Serverless Computing ("Computación sin servidor")

DOI: 10.29236/sistemas.n168a6

Reflexiones básicas sobre seguridad y control.

Resumen
La computación sin servidor (CSS) es parte de la evolución de la computación en la nube. Este nuevo paradigma de computación establece oportunidades relevantes para las organizaciones de cara a su estrategia digital para concretar iniciativas ágiles y eficientes que den cuenta de las experiencias que exigen sus clientes en la actualidad. Sin perjuicio de lo anterior, la CSS implica un alto acoplamiento e interacción de sus diferentes componentes lo que aumenta la complejidad de su ejecución y, por tanto, los riesgos de seguridad que pueden terminar comprometiendo tanto la infraestructura, como los datos de las personas. En este sentido, este artículo hace una revisión básica de las oportunidades y retos de seguridad y control de la CSS como una primera aproximación conceptual para mantener en foco no sólo sus ventajas, sino disuadir al adversario desde la seguridad de la nube y en la nube para hacer más resistentes las propuestas digitales en este nuevo ecosistema tecnológico.

Palabras claves
Computación en la nube, Computación sin servidor, Función como servicio, Multitenencia, Ecosistema tecnológico

Introducción

La acelerada transformación digital de las organizaciones, las exigencias de mayores y mejores experiencias que demandan los clientes y un aumento exponencial de ataques cibernéticos, establece el contexto actual donde las organizaciones deben adelantar sus iniciativas digitales, comoquiera que, es allí donde se concretan las nuevas apuestas y promesas de valor que permiten crear escenarios inéditos para desarrollar ideas de negocio novedosas y disruptivas que cambian la vista tradicional de lo que hacen las empresas en la actualidad (Woerner et al., 2022).

Para hacerlo, se requieren condiciones de agilidad y despliegue eficiente de soluciones, lo que implica acelerar los ciclos de desarrollo y utilizar marcos de trabajo que integren diferentes componentes de forma modular y funcional para contar con las aplicaciones esperadas, con las exigencias particulares de los clientes y sobremanera atendiendo los retos del negocio y los desafíos de cumplimiento propio de los diferentes sectores de las empresas.

En línea con lo anterior, el paradigma de "computación sin servidor" (CSS) (en inglés *serverless computing*) aparece como la apuesta clave de las organizaciones para concretar un uso más eficiente de las soluciones en la nube, con el fin de acelerar las iniciativas digitales y crear ventajas competitivas que transformen rápidamente la manera de hacer las cosas. La CSS ofrece muchas ventajas sobre la arquitectura tradicional basada en la nube: menores costos operativos, procesos simplificados, mayor flexibilidad y escalabilidad, y despliegue más rápido (Chichioco, 2019).

Si bien estas ventajas se concretan en las iniciativas digitales que transforman las organizaciones, igualmente generan riesgos cibernéticos relevantes que deben ser analizados y atendidos, para efectos de decidir el nivel de apetito de este riesgo y las capacidades que las organizaciones deben desarrollar para ajustarse a esta decisión y así, apalancar no sólo los despliegues de las aplicaciones de forma ágil y eficiente, sino igualmente confiables y menos inseguras con el fin de generar confianza digital en sus clientes.

En este sentido, este artículo hace una breve revisión de los fundamentos de la CSS, así como los retos de seguridad y control propio de este nuevo paradigma, que no sólo actualiza la lectura de las prácticas de seguridad en la computación en la nube, sino que explora los temas particulares de aseguramiento y endurecimiento de aplicaciones en la modalidad de "*Function as Service*" (Función como servicio).

"Computación sin servidor". Evolución de la computación en la nube

La CSS es parte de la evolución de la computación en la nube. Es un ejercicio por hacer aún más transparente tanto para los clientes como para los desarrolladores aquello que se usa y cómo se usa sin importar dónde se encuentre. En este contexto, la computación en la nube marca una transformación de la computación corporativa creando un mundo basado en pago por consumo, ubicación compartida o exclusiva y creación de múltiples máquinas virtuales según necesidad, que cambia la lectura de los grandes centros de datos empresariales, los retos de las actualizaciones de software base y los desafíos de la obsolescencia tecnológica ahora todo en manos de un tercero de confianza.

