Malware Detection

Prof. Ravi Sandhu
Executive Director and Endowed Chair

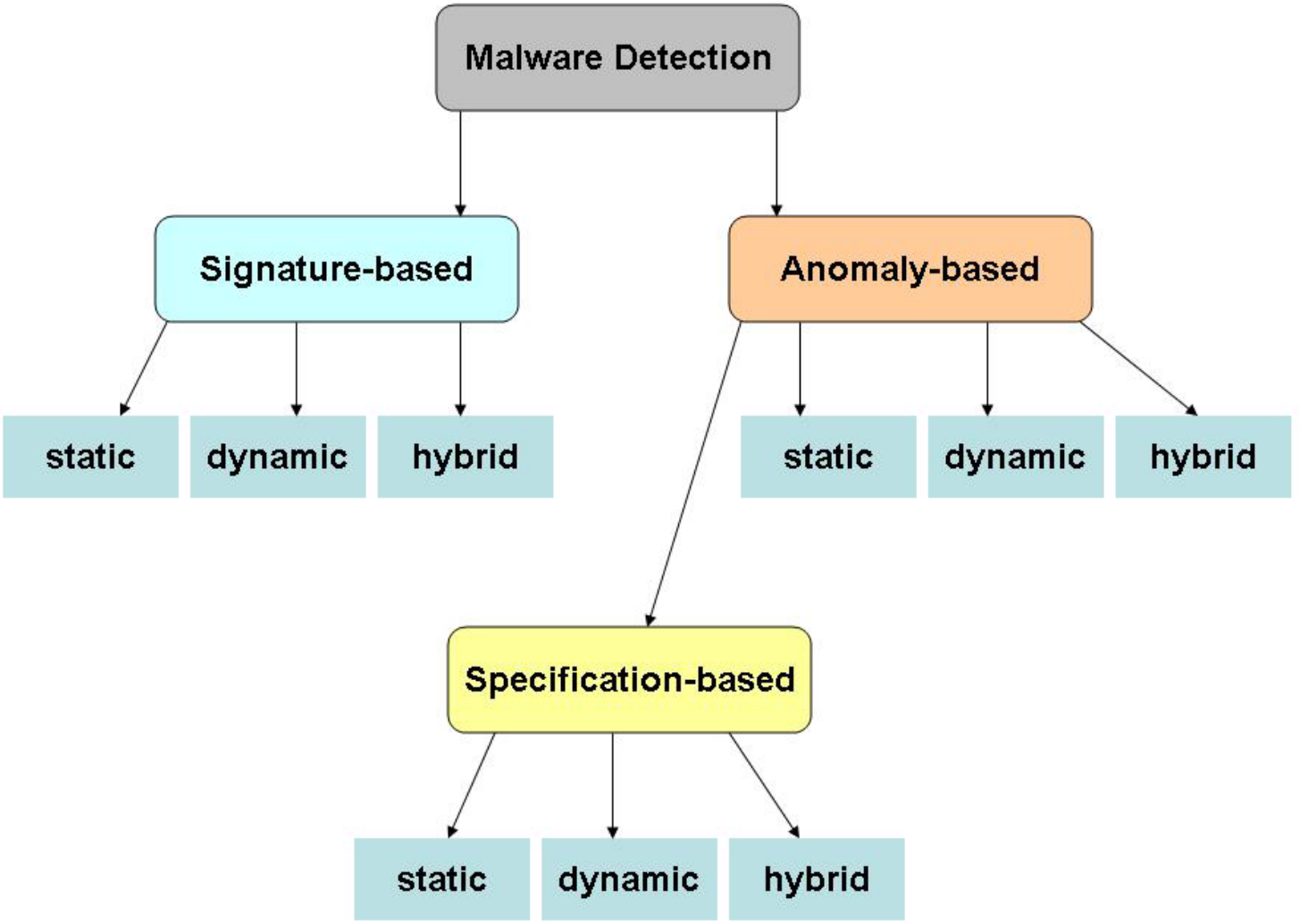
Lecture 12

ravi.utsa@gmail.com
www.profsandhu.com

Highlights

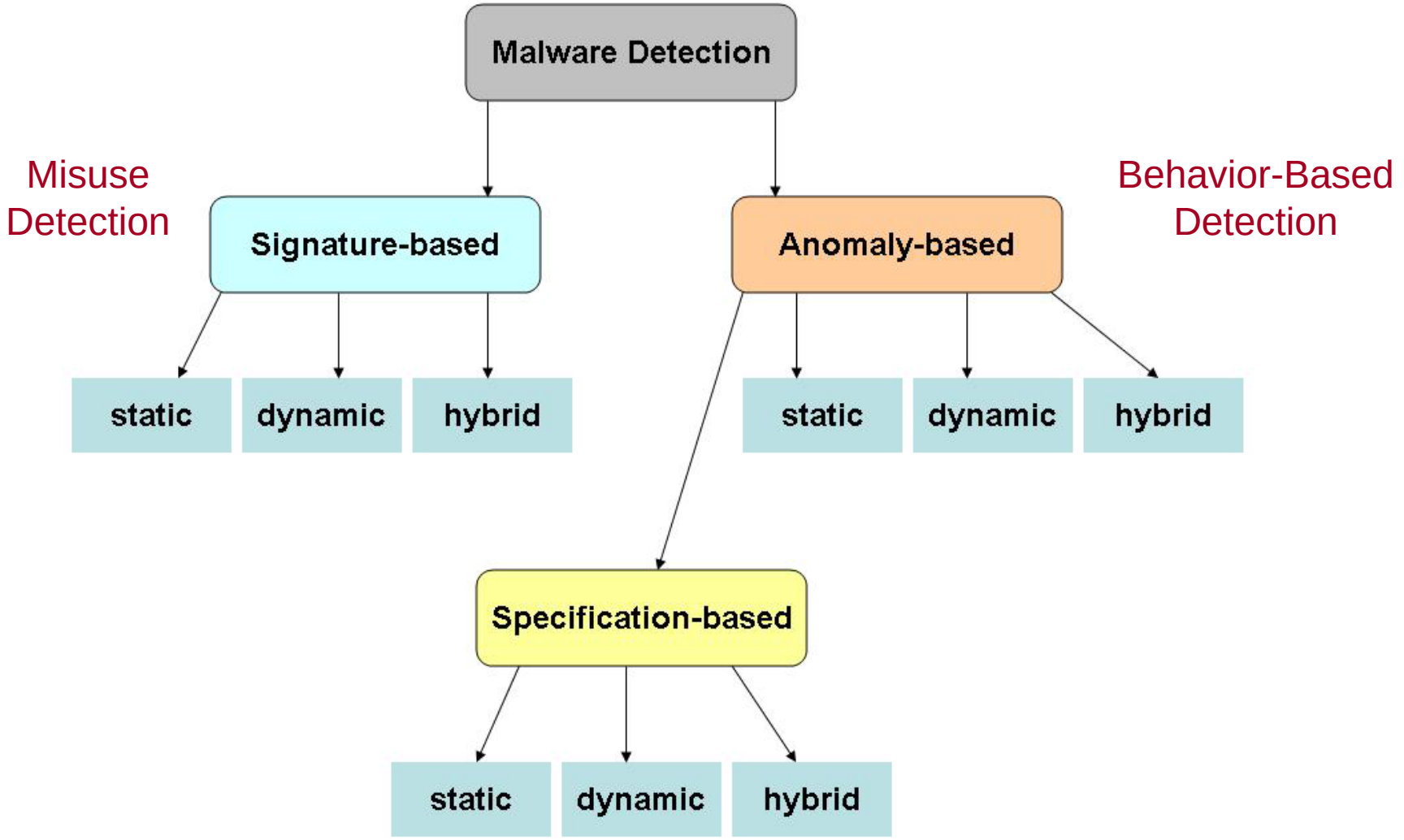
- Virus detection is undecidable
  - Cohen dissertation (1985), paper (1987)
- Anti-virus (more generally anti-malware) is a great business model
  - Need regular updates
