A Taxonomy for Attack Patterns on Information Flows in Component-Based Operating Systems

Michael Hanspach[1] Jörg Keller[2]

[1]Fraunhofer FKIE, Wachtberg, Germany

[2]University Hagen, Germany

7th Layered Assurance Workshop, December 2013

Outline

1. Introduction
2. Basic Scenario
3. Information Flows
4. Attack Patterns
5. Conclusions

Introduction

- component-based operating systems threatened by illegitimate information flows (e.g. the MILS architectural approach)
- attack patterns based on shared resources, colluding components or physical characteristics
- confidentiality and integrity targeted by analysis

Contribution

1. simple algebra describing information flows in operating systems
2. taxonomy of attack patterns on these information flows (all possible information flows targeted)
3. identification of a new covert channel category: covert physical channel

Operating system model

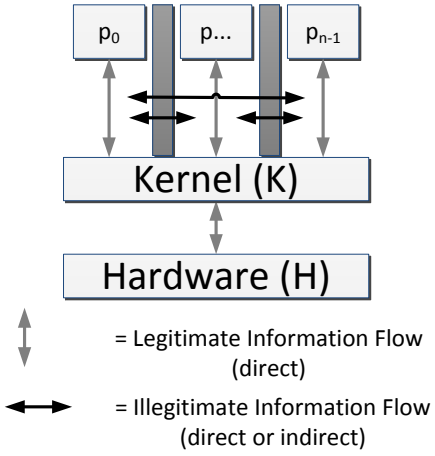
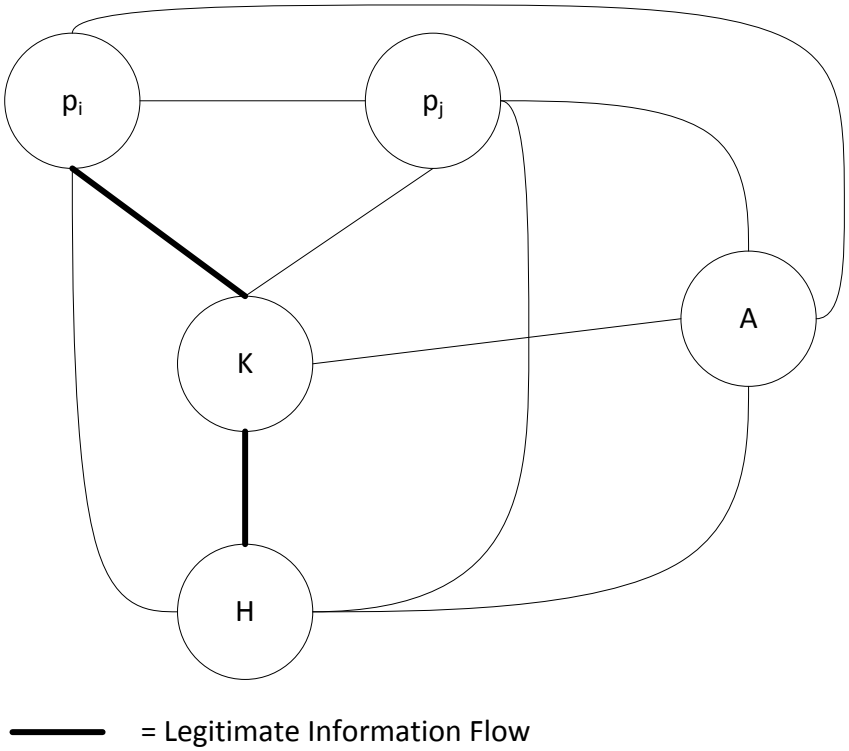


Figure: Scenario for attacks on a component-based operating system

- $n \geq 2$ isolated application partitions
- virtual machines or isolated applications
- pure isolation hypervisors are targeted

Graph of information flows

- $P = \{p_0, \ldots, p_{n-1}\}$ represents isolated application partitions
- K represent operating system kernel
- H represent underlying hardware.
- A is the attacker



Legitimate Information Flows

$\Longleftrightarrow$ is defined as bidirectional direct information flow

1. $p_i \Longleftrightarrow K$
2. $K \Longleftrightarrow H$

   may be concatenated to:
3. $p_i \Longleftrightarrow K \Longleftrightarrow H$

Recursive Walk Algorithm

```
function walk-to-neighbors (start, walk)

  for all neighbors of start do

      local-walk := concatenate(walk, neighbor)
      walk-legitimate := test-walk(start, neighbor)

      if (walk-legitimate = true)
         then
            walk-to-neighbors(neighbor, local-walk)
         else
            store local-walk
      end-if

  end-for

end-function
```

