Microkernel Architecture and Security
1. Introduction
2. Microkernel Architecture
4. Microkernel Architecture and Security
5. Security Focused Operating Systems
6. Some CVEs
7. Summary
Introduction
Exploring the Relationship between Microkernel Architecture and Security

Goal:
• Overview of fundamental security guarantees
• Discuss additional security considerations
Kernel

• Core component of an OS
• Manages system resources
• Bridges hardware and software interaction

• Kernel design categories:
  • Monolithic kernels
  • Microkernels
  • Hybrid kernels

Monolithic kernel design:

Source: https://medium.com/
Kernel

Monolithic components

Monolithic kernel OS

Static deployment

Dynamic deployment

Fine-grained components

Microkernel OS

Source: Martin Děcký, Microkernel-based and Capability-based Operating Systems
Microkernel Architecture
### Basic Principles

<table>
<thead>
<tr>
<th>Split of mechanism and policy</th>
<th>Component-based</th>
<th>Separation of concerns</th>
<th>Least privilege</th>
<th>Modularity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernel contains just the basic and fundamental mechanisms</td>
<td>System composed of isolated components</td>
<td>Each component focuses on a well-defined functionality</td>
<td>Components have minimal privileges</td>
<td>Replacing component implementations</td>
</tr>
</tbody>
</table>
Monolithic Kernel vs. Microkernel

Monolithic components vs. Fine-grained components

Static deployment vs. Dynamic deployment

Monolithic kernel OS

Microkernel OS

Source: Martin Děcký, *Microkernel-based and Capability-based Operating Systems*
Monolithic Kernel vs. Microkernel

Monolithic kernel
- Application
- System call
- IPC, File system
- Virtual memory
- Device drivers
- Hardware

Microkernel
- Application
- Application IPC
- UNIX server
- Device drivers
- File server

Kernel mode

Basic IPC, Virtual memory, Scheduling
Hardware

Source: https://www.baeldung.com
## Monolithic Kernel vs. Microkernel

<table>
<thead>
<tr>
<th>Monolithic kernel</th>
<th>Microkernel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same address space</td>
<td>Separate address spaces</td>
</tr>
<tr>
<td>Configurability via compile-time options and parametrization</td>
<td>Configurability via different use (policy in user space)</td>
</tr>
<tr>
<td>Modularity via run-time dynamic linking</td>
<td>Modularity via extension in user space</td>
</tr>
<tr>
<td>Tight module coupling</td>
<td>Loose module coupling</td>
</tr>
<tr>
<td>OS is easier to implement</td>
<td>OS is more complex to implement</td>
</tr>
<tr>
<td>TCB is larger in size</td>
<td>TCB is smaller in size</td>
</tr>
<tr>
<td>If one component fails, the entire system crashes</td>
<td>If one component fails, it doesn't affect the working of the microkernel</td>
</tr>
</tbody>
</table>
Microkernel OS Types

- Static deployment
- Dynamic deployment

Monolithic kernel OS
- Monolithic components
- Static deployment
- Dynamic deployment

Microkernel OS
- Fine-grained components
- Static deployment
- Dynamic deployment

Monolithic single server OS
- Monolithic components
- Dynamic deployment

Microkernel multiserver OS
- Fine-grained components
- Static deployment
- Dynamic deployment

Source: Martin Děcký, Microkernel-based and Capability-based Operating Systems
Microkernel OS Types

Single-Server Microkernel OS

- Application
- Application
- Application
- User mgmt
- Network stack
- System server
- Memory mgmt
- IPC
- Microkernel
- Hardware

Multiserver Microkernel OS

- Application
- Application
- Application
- Network stack
- Security server
- Location server
- Memory mgmt
- IPC
- Microkernel
- Hardware

Source: Martin Děcký, Microkernel-based and Capability-based Operating Systems
Security Aspects of Operating Systems
OS Security

= Ensuring: **Confidentiality, Integrity, Availability** (CIA)

= Aims to protect everything within the system

= Protects system resources including CPU, memory, disk, programs, and data
Importance

- Flaws lead to vulnerabilities in software
- Protecting sensitive data, ensuring integrity and user privacy
- Essential against cyber threats
- Lots of money has been lost
Who are the Attackers?

For example:

• Hackers *driven by the challenge*
• Insiders seeking *revenge* or gain *informal benefits*
• Criminals seeking *financial* gain
• Terrorist groups or nation states trying to *influence national policy*
• Agents seeking information for *economic, political purposes*
What are the Vulnerabilities?

Security vulnerabilities that affect the operating system:

• Automatically running active content
• Open ports
• Incorrect configuration
• Backdoor
• Unencrypted communication
• Limited resources
• Vulnerabilities in software
### Security Goals

<table>
<thead>
<tr>
<th>Goal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication</td>
<td>Verifying the identity of users or systems</td>
</tr>
<tr>
<td>Authorization</td>
<td>Granting or denying access to resources</td>
</tr>
<tr>
<td>Data confidentiality</td>
<td>Ensuring that sensitive information is protected</td>
</tr>
<tr>
<td>Integrity</td>
<td>Ensuring the accuracy and reliability</td>
</tr>
<tr>
<td>Availability</td>
<td>Ensuring consistently accessibility</td>
</tr>
</tbody>
</table>

**How can this be implemented?**
How to ensure Operating System Security?

- Built-in security
- Additional security measures
Built-in Security

Hardware access control:
• Operating system regulates hardware access for processes
• Prevents one process from compromising another's security

Control of operating system services:
• Monitoring and control of services
• For example: file systems, memory management, and interprocess communication
• Control over system calls:
  • Processes access system services through system calls
  • OS monitors system calls, determining process authorization
Additional Security Measures

Further security improvement by:

• Configuring security settings
• Regular updates and patches
• Installing security software
• Security auditing and monitoring
• Supply chain management
• User training and awareness
Microkernel Architecture and Security
“Operating-system structure has a strong effect on security. 96% of critical Linux exploits would not reach critical severity in a microkernel-based system, 57% would be reduced to low severity.”