  - Infinite supply of new malware
- Malware can be stealthy
- Malware can be really stealthy

Malware Detection Techniques



Nwokedi Idika and Aditya Mathur, A Survey of Malware Detection Techniques, Purdue University, Feb 2007.

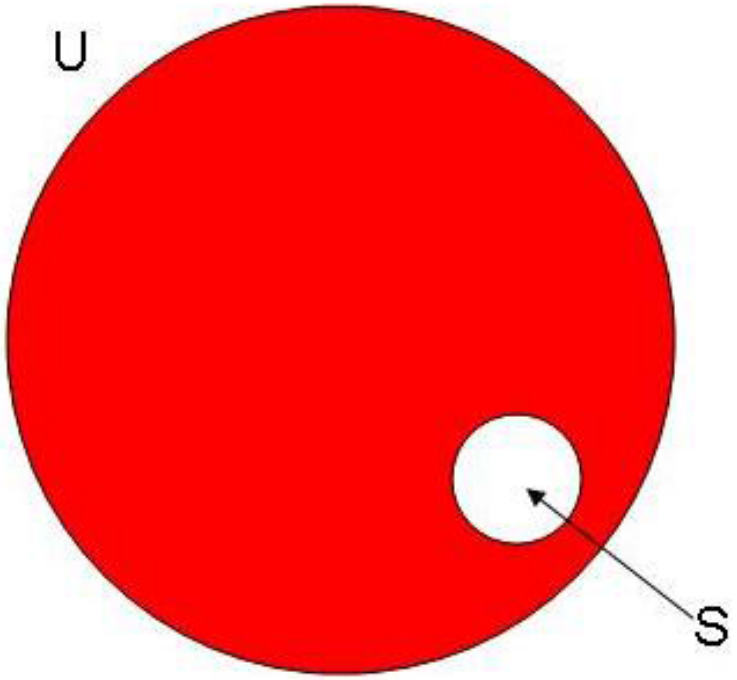
Malware Detection Techniques



Nwokedi Idika and Aditya Mathur, A Survey of Malware Detection Techniques, Purdue University, Feb 2007.