Recursive Walk Algorithm

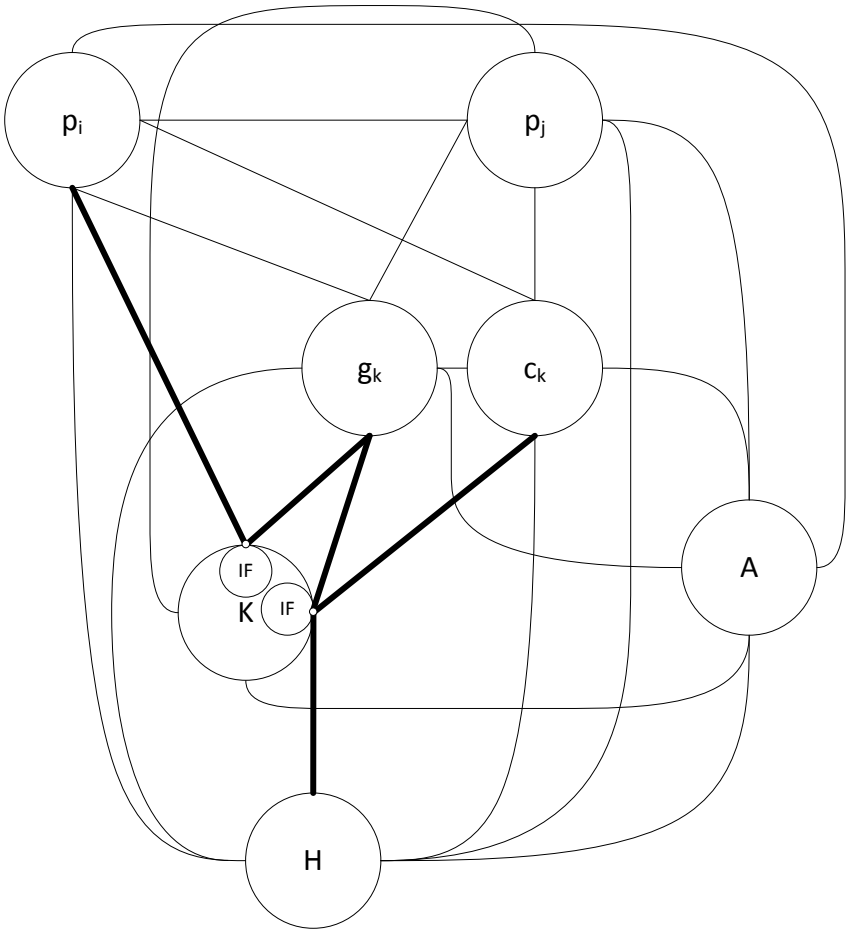
- start at p_i
- algorithm stops if information flow identified as illegitimate
- illegitimate information flows can be extended
- append $\Longleftrightarrow$ p_j to demonstrate channel between p_i and p_j
- append $\Longleftrightarrow$ A to demonstrate an attack
- algorithm verified with Perl program

Illegitimate Information Flows (Basic Scenario)

1. $p_i \Longleftrightarrow A$
2. $p_i \Longleftrightarrow p_j$
3. $p_i \Longleftrightarrow H$
4. $p_i \Longleftrightarrow K \Longleftrightarrow A$
5. $p_i \Longleftrightarrow K \Longleftrightarrow p_j$
6. $p_i \Longleftrightarrow K \Longleftrightarrow H \Longleftrightarrow p_j$
7. $p_i \Longleftrightarrow K \Longleftrightarrow H \Longleftrightarrow A$

Scenario with operating system guards

- using a trustworthy middleware (e.g. the MILS architectural approach)



Legitimate Information Flows (Extended Scenario)

1. $c_k \Longleftrightarrow K \Longleftrightarrow H$
2. $g_k \Longleftrightarrow K \Longleftrightarrow c_k$
3. $p_i \Longleftrightarrow K \Longleftrightarrow g_k$
4. $g_k \Longleftrightarrow K \Longleftrightarrow H$

concatenated information flows:

5. $p_i \Longleftrightarrow K \Longleftrightarrow g_k \Longleftrightarrow K \Longleftrightarrow c_k \Longleftrightarrow K \Longleftrightarrow H$
6. $p_i \Longleftrightarrow K \Longleftrightarrow g_k \Longleftrightarrow K \Longleftrightarrow H$
   illegitimate access to a component:
7. $p_i \Longleftrightarrow K \Longleftrightarrow c_k \Longleftrightarrow K \Longleftrightarrow p_j$