“From the security point of view, the monolithic OS design is flawed.”

Source: https://trustworthy.systems
Security Aspects

- Only necessary permissions
- Secure inter-process communication
- Minimizes attack surface
- Reduce impact of security vulnerabilities
Additional Security Measures

- Only necessary permissions
- Secure inter-process communication
- Minimizes attack surface
- Reduce impact of security vulnerabilities
- Secure implementation
- Secure service design
- Security monitoring
seL4 Microkernel

Secure embedded L4:
• Provides security guarantees at OS and application levels
• Only the kernel operates in privileged mode
• Focus on formal verification

Basis:
• For various OS and runtime environments
• For example: Genode OS Framework

Source: http://www.microkernel.info/
Genode

• Tool kit for building operating systems
• Open-ended framework
• Microkernel architecture

Integration of seL4 in Genode:
• Genode provides a platform on which various microkernels can be run
  • Including seL4
• Using the security features and formal verification of seL4
• Retaining flexibility and modularity of Genode framework
Genode can be operated on various **microkernels**

For example:

- seL4 Microkernel
- NOVA Microhypervisor
- Muen Separation Kernel
- In-House Development
More Operating Systems

- Escape
- Fuchsia
- MINIX 3
- L4Re
- Helen OS

Source: http://www.microkernel.info/
“The operating system L4Re Secure Separation Kernel has been approved by the German Federal Office for Information Security (BSI) for the processing of classified information up to classification level German GEHEIM”

Source: https://www.kernkonzept.com
Security Focused Operating Systems
Secure OS’s

Two types of secure OS’s:
  • Security-focused OS
    • Implements measures like sandboxing, compartmentalization, and cryptographic isolation
    • Examples: Qubes OS
  
  • Security-evaluated OS
    • Certified by security-auditing organizations
    • Examples: SUSE Linux, Windows 10 Enterprise
Qubes OS

• Free and open-source operating system
• Based on Xen Hypervisor
• Uses virtual machines to run applications in separate environments
• Rely on the isolation for protection

• Designed with a focus on security:
  • Implements secure components called qubes
  • Efficient isolation of tasks and applications
  • Minimizing the impact of vulnerabilities
Qubes OS and Microkernel

• Qubes OS is not microkernel-based
• BUT...

Source: https://github.com
Some CVEs
Problem:

• Security vulnerability in the OZWPAN driver
• Error: Integer signedness error → negative result from subtraction
• Threatened the security of Linux systems:
  • Denial of service
  • Execution of arbitrary code with kernel privileges

Mitigation by microkernel:

• Driver runs as a server at user level in a separate address space
• Isolation from the kernel prevents direct access to its memory
CVE-2014-9803

Problem:
• Security vulnerability on certain Nexus devices
• Error: Incorrect handling of execute-only pages
• Threatened the security of Linux systems:
  • Allowing an application to gain kernel privileges

Mitigation by microkernel + formal verification:
• This operation must occur in kernel mode ➔ possible in microkernel
• Formal verification ensures the correctness of the microkernel's implementation ➔ not possible in formally verified microkernel
Problem:

• Security vulnerability in the ext4_journal_stop function
• Error: Unauthorized access to a specific error field
• Threatened the security of Linux systems:
  • Full file system disclosure or a kernel crash
  • Posing significant risks to system integrity and data security

Partial mitigation by microkernel architecture:

• File system is implemented as a user-level server
  • No kernel crash, as the file system operates independently
  • Still allow to gain access to files, compromising data confidentiality
Summary
Summary

- OS security crucial for OS and its applications
- Security-focused OS’s demonstrate robust security through microkernel
- Microkernel architecture can boost security
- Microkernel are not the key for general security
Sources
Sources

• Microkernel
  • https://learning.oreilly.com/library/view/operating-system-design/9781439881118/chapter-08.html
  • http://www.microkernel.info/
  • Martin Děcký, Microkernel-based and Capability-based Operating Systems
Sources

• OS & Security
  • https://www.techopedia.com/definition/24774/operating-system-security-os-security
  • https://ics.uci.edu/~goodrich/teach/cs201P/notes/Ch03-OSSec.pdf
  • http://ndl.ethernet.edu.et/bitstream/123456789/87933/8/Chapter%20%207.pdf
  • https://www.tutorialspoint.com/operating_system/pdf/os_security.pdf
  • https://www.researchgate.net/publication/372803341_Secure_Operatin_g_System
Sources

- Microkernel & Security
  - https://www.researchgate.net/publication/241623590_Measures_to_improve_security_in_a_microkernel_operating_system
  - https://sel4.systems/
  - https://genode.org/
  - https://l4re.org/
Sources

• Security Focused Operating Systems
  • https://en.wikipedia.org/wiki/Security-focused_operating_system
  • https://www.stationx.net/secure-operating-systems/
  • https://www.qubes-os.org/

• CVE
  • https://blogs.blackberry.com/en/2020/09/study-confirms-that-microkernel-is-inherently-more-secure