Signature Limitations

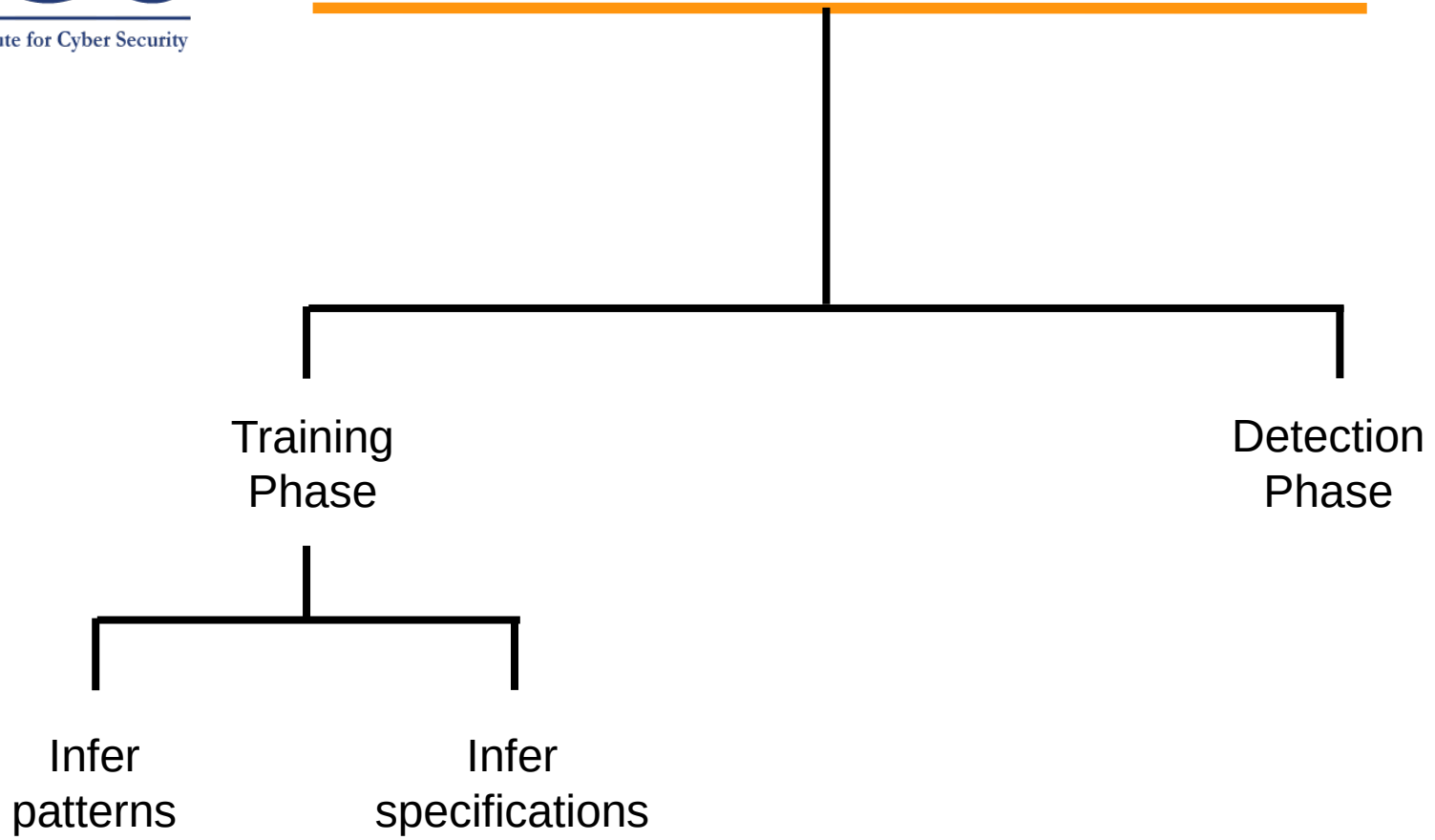
U = set of all malicious behavior
S = set of all known signatures



S needs regular updates

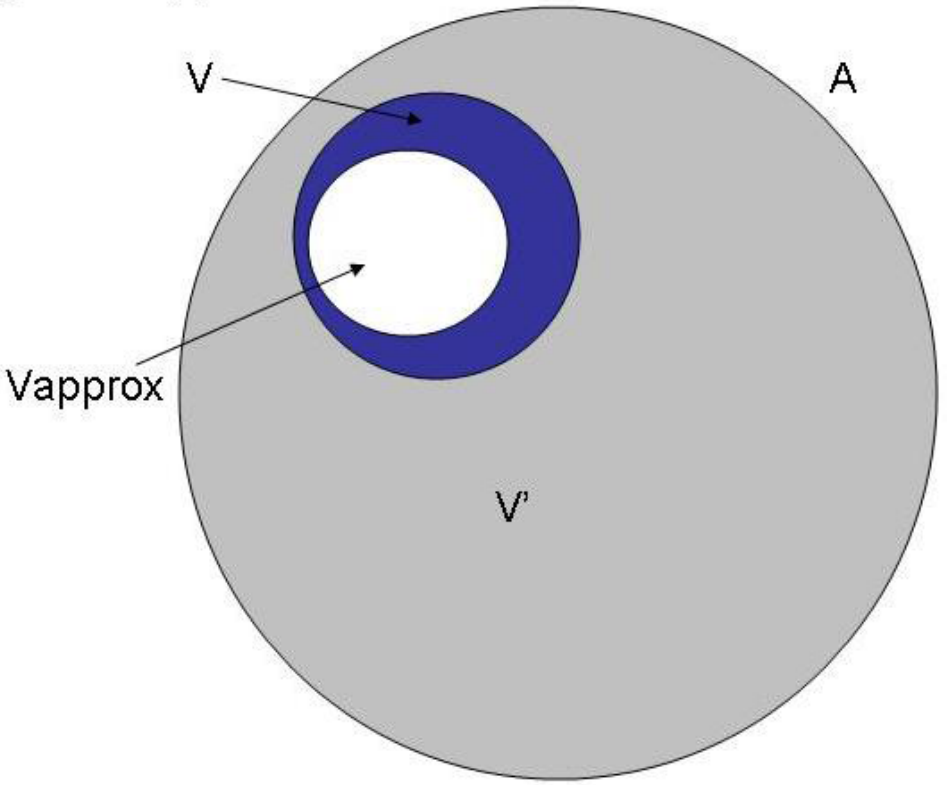
Nwokedi Idika and Aditya Mathur, A Survey of Malware Detection Techniques, Purdue University, Feb 2007.