Attack Patterns

- information flows mapped to attack patterns
- map information flows to attack patterns
- attack patterns: specific methods for establishment of illegitimate information flows
- information flows:attack patterns as a many-to-many-relation
- multiple patterns mapped to a single information flow
- multiple information flows mapped to a single attack pattern

Attack patterns mapped to information flows

No.	Information Flow	Basic Attack Patterns
1	$p_i \Longleftrightarrow A$	Use insider attack
2	$p_i \Longleftrightarrow p_j$	Exploit flaws in the system policy
3	$p_i \Longleftrightarrow H$	Use physical attacks, Exploit covert channels, Exploit storage channels
4	$p_i \Longleftrightarrow K \Longleftrightarrow A$	Exploit flaws in the system policy
5	$p_i \Longleftrightarrow K \Longleftrightarrow p_j$	Exploit covert channels, Exploit flaws in the system policy
6	$p_i \Longleftrightarrow K \Longleftrightarrow H \Longleftrightarrow p_j$	Use physical attacks, Exploit covert channels
7	$p_i \Longleftrightarrow K \Longleftrightarrow H \Longleftrightarrow A$	Use physical attacks
8	$p_i \Longleftrightarrow K \Longleftrightarrow c_k \Longleftrightarrow K \Longleftrightarrow p_j$	Exploit flaws in the system policy

Table: Illegitimate information flows mapped to basic attack patterns

Basic and extended attack patterns

No.	Basic Attack Pattern	Extended Attack Patterns
1	Use insider attack	N/A
2	Use physical attacks	Exploit physical access, Exploit side channels
3	Exploit implementation flaws	N/A
4	Exploit covert channels	Exploit covert storage channels, Exploit covert timing channels, Exploit covert physical channels
5	Exploit storage channels	N/A
6	Exploit flaws in the system policy	Exploit corrupt policy channels, Exploit illegitimate access to components

Table: Basic attack patterns mapped to extended attack patterns

Attack patterns and references

No.	Attack Pattern	Description	References
1	Use insider attack	A is granted direct access to extract information from p_i.	Baracaldo and Joshi, Liu et al., Yu and Chiueh.
2	Exploit physical access	A has physical access to the computing system and can directly extract information, which has been stored at H, from p_i.	Halderman et al.
3	Exploit side channel	A uses side channel attacks to extract information from p_i over H.	van Eck, Dürmuth, Backes et al., LeMay and Tan, Shamir and Tromer, Halevi and Saxena.
4	Exploit implementation flaws	A uses implementation flaws of p_i, K or H to extract information from p_i	Klein et al., Boettcher et al., Robinson et al.
5	Exploit covert storage channel	A covert channel is established between p_i and p_j by storing a hidden message within shared resources.	Lampson, National Computer Security Center.
6	Exploit covert timing channel	A covert channel is established between p_i and p_j by encoding a message via the timing behavior of shared resources.	Lampson, National Computer Security Center.
7	Exploit covert physical channel	A covert physical channel is established between p_i and p_j by encoding a message, sending it out into the physical environment via H and receiving it at a different interface of H.	N/A.
8	Exploit storage channels	Information can be exchanged between p_i and p_j via shared storage resources.	Lampson
9	Exploit corrupt policy channel	Information can be exchanged directly between p_i and p_j due to a flawed MAC policy.	Agreiter, Zhai et al.
10	Exploit illegitimate access to components	Information can be exchanged between p_i and p_j by exploiting illegitimate access to a component.	N/A

Table: Description and references regarding to attack patterns

A new type of covert channel

- concept of a *covert physical channel*
- uses physical signals
- transmits covert messages between isolated application partitions
- not a CPC:covert storage channel based on disk arm optimization (Karger and Wray)
- introducing a variable E (describing the shared physical environment)
- disk arm channel described by: $p_i \Longleftrightarrow K \Longleftrightarrow H \Longleftrightarrow K \Longleftrightarrow p_j$
- covert physical channel described by: $p_i \Longleftrightarrow K \Longleftrightarrow H \Longleftrightarrow E \Longleftrightarrow H \Longleftrightarrow K \Longleftrightarrow p_j$
- not a CPC:covert channel based on clock skew manipulations as result of different heat output levels (Murdoch)
- signal is not directly transferred between sender and receiver over shared physical environment

