Virtualization-based security and fault tolerance

Laboratory of Cryptography and System Security
Budapest University of Technology and Economics
Department of Telecommunications
www.crysys.hu

AUSTRIAN INSTITUTE OF TECHNOLOGY

Motivation

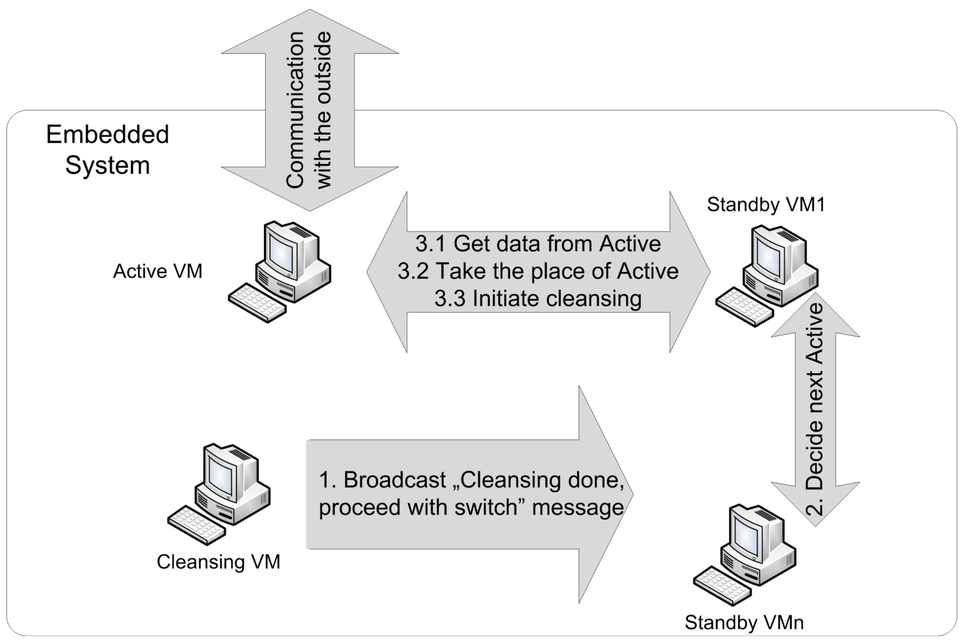
Safety-critical systems have been traditionally designed to resist against random failures, but recently security of such systems has become an equally important non-functional requirement. Besides preventing or detecting intelligent attacks, the secured system must satisfy all safety critical requirements too. In addition, there are non-trivial interferences between safety and security requirements. Redundancy is often used for ensuring reliability and resisting random failures. What if redundancy can also be used for securing safety-critical systems?

Our approach

As an approach for safety & security co-design of safety-critical multi-core systems, we investigate the possibility of using redundant virtual machines (VM) to realize security and fault tolerance.

Multiple virtual machines (VMs) switching between active, standby and cleansing roles

- **Active** VM - there is always one VM providing services
- **Standby** VMs - there are several VMs, in which one of them is ready to become the next Active VM
- **Cleansing** VM - the VM that steps down from the Active state, it is then restored to a clean state to remove any potential malware infection



Technical challenges

How to switch VMs?

- Switching of VMs should be transparent to outside entities and services.
- An active VM must handover all functions to the next active VM, once it steps down from the active state. The handover should meet all real-time requirements.

How to propagate data and application between VMs?

- No malicious content including malware should be propagated to other VMs.
- The active VM must not make any changes to data that has already been transmitted to the VM becoming active.

How to ensure that applications running in VMs are aware of the switch?

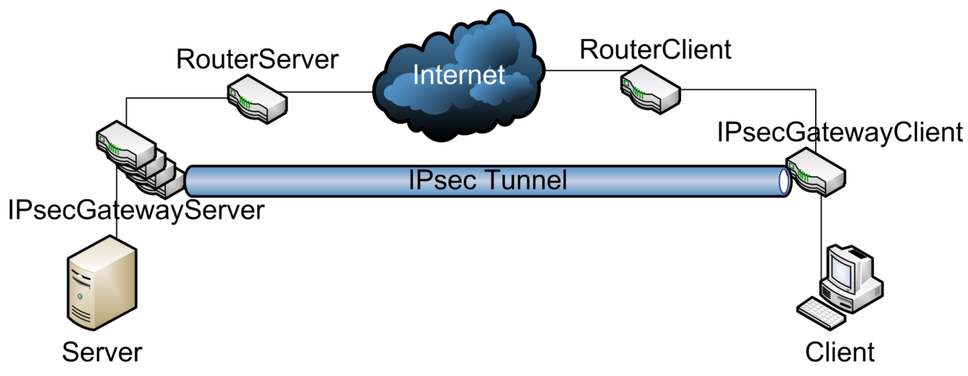
- Applications may need adaptation or extension to work in this paradigm.

Proof-of-concept

Use case: 2 VPN endpoints establish an IPsec tunnel as a secure communication link. *IPsecGatewayClient* is the virtualization-based secure and fault-tolerant VMs composed of 4 VMs. It interacts with *IPsecGatewayServer* at the other end of the IPsec tunnel.

Implementation: Ubuntu 14.04 Server LTS (Trusty Tahr) with OpenS/WAN 2.6.38 using the NETKEY IPsec protocol stack (shipped with the OS)

- Additional dependencies: ipsec-tools 0.8.0



REFERENCES

1. M. T. Jones, “Virtualization for embedded systems” http://www.ibm.com/developerworks/library/l-embedded-virtualization/ (2011)
2. A.K. Bangalore, and A.K. Sood, “Securing Web Servers Using Self Cleansing Intrusion Tolerance (SCIT)” DEPEND ’09 (2009).
3. S. Kent, and K. Seo, “Security Architecture for the Internet Protocol”, RFC 4301 (2005)
4. OpenS/WAN https://www.openswan.org/

Contact

Dorottya Papp dorottya.papp.fl@ait.ac.at
Zhendong Ma zhendong.ma@ait.ac.at
Levente Buttyán buttyan@crysys.hu