Anomaly Based



Anomaly Based Limitations

A = set of all behaviors
V = set of all valid behaviors
Vapprox = approximation to V



Blue area is false positives
If white area extends outside blue area we have false negatives

Nwokedi Idika and Aditya Mathur, A Survey of Malware Detection Techniques, Purdue University, Feb 2007.

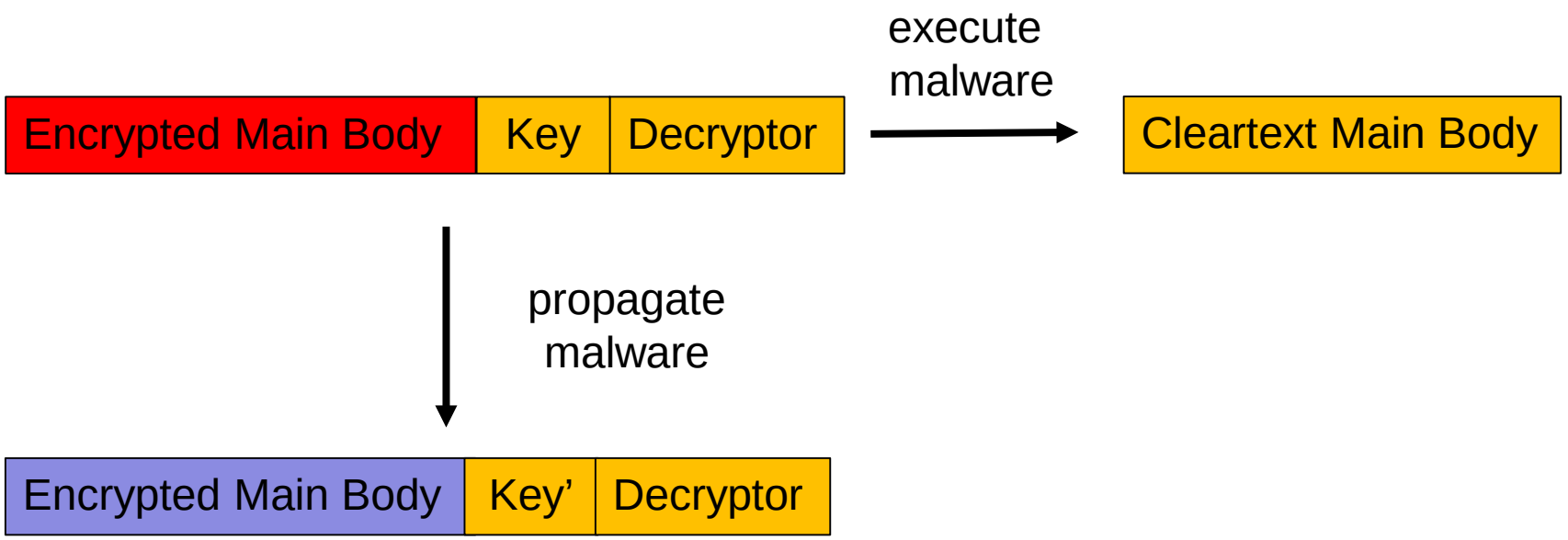
Stealthy Malware

- Defeat signature-based detection
  - Encrypted malware
  - Polymorphic malware
  - Metamorphic malware