Attack patterns for physical communication

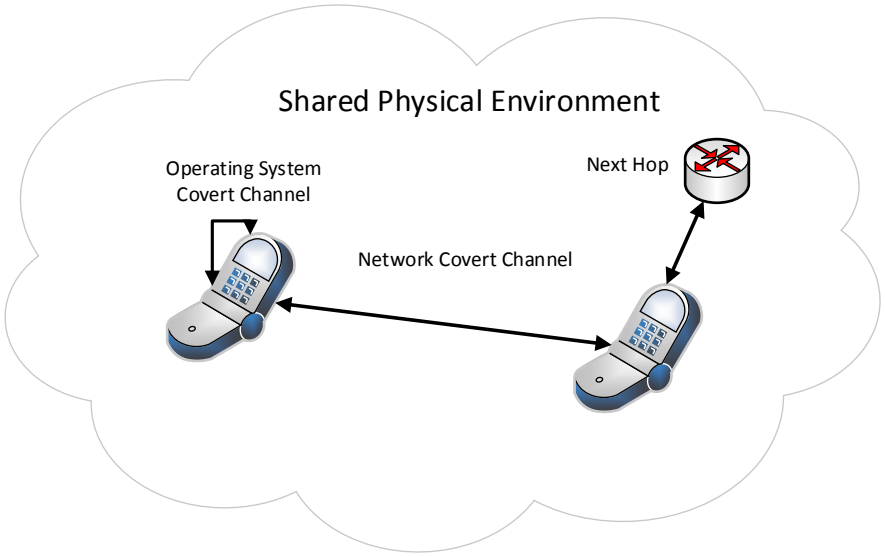


Figure: Physical Attack Patterns

Examples for physical communication

1. a covert physical channel
2. covert acoustical mesh network (multi-hop, 20 bit/s up to 19.7 m) (Hanspach and Goetz, Journal of Communications)