El primer cambio hacia la computación en la nube se da en función de la infraestructura tecnológica, lo que se ha denominado "*Infrastructure as a Service*" (IaaS) (Infraestructura como servicio). Una estrategia que implica la simplificación de la administración haciendo más fácil la configuración y gestión de la infraestructura (*hardware*), particularmente a través de servidores virtuales y redes de centros datos masivos de múltiple tenencia. En este escenario, el proveedor se hace responsable y custodio de la infraestructura de almacenamiento, procesamiento y redes con el fin de asegurar un entorno de computación confiable para las empresas y sus necesidades (Schleier-Smith et al., 2021).

La segunda transición se refleja en el ocultamiento de los servidores y creación de entornos de programación para desarrolladores que la literatura denomina "*Platform as a Service*" (PaaS) y "*Software as a Service*" (SaaS). Si en la primera etapa fue el hardware, en esta segunda son las máquinas y el software como marcos de trabajo simplificados que permite a los desarrolladores avanzar rápidamente en los despliegues de aplicaciones y productos especializados, para dar cumplimiento con los retos que tienen las organizaciones mejorando del "time to market" (tiempo de puesta en operación). Esto es, los desarrolladores no tienen por qué preocuparse por actualizar el sistema operativo, las herramientas de desarrollo y ni mantener el hardware (Schleier-Smith et al., 2021).

En un tercer momento de la evolución ahora se ocultan los servidores y la programación de las aplicaciones y su operación lo que se ha denominado "*Serverless computing*" (Computación sin servidor). Actualmente, la CSS se presenta en dos versiones diferentes, conocidas como backend como servicio (BaaS) y función como servicio (FaaS). La idea central de BaaS es proporcionar a los desarrolladores de software un conjunto de servicios y herramientas (por ejemplo,

bases de datos, API (*Application Program Interface*), almacenamiento de archivos o notificaciones push) para facilitar y acelerar el desarrollo de aplicaciones móviles y web. En cuanto a FaaS, se centra en permitir a los desarrolladores de software desplegar y ejecutar sus propias funciones en la nube (sin perjuicio que que las funciones también pueden utilizar servicios adicionales como los ofrecidos en BaaS) (Marin et al., 2022).

Una plataforma "sin servidores" se compone al menos de cinco componentes: (Marin et al., 2022)

- Funciones - Las funciones suelen ejecutarse dentro de un entorno de ejecución aislado y recién generado (por ejemplo, un contenedor) dentro de un nodo operacional.
- Pasarelas API – Puerta de enlace para que las aplicaciones accedan a los datos, lógica empresarial o funcionalidad requerida por los clientes.
- Servicios en la nube (compartidos) – Integración de diferentes gamas de servicios como recopilar varios tipos de datos para reaccionar rápidamente a eventos en las plataformas, gestión del ciclo de vida de aplicaciones, habilitar capacidades DevOps o almacenamiento de corto o largo plazo.
- Herramientas de seguridad - Conjunto de herramientas y servicios para facilitar la gestión de la seguridad de los flujos de trabajo. Algunas de ellas IAM (Identity and Access Management), que permite configurar controles de acceso detallados para autenticar y restringir los recursos a los que tienen acceso las funciones y VPC (Virtual Private Cloud), que permite crear redes privadas y aisladas para comunicaciones seguras entre aplicaciones que pertenecen a la misma organización.
- Plano de control - Componente de monitorización que se utiliza para comprobar periódicamente el estado de los nodos del sistema de trabajo, el software que ejecutan y los entornos de ejecución que se implementan en ellos.