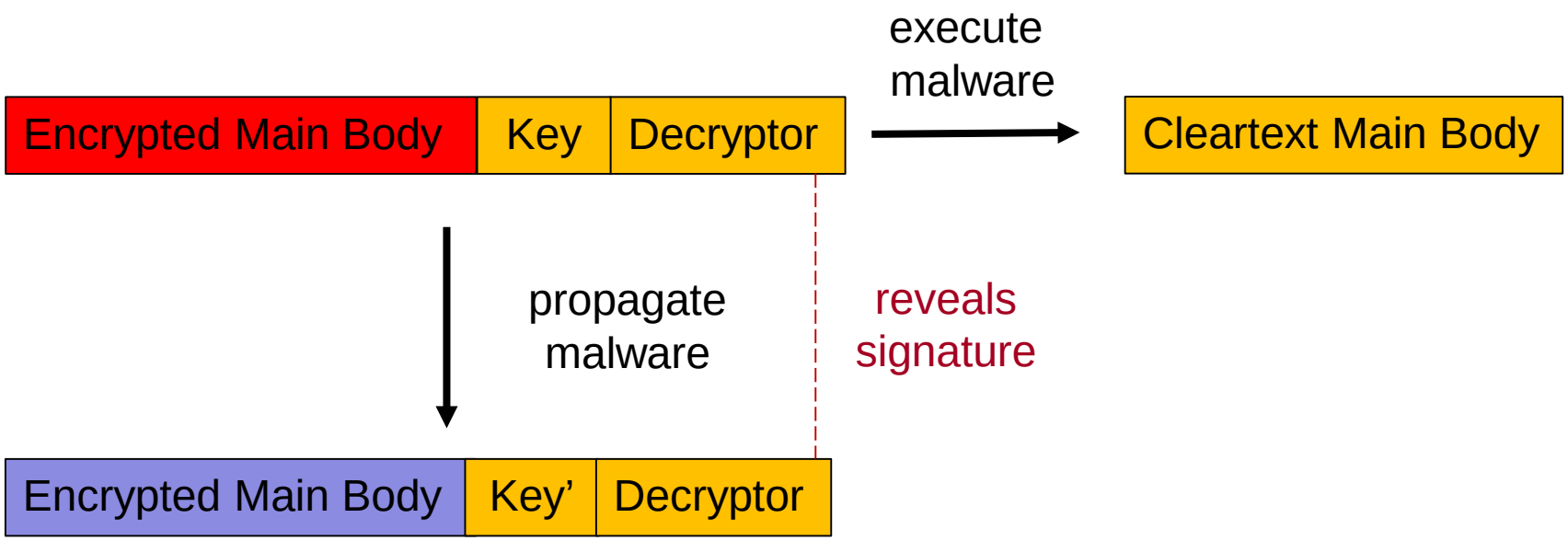
You, I., and Yim, K. Malware obfuscation techniques: A brief survey. IEEE International Conference on Broadband, Wireless Computing, Communication and Applications, Nov 2010, pp. 297-300.