Offtopic: Covert Acoustical Mesh Network Experiment

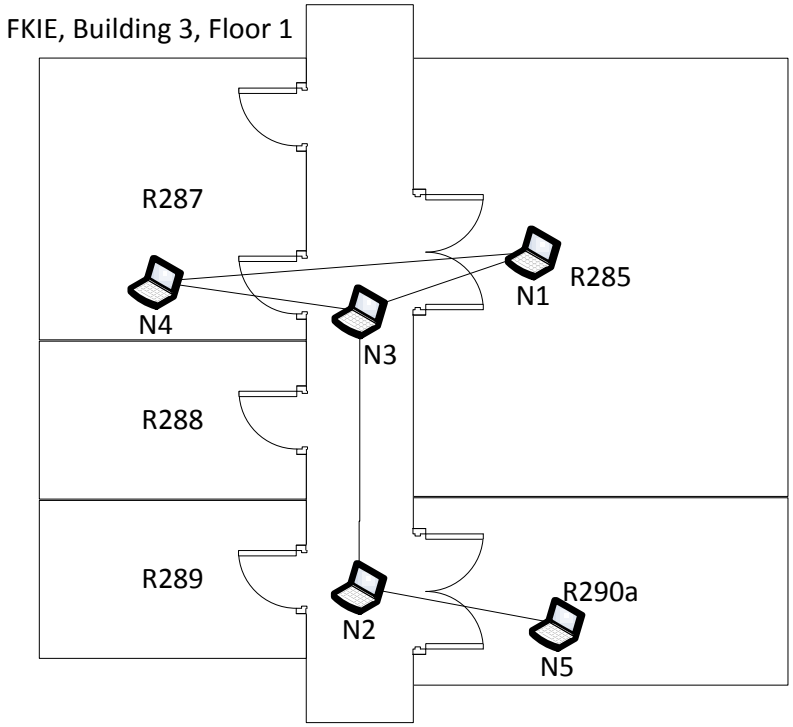


Figure: Schematic view of the interconnection experiment at Fraunhofer FKIE

Offtopic: A closer look at the audio filtering guard

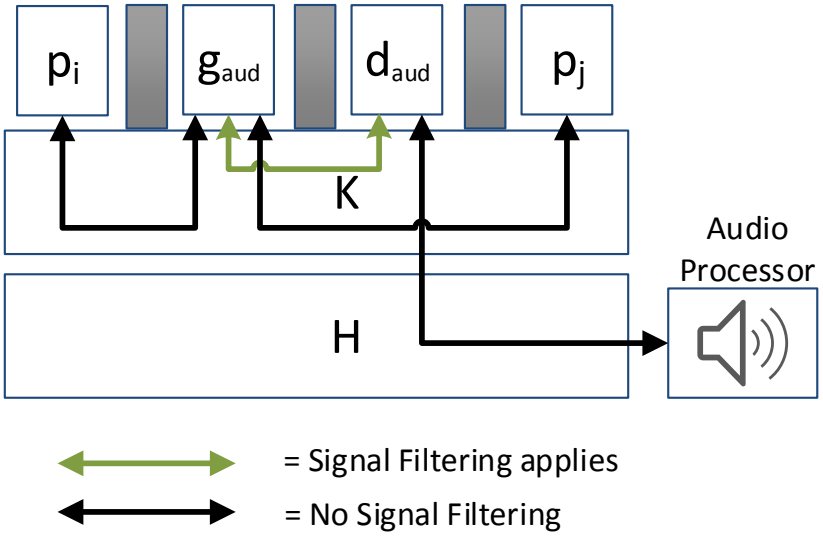


Figure: An audio filtering guard used by two different application partitions

Suggested Taxonomy

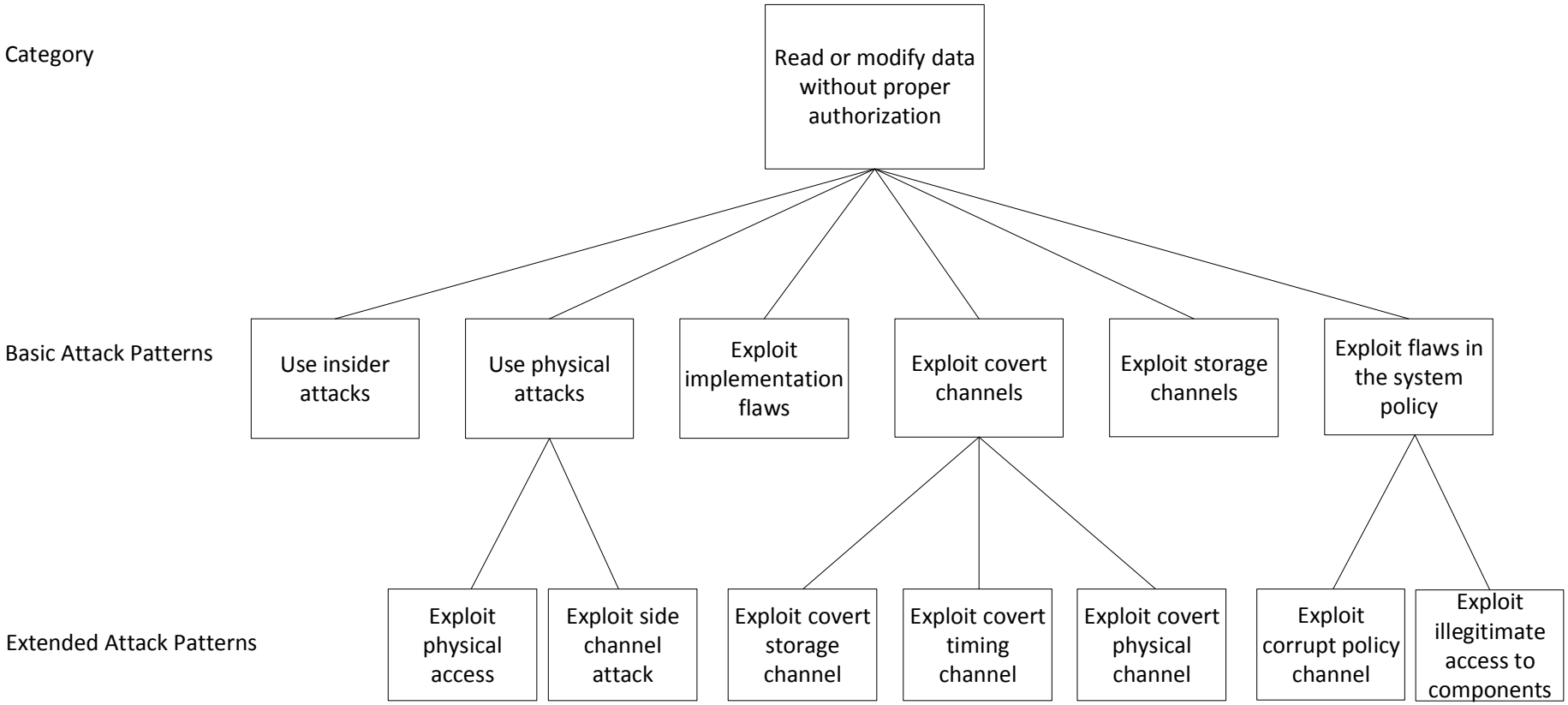


Figure: Taxonomy of attack patterns in component-based operating systems

Conclusions

- systematic study of information flows in operating system models
- every possible information flow covered
- information flows are mapped to specific attack patterns
- more detailed attack patterns might be developed in future
- algebra might be used for future information flow analyses
- covert physical channels are introduced (more work coming soon...)
- attack patterns might be used to systematically harden operating systems