Resumen

La disponibilidad de nuevos procesadores más potentes para aplicaciones empotradas ha permitido el desarrollo de aplicaciones que abordan problemas de mayor complejidad. Debido a esto, las aplicaciones empotradas actualmente tienen más funciones y prestaciones, y como consecuencia de esto, una mayor complejidad. Por este motivo, existe un interés creciente en permitir la ejecución de múltiples aplicaciones de forma segura y sin interferencias en un mismo procesador y memoria. En este marco surgen las arquitecturas de sistemas particionados basados en hipervisores como una solución apropiada para construir sistemas seguros.
Uno de los principales retos en la construcción de sistemas particionados, es la verificación del correcto funcionamiento del hipervisor, dado que es el componente crítico sobre el que descansa la seguridad de todo el sistema particionado. Las técnicas tradicionales de V&V, como testing, inspección y análisis, presentan limitaciones para la verificación exhaustiva del comportamiento del sistema, debido a que el espacio de entradas a verificar crece de forma exponencial con respecto al número de entradas a verificar. Ante estas limitaciones las técnicas de verificación basadas en métodos formales surgen como una alternativa para completar las técnicas de validación tradicional.

Esta disertación se centra en la aplicación de métodos formales para validar la corrección del sistema particionado, en especial del hipervisor XtratuM. La validación de la metodología se realiza aplicando las técnicas propuestas a la validación del hipervisor. Para ello, se propone un modelo formal del hipervisor basado en máquinas de autómatas finitos, este modelo formal permite la definición de las propiedades que el diseño hipervisor debe cumplir para asegurar su corrección. Adicionalmente, esta disertación analiza cómo asegurar la corrección funcional de la implementación del hipervisor por medio de técnicas de verificación deductiva de código.

Por último, se estudian las vulnerabilidades de tipo *information leak* (CWE-200 [CWE08b]) debidas a la perdida de la confidencialidad de la información manejada en el sistema particionado. En este ámbito se modelan las vulnerabilidades, se aplican técnicas de análisis de código para la detección de vulnerabilidades en base al modelo definido y por último se valida la técnica propuesta por medio de un caso práctico sobre el núcleo del sistema operativo Linux que forma parte del sistema particionado.