  - Rootkit can misrepresent the existence or content of executable files

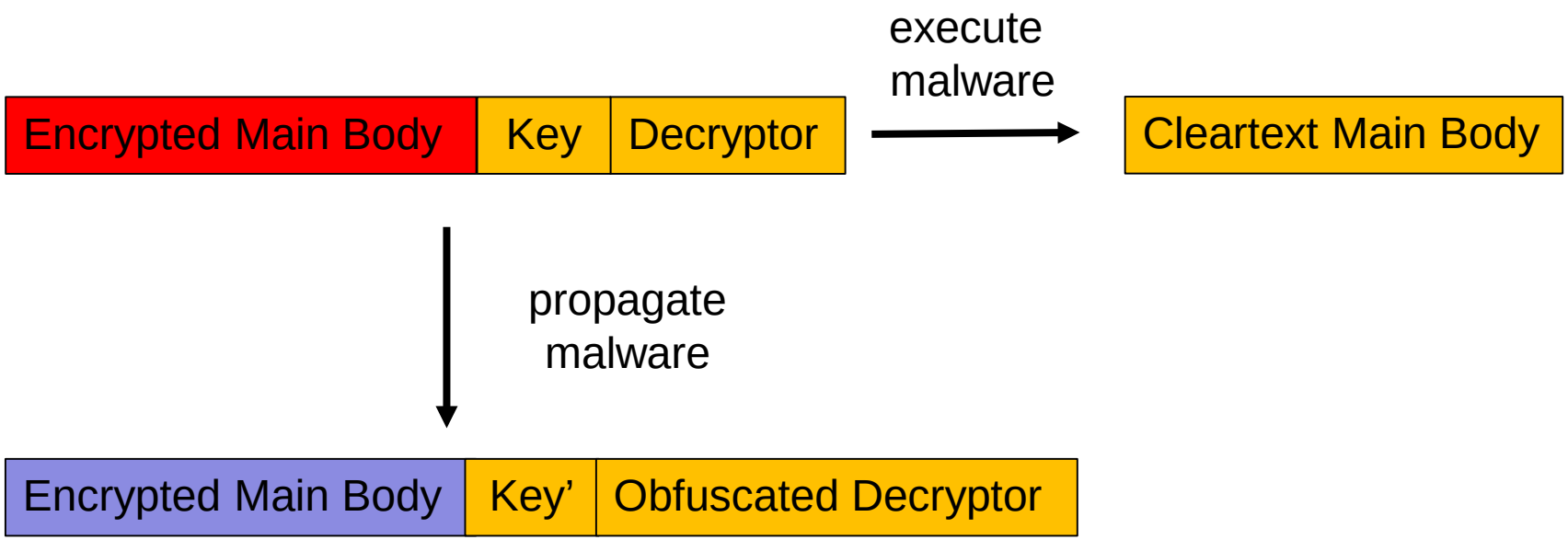
Encrypted Malware



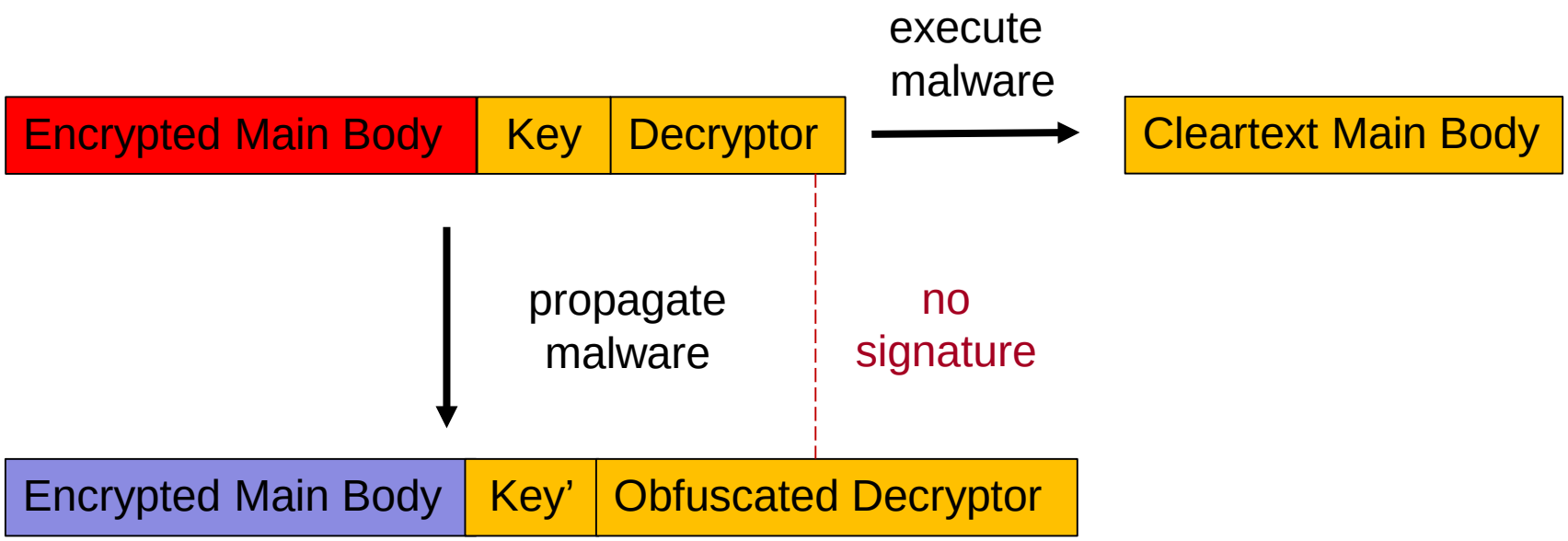
Encrypted Malware



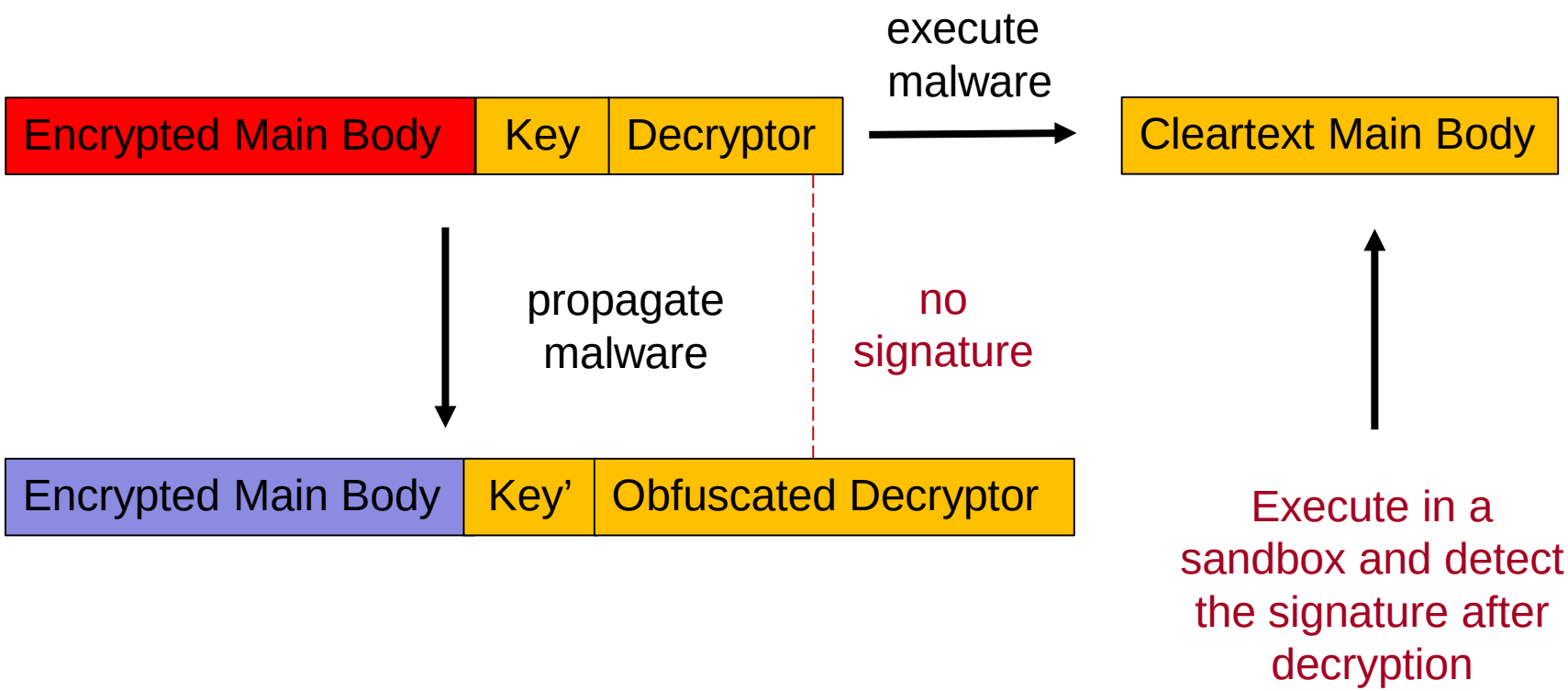
Polymorphic Malware



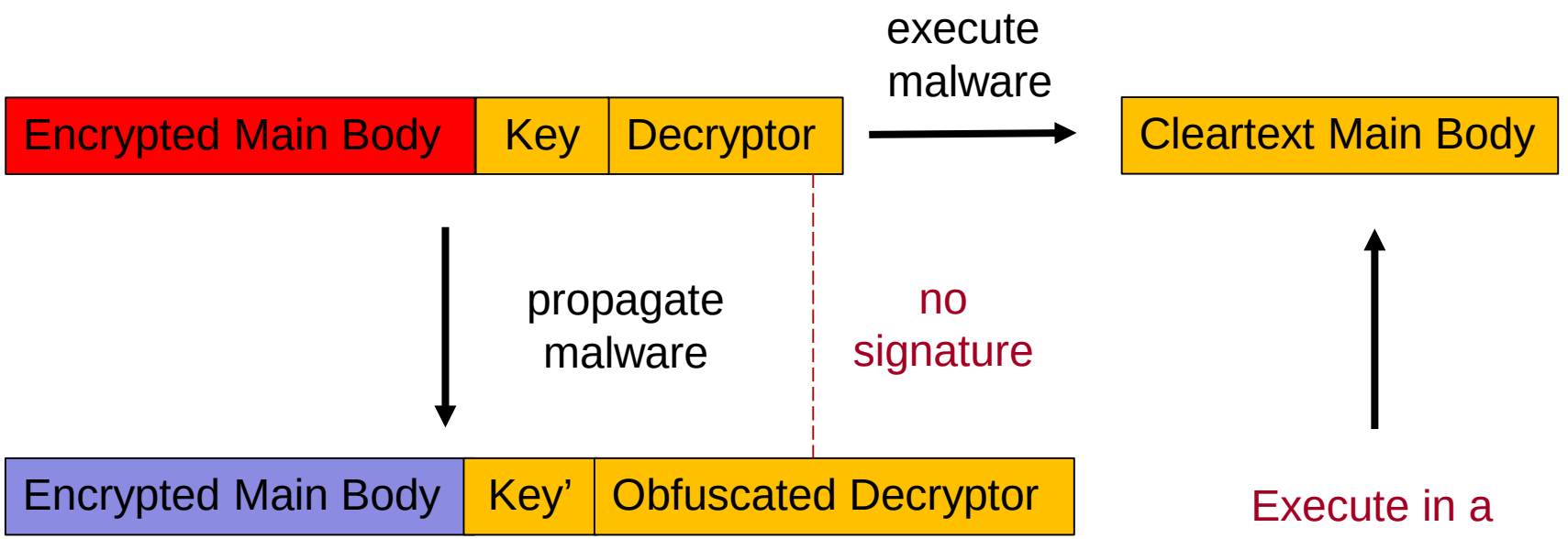
Polymorphic Malware



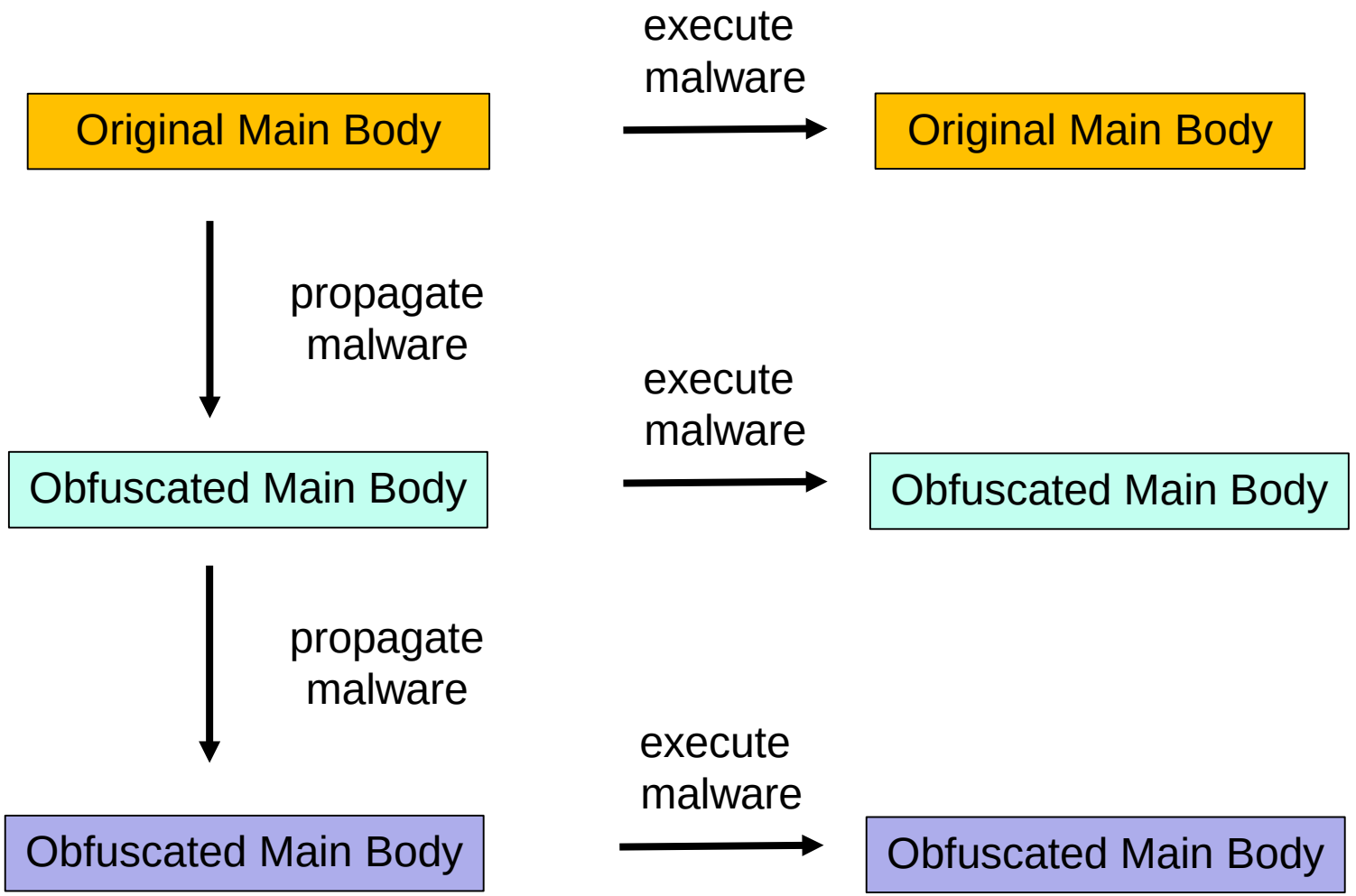
Polymorphic Malware



Polymorphic Malware



Metamorphic Malware



no signature

Obfuscation Techniques

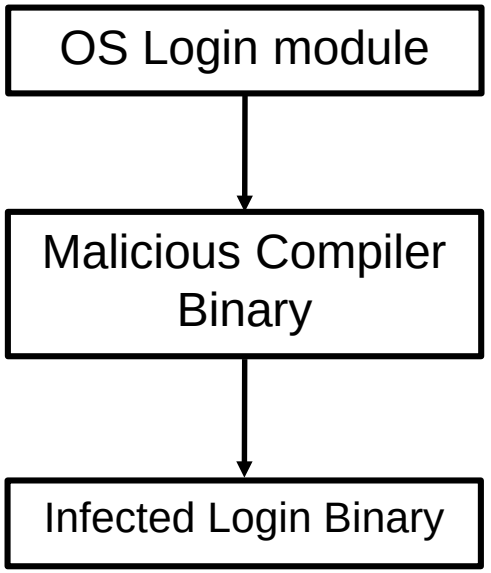
- Dead-Code Insertion
- Register Reassignment
- Subroutine Reordering
- Instruction substitution
- Code transposition
- Code Integration

Really Stealthy Malware

- Not visible in source code
- Reappears in binary code due to malware infected compiler
- In theory could reappear in binary code due to other components in binary execution workflow
  - Loader
  - Linker
  - OS
  - BIOS

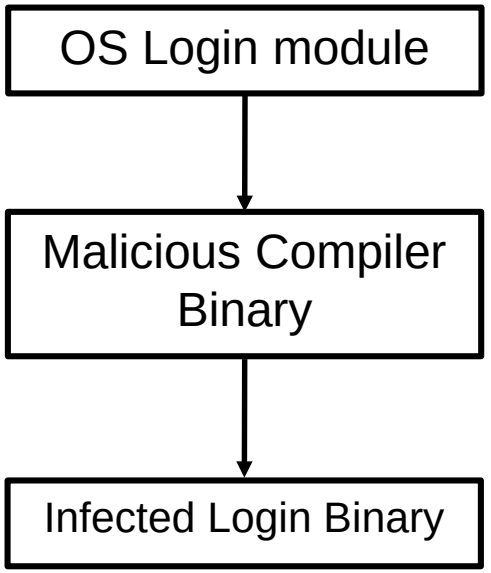
Ken Thompson. Reflections on trusting trust. Commun. ACM 27, 8 (August 1984), 761-763.

Malicious Compiler Inserts a Backdoor