"Computación sin servidor". Adversarios y amenazas de seguridad

Entender los retos de seguridad y control de la CSS implica reconocer dos tipos de adversario: los externos y los internos, los cuales tienen perspectivas distintas que es importante reconocer y analizar para concretar un mejor aseguramiento de este tipo de computación.

Los *adversarios externos* suelen llevar a cabo sus ataques desde fuera de la nube aprovechando los campos de entrada controlados por el usuario en cualquiera de las APIs existentes que se ofrecen para gestionar eventos. Estos ataques pueden permitir a los atacantes ejecutar comandos arbitrarios dentro de la función con el fin de recuperar

datos confidenciales (por ejemplo, tokens de sesión almacenados en tablas de entorno) o manipular la ejecución de cualquier función (o servicio en la nube que reciba datos de entrada maliciosamente elaborados y no aplique técnicas adecuadas de aseguramiento de datos de entrada) (Marin et al., 2022).

De otra parte, los *adversarios internos* pueden tomar el control total de una (o más) funciones y realizar ataques desde el interior de la nube. En el caso de las nubes públicas, es relativamente sencillo para estos adversarios desplegar funciones maliciosas para intentar realizar ataques desde dentro. Estos adversarios pueden intentar: (Marin et al., 2022).

- crear canales encubiertos,
- realizar ataques de escalada de privilegios (por ejemplo, para comprometer otras funciones co-residentes o nodos trabajadores),
- recuperar o manipular datos sensibles (por ejemplo, datos en servicios de almacenamiento),
- recopilar conocimiento sobre entornos de ejecución e infraestructura,
- llevar a cabo varios tipos de ataques de denegación de servicio.

Como modelo de servicio en la nube multiusuario, la CSS es susceptible de amenazas a la seguridad, que considerando los dos tipos de adversarios previamente comentados, pueden dividirse en cinco (5) categorías en función del lugar desde el que se lanzan: (Li et al., 2023)

1. Ataques externos a las aplicaciones por parte de usuarios malintencionados, como ataques de cross-site scripting y ataques de inyección.
2. Ataques internos a las aplicaciones por parte de intrusos malintencionados, como el acceso interno ilegal y los ataques de sniffing. Los adversarios de la red interna pueden incluso percibir información sensible a partir del patrón de comunicación y el nivel de actividad de las funciones.
3. Las tres categorías restantes son ataques horizonta*les entre inquilinos*:
   i. ataques de canal lateral,
   ii. ataques verticales a infraestructuras sin servidor desde inquilinos maliciosos, y
   iii. ataques verticales a aplicaciones desde plataformas maliciosas.

"Computación sin servidor". Retos de seguridad y control

De acuerdo con un reciente estudio realizado por Radware (2022), se tiene que el 95% de las organizaciones utilizan al menos dos tipos de infraestructura, y casi la mitad de las organizaciones despliegan aplicaciones en cuatro o más plataformas diferentes. Como resultado, el despliegue de aplicaciones en entornos multi-nube y de nube híbrida se ha convertido en la nueva normalidad.

En este contexto, los retos de seguridad y control se vuelven relevantes no sólo para las aplicaciones que actualmente se ejecutan en estos entornos, sino para la dinámica de la organización y sus desafíos de negocio, que demandan mayor agilidad y mejores experiencias para los clientes (Sharaf, 2020), y así concretar la confianza digital. En este sentido, se plantean al menos cinco desafíos para la seguridad en la CSS como son: (Li et al., 2023)

