Google destaca la función esencial de los sanitizadores Clang en fortalecer la seguridad de la banda base celular en el sistema operativo Android y prevenir tipos específicos de vulnerabilidades.

Esto incluye el Sanitizador de Desbordamiento de Enteros (IntSan) y el Sanitizador de Límites (BoundSan), ambos integrados en el Sanitizador de Comportamiento No Definido (UBSan), una herramienta diseñada para detectar diversas formas de comportamiento no definido durante la ejecución del programa.

> Ivan Lozano y Roger Piqueras Jover señalaron en una publicación del martes que *«Estos sanitizadores son independientes de la arquitectura, aptos para despliegue en hardware básico, y se deben habilitar en códigos existentes en C/C++ para contrarrestar vulnerabilidades desconocidas».*

Este avance surge meses después de que la compañía tecnológica anunciara colaboraciones con socios del ecosistema para mejorar la seguridad del firmware que interactúa con Android, dificultando así que actores malintencionados logren ejecución remota de código en el SoC de Wi-Fi o la banda base celular.

IntSan y BoundSan son dos de los sanitizadores basados en el compilador que Google ha implementado como medida de mitigación de exploits, detectando desbordamientos aritméticos y realizando verificaciones de límites alrededor de los accesos a matrices, respectivamente.

Aunque Google reconoce que tanto BoundSan como IntSan generan una sobrecarga de rendimiento considerable, los ha activado en áreas críticas para la seguridad antes de una implementación completa en todo el código. Estas áreas incluyen:

- Funciones que analizan mensajes transmitidos por el aire en redes 2G, 3G, 4G y 5G.
- Bibliotecas que codifican/decodifican formatos complejos (p. ej., ASN.1, XML, DNS, etc.).

- Pilas IMS, TCP e IP.
- Funciones de mensajería (SMS, MMS).

> *«En el caso específico de 2G, la estrategia óptima implica desactivar por completo la pila mediante el soporte del 'interruptor 2G' de Android. A pesar de ello, 2G sigue siendo una tecnología de acceso móvil esencial en ciertas regiones y algunos usuarios podrían necesitar habilitar este protocolo heredado»,* mencionaron los investigadores.

Es importante destacar que, aunque los sanitizadores brindan beneficios «*tangibles*«, no abordan otras categorías de vulnerabilidades, como aquellas relacionadas con la seguridad de la memoria, lo que hace necesario un cambio del código a un lenguaje seguro para la memoria, como Rust.

A principios de octubre de 2023, Google informó que había reescrito el firmware de la Máquina Virtual de Android (AVF) en Rust para establecer una base segura para la raíz de confianza de la pVM.

> *«A medida que el sistema operativo de alto nivel se convierte en un objetivo más desafiante para los atacantes, anticipamos que los componentes de nivel inferior, como la banda base, atraerán más atención. Al emplear herramientas y tecnologías de mitigación de exploits modernas, también se puede elevar el nivel de dificultad para atacar la banda base»,* concluyeron los investigadores.