JOAN: shepherd application privacy with virtualized special purpose memory

Mingyuan Xia (student), Miao Yu (student), Zhengwei Qi, Haibing Guan
Shanghai Jiao Tong University
{kenmark, superymk, qizhwei, hbguan} @ sjtu.edu.cn

1. MOTIVATION

Networked applications are ubiquitous in home, enterprise, commercial and government. Most of these applications are designed to handle a certain portion of sensitive data, such as various request information in centralized networks and the peer information in p2p networks. The privacy of networked applications faces risks from its own robustness, operating system that runs beneath them and peers that they exchange sensitive data with. The expanded TCB suggests that they will not achieve adequate privacy assurance.

Existing virtualization-based security systems do not address the problem of program vulnerability [1, 3] or confine the programming flexibility of sensitive code [2]. To fully employ the strong isolation of virtualization to protect program data secrecy, user should be able to assign programming friendly protection policy to configurable TCB.

2. DESIGN

To ameliorate the problem, we propose a virtualization-based system JOAN, which utilizes memory virtualization to expose two privacy aware memory primitives. Both memories exhibit double view according to different code context. User can gather sensitive code (via compiler directives) in *Sealed memory*, which is protected from modifications done by untrusted code. *Exchange memory* cryptographically isolates its content so that only sensitive code can view the plain text sensitive data and untrusted code can make verbatim copy of the encrypted data. Figure 1 demonstrate the TCB and main ideas of JOAN. On high level, the *sealed memory* ensures the sensitive code integrity and reduce the TCB size, while the *exchange memory* protects the integrity and privacy of sensitive data. With both virtualized memories, privacy-critical applications can maintain small TCB size and achieve flexible data exchange semantics. Our contributions can be concluded as follows:

- We reduce the TCB to include only user-selected sensitive code and the hypervisor, which is a few order of magnitude smaller than operating system components. Sensitive code is protected by virtualization-enforced memory primitives from illegal modifications from untrusted code.

- We exploit memory virtualization to provide privacy aware memory primitives. Privacy-critical can exploit these programming friendly memory primitives to fully integrate with our system while enjoy flexible and secure sensitive data exchange.

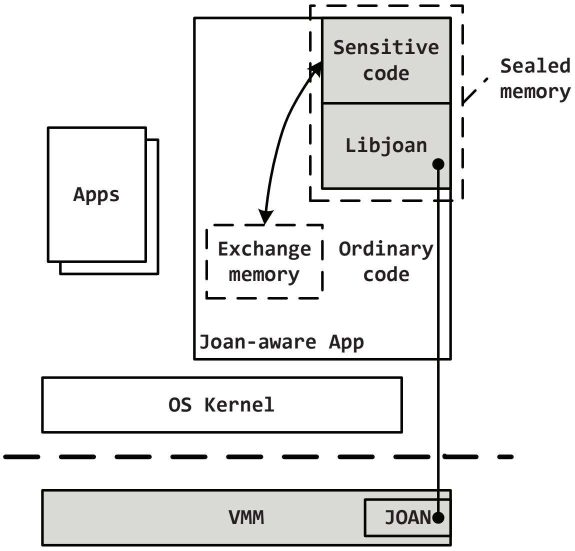

Figure 1: Architecture of JOAN. The grey parts are the TCB, which excludes OS kernel and untrustworthy device drivers. JOAN-aware applications receive protection from sealed memory exposed by JOAN. Exchange memory bridges a flexible yet secure channel to interchange sensitive data.

- We deploy our system in off-shelf hardware environment and enhance the privacy of various practical networked applications.

3. REFERENCES

[1] Xiaoxin Chen, Tal Garfinkel, E. Christopher Lewis, Pratap Subrahmanyam, Carl A. Waldspurger, Dan Boneh, Jeffrey S. Dwoskin, and Dan R. K. Ports. Overshadow: a virtualization-based approach to retrofitting protection in commodity operating systems. In *ASPLOS*, pages 2–13, 2008.

[2] Jonathan M. McCune, Yanlin Li, Ning Qu, Zongwei Zhou, Anupam Datta, Virgil Gligor, and Adrian Perrig. Trustvisor: Efficient tcb reduction and attestation. In *IEEE Symposium on Security and Privacy*, 2010.

[3] Richard Ta-Min, Lionel Litty, and David Lie. Splitting interfaces: Making trust between applications and operating systems configurable. In *OSDI*, pages 279–292, 2006.