- aislamiento de recursos – La CSS se basa en la liberación y reutilización dinámicas de software y hardware para mejorar la utilización de recursos y reducir costos. Por lo tanto, existe un fuerte requisito de aislamiento de recursos, especialmente entre múltiples inquilinos.
- monitoreo de la seguridad - La naturaleza efímera de las funciones reduce la superficie de ataque, también estrecha significativamente la ventana para que los desarrolladores descubran y diagnostiquen los problemas.
- controles de seguridad - Los fronteras fragmentadas de las aplicaciones sin servidor aumentan la dificultad del control de la seguridad. La naturaleza distribuida de las funciones agrava aún más este reto.
- protección de datos - Ya sea durante la ejecución de código en un entorno no controlado o durante el procesamiento de datos sensibles a través de los servicios de la plataforma, pueden surgir problemas de privacidad y cumplimiento.
- investigaciones forenses - Uno de los principales retos de la investigación forense en la nube es el hecho de que los datos se almacenan a menudo en servidores situados en múltiples ubicaciones geográficas, con múltiples proveedores y modelos compartidos, lo que puede dificultar la identificación y recopilación de los datos pertinentes.

En consecuencia, cuatro elementos claves son relevantes para efectos de avanzar frente a los riesgos enumerados en secciones anteriores: (Sharaf, 2020)

- Permisos de gestión - reducir la probabilidad de que un funcionamiento independiente contenga una amenaza de seguridad oculta, por ejemplo la ejecución de un malware.
- Perímetro de seguridad - reducir el riesgo de múltiples puntos de vulnerabilidad introducidos por la aplicación sin servidor, asociados con cada función.
- Dependencia de terceros – reducir la dependencia de paquetes de terceros, reduce la complejidad de implementar la seguridad perimetral a cada función.
- Cifrado doble y fragmentación de datos - En esta técnica, reduce el acceso no autorizado y la fuga de información combinando dos técnicas de cifrado diferentes para cifrar los datos y la

fragmentación de datos para dividirlos en varias partes y distribuirlos por redes/servidores (Kumari et al, 2022).

La seguridad en la CSS exige al igual que en la computación en la nube que el aseguramiento propio de los terceros sea parte fundamental del servicio que se presta, lo que implica mantener en foco en vulnerabilidades como: (Parast et al., 2022)

- Ataques DDoS – Impedir acceso legítimo a los recursos deseados.
- Ataques a máquinas virtuales – Control de máquinas virtuales aprovechando vulnerabilidades del hipervisor.
- Secuestro de sesión – Acceso a credenciales de los usuarios y robo de identidad.
- Multitenencia – Coexistencia de múltiples usuarios en una instancia de recursos físicos de máquinas virtuales.
- Amenaza interna del proveedor – Empleado del proveedor accede a datos sensibles y utiliza la información de forma no autorizada.
- Fuga de información – Configuraciones y accesos inadecuados aumentan la filtración de datos en entorno de nube.

Reflexiones finales

Si bien la computación en la nube se ha venido convirtiendo en una estrategia natural de las organizaciones para hacer frente a las demanda de flexibilidad, elasticidad, agilidad y despliegue efectivo y eficiente de nuevas aplicaciones y estrategias digitales de las empresas, la computación sin servidor, como parte de la evolución de la nube, establece un nuevo paradigma que requiere ser revisado y analizado en profundidad para capitalizar las oportunidades que ofrece y atender los retos que esta forma de computación revela tanto a programadores como a los proveedores (O'Meara & Lennon, 2020).

La computación sin servidor abre un espacio de construcción abierto y funcional que requiere un conocimiento en detalle de las funciones y su contexto en el cual se despliegan, por tanto la configuración de los ecosistemas tecnológicos donde residen debe responder no sólo a condiciones óptimas y específicas para el procesamiento, almacenamiento y enrutamiento de la ejecución de las aplicaciones, sino igualmente a consideraciones de seguridad y control que generen confianza digital en cada una de las relaciones e interacciones de sus compontes (Kumari et al., 2022).

Así las cosas, la CSS deberá estar alineada en dos dimensiones: seguridad *de la* nube y seguridad *en la* nube. La *seguridad de la nube* relacionada con la responsabilidad de los proveedores de la nube y las medidas establecidas para mantener la infraestructura subyacente y los servicios en la nube resistentes a las acciones no autorizadas de los adversarios. Y por otro lado, la

seguridad en la nube asociada con la responsabilidad de los desarrolladores de software. Esto es, los mecanismos de seguridad empleados para: prevenir vulnerabilidades en las funciones, proteger los datos de la aplicación (almacenados en servicios en la nube) y, asegurar la totalidad de los flujos de trabajo (por ejemplo, garantizando que todas las funciones se ejecuten con los privilegios mínimos necesarios) (Marin et al., 2022).

Finalmente, la CSS busca ampliar la capacidad de la computación en la nube para permitir a las empresas centrarse únicamente en el código y pagar únicamente por el uso de la potencia de computación, lo que necesariamente aumenta los riesgos inherentes a su configuración debido a la complejidad que surge en el despliegue de servicios en la nube en función de la demanda (Sharaf, 2020). La CSS requiere una asignación dinámica de recursos tecnológicos lo que implica un marco de seguridad y control igualmente dinámico y flexible, un reto que aún está por resolver y que demanda un enfoque diferencial que repiense la postura de seguridad de las arquitecturas sin servidores y en ecosistemas multinube.

Referencias

Chichioco, A. (2019). Going Serverless? Common Serverless Security Issues and Best Practices. Hakin9. https://hakin9.org/going-serverless-common-serverless-security-issues-and-best-practices/

Kumari, S., Solanki, K., Dalal, S. & Dhankhar, A. (2022). Analysis Of Cloud Computing Security Threats and Countermeasures. 2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India. pp. 1-6, doi: 10.1109/ICRITO56286.2022.9964632.

Li, X., Leng, X. & Chen, Y. (2023). Securing Serverless Computing: Challenges, Solutions, and Opportunities. IEEE Network, 37(2). 166-173. doi: 10.1109/MNET.005.2100335.

Marin, E., Perino, D. & Di Pietro, R. (2022). Serverless computing: a security perspective. Journal of Cloud Computing. 11(69). https://doi.org/10.1186/s13677-022-00347-w

O'Meara, W. & Lennon, R. G. (2020). Serverless Computing Security: Protecting Application Logic. 2020 31st Irish Signals and Systems Conference (ISSC), Letterkenny, Ireland. 1-5. doi: 10.1109/ISSC49989.2020.9180214.

Parast, F. K., Sindhav, C., Nikam, S., Yekta, H. I., Kent, K. B. & Hakak, S. (2022). Cloud computing security: A survey of service-based models. Computers & Security. 114. https://doi.org/10.1016/j.cose.2021.102580

Radware (2022). Application Security In A Multi-Cloud World. https://www.radware.com/multi-cloud-report-2022/

Schleier-Smith et al. (2021). What serverless computing is and should become: the next phase of cloud computing. Communications of the ACM. 64(5). Doi: 10.1145/3406011

Sharaf, S. (2020). Security Issues in Serverless Computing Architecture. Inter-

national Journal of Emerging Trends in Engineering Research. 8(2). https://doi.org/10.30534/ijeter/2020/43822020

Woerner, S., Weill, P. & Sebastian, I. (2022). Future ready. The four pathways to capturing digital value. Boston, MA. USA: Harvard Business Review Press

Jeimy J. Cano M., Ph.D, CFE, CICA. *Ingeniero y Magíster en Ingeniería de Sistemas y Computación, Universidad de los Andes. Especialista en Derecho Disciplinario, Universidad Externado de Colombia; Ph.D en Business Administration, Newport University, CA. USA. y Ph.D en Educación, Universidad Santo Tomás. Profesional certificado como Certified Fraud Examiner (CFE), por la Association of Certified Fraud Examiners y Certified Internal Control Auditor (CICA) por The Institute of Internal Controls. Profesor Distinguido de la Facultad de Derecho, Universidad de los Andes. Es director de la Revista SISTEMAS de la Asociación Colombiana de Ingenieros de Sistemas –ACIS–.*