MULTILEVEL WORKSTATION

HIGH-SECURITY FRAMEWORK, PILOT, AND FORMALIZATION ARCHITECTURE

Ultra High-Security Design for Multilevel Computing
Advanced Component-Based Security Architecture
Built on a Separation Kernel
Formal Methods Applied to Trusted Components to
Enhance Software Assurance
Versatile Client
- Multiple security domain access from a single device
- Domain isolation via advanced IPsec architecture
- Web browser and remote desktop support for Web and cloud services
- Support for Microsoft RDP, VNC, NX, X11, and SSH protocols
- Isolated public browsing network session

Flexible Administration
- Configuration via SINA PKI and management infrastructure
- Two-factor smart card authentication and configuration
- Independent security domain administration by separate organizations

Major Cost Savings
- Reduce cost of user access devices and network infrastructure per security domain
- Will run on COTS PCs, laptops, and servers
- Straightforward integration of new applications and protocols
- Minimal delta certification and re-evaluation efforts on system changes
Ultra High Security via Component-based Architecture

- We decompose our systems into a set of isolated components with well-defined information flows assuring that only some components have to be trusted.
- Critical trusted components are formally verified for correctness.
- Untrusted components do not impact security architecture and are easily adaptable.
- We use a Separation Kernel to enforce component integrity and information flow between components.
- In contrast, monolithic operating systems are unsuitable to isolate trusted and untrusted components.

Monolithic architecture:
Isolation in monolithic systems is insufficient, leading to a large trusted code base. Errors or weaknesses in the trusted code base may affect system security.

Component-based architecture:
Small independent trustworthy components ensure the confidentiality, integrity, and authenticity of user session data. The Separation Kernel protects the integrity and information flow between trusted components.

Formal Verification as a Foundation for Trustworthy Components

- Testing is insufficient as it cannot show absence of errors.
- Formal Methods are an integral part of our development process.
- The SPARK programming language in conjunction with the interactive theorem prover Isabelle enables us to exclude important error classes like buffer and integer overflows.
- Where required, we show conformance of source code to a formal specification.
- As trusted components are small by design, formal methods are feasible and provide practical results.

For details see: www.secunet.com/multilevel
Unprecedented Security …

… Confidential and stateless multi domain user access

… Security architecture that exclusively depends on trustworthy components

… Security properties of trustworthy components are formally proven

… Very small trusted code base

… Protected by a Separation Kernel

… Runs on a COTS platform

Contact
Will Keegan
Security Software Specialist
LynuxWorks, Inc.
Phone: +1 832-800-4044
wkeegan@lnxw.com

secunet Security Networks AG
Kronprinzenstraße 30
45128 Essen, Germany

 Phone: +49 - 201- 54 54 - 0
 Fax: +49 - 201- 54 54 -1000
 E-mail: info@secunet.com

Contact
Dr. Kai Martius
Head of Business Unit High Security
Phone: +49 201 54 54-3504
kai.martius@secunet.com

Alexander Senier
Principal System Security Architect
Phone: +49 201 54 54-3544
alexander.senier@secunet.com