susceptibilidad del producto a la explotación. El Proyecto de Seguridad de Aplicaciones Web Abiertas (OWASP, por sus siglas en inglés) y otras entidades brindan orientación sobre métodos de prueba con las técnicas disponibles.

---

[4] OS Command Injection Defense - OWASP Cheat Sheet Series. (Sin fecha). https://cheatsheetseries.owasp.org/cheatsheets/OS_Command_Injection_Defense_Cheat_Sheet.html#defense-option-3-parameterization-in-conjunction-with-input-validation; A03 Injection - OWASP Top 10:2021. (Sin fecha).

[5] Asegúrese de que los argumentos de os.mkdir() no creen una vulnerabilidad de salto de directorio. Consulte la alerta anterior de Seguro desde el diseño: https://www.cisa.gov/resources-tools/resources/secure-design-alert-eliminating-directory-traversal-vulnerabilities-software.

[6] Command injection prevention for Python | Semgrep. (12 de abril de 2023). https://semgrep.dev/docs/cheat-sheets/python-command-injection.

Principio 2: Adoptar métodos radicales de transparencia y rendición de cuentas

Los fabricantes deben actuar con transparencia al revelar las vulnerabilidades de sus productos. Para tal fin, los fabricantes deberían rastrear las vulnerabilidades asociadas a sus productos y revelarlas a sus clientes a través del programa CVE. Los fabricantes deben asegurarse de que sus registros de CVE sean correctos y completos. Además de proporcionar CVE, es especialmente importante que los fabricantes proporcionen un CWE preciso para que la industria pueda rastrear las clases de defectos de software, y los clientes puedan comprender las áreas en las que las prácticas de desarrollo de un determinado proveedor pueden requerir mejoras.[7]

Muchas, pero no todas, las vulnerabilidades de inyección de comandos del OS son el resultado de CWE-78. Como tal, los fabricantes deben identificar y documentar las causas fundamentales de las vulnerabilidades de inyección de comandos de OS y declarar como objetivo comercial trabajar para eliminar esa clase al completo. Los fabricantes de software también deberían mantener un programa de divulgación de vulnerabilidades (VDP, por sus siglas en inglés) moderno. **Nota:** la CISA proporciona recursos para ayudar a las organizaciones a establecer y mantener un VDP.

Principio 3: Construir estructura organizativa y liderazgo para lograr estos objetivos

Los ejecutivos de fabricación de tecnología deben hacer lo siguiente:

- Dar a la seguridad de sus productos el mismo nivel de cuidado que dan al costo.
- Considerar el panorama completo: que los clientes, nuestra economía y nuestra seguridad nacional actualmente soportan el peso de las decisiones comerciales de no incorporar seguridad a sus productos, como lo refleja con claridad las campañas de agentes de amenazas recientes descrita en esta Alerta.
- Tener en cuenta que la implementación total del desarrollo de software seguro desde el diseño puede reducir los costos financieros y de productividad, así como la complejidad.
- Realizar las inversiones apropiadas y desarrollar las estructuras de incentivos correctas que promuevan la seguridad como un objetivo comercial declarado.
- Liderar programas para erradicar clases enteras de vulnerabilidad en lugar de abordarlas caso por caso.
- Establecer estructuras organizativas que prioricen medidas proactivas, como la adopción de prácticas recomendadas estándar, para erradicar las vulnerabilidades de inyección de comandos de OS en la fuente.
- Asegurarse de que su organización realice revisiones para detectar vulnerabilidades comunes y conocidas, como inyección de comandos de OS, para determinar su susceptibilidad e implementar las medidas de mitigación existentes, efectivas y documentadas.
    - Las organizaciones deben realizar estas revisiones de forma continua para erradicar las clases de vulnerabilidad, ya que algunas vulnerabilidades pueden cambiar o desarrollarse con el tiempo.
    - Los ejecutivos deben solicitar actualizaciones periódicas para evaluar: (1) el progreso de la empresa en la identificación de clases recurrentes de vulnerabilidad, (2) el progreso de la empresa para eliminarlas y (3) los recursos apropiados necesarios para seguir progresando.

Punto de acción para los fabricantes de software

Para demostrar su compromiso de desarrollar productos seguros desde el diseño, los fabricantes de software deberían asumir el Compromiso de seguridad desde el diseño. El compromiso establece siete objetivos clave que los firmantes se comprometen a lograr con la demostración de un progreso cuantificable, incluida la reducción de clases sistémicas de vulnerabilidad como la inyección de comandos del OS.

Esta Alerta de seguridad desde el diseño es parte de una serie en curso que tiene como objetivo promover las prácticas recomendadas en toda la industria que eliminen clases enteras de vulnerabilidad durante las fases de diseño y desarrollo del ciclo de vida del desarrollo del producto. A través de la iniciativa Seguro desde el diseño, buscamos fomentar un cambio cultural en toda la industria al normalizar el desarrollo de productos tecnológicos que sean seguros de usar desde el primer momento. Visite cisa.gov para obtener más información sobre los principios de Seguro desde el diseño, asumir el Compromiso de seguro desde el diseño y mantenerse informado sobre las últimas alertas de Seguro desde el diseño.

[7] La clasificación de enumeración de debilidades comunes (CWE, por sus siglas en inglés) identifica clases de debilidades de software y hardware (incluidas vulnerabilidades y defectos); la clasificación de vulnerabilidades y exposiciones comunes (CVE, por sus siglas en inglés) identifica y etiqueta vulnerabilidades únicas en productos de software y hardware específicos.

Descargo de responsabilidad

La información contenida en este informe se proporciona “tal cual” solo con fines informativos. Las organizaciones autoras no respaldan ninguna entidad comercial, producto, empresa ni servicio, incluidas las entidades, los productos o los servicios vinculados en este documento. Cualquier referencia a entidades comerciales específicas, productos, procesos o servicios mediante marcas de servicio, marcas comerciales, fabricantes o de otro modo, no constituye ni implica respaldo, recomendación o favoritismo por parte de las organizaciones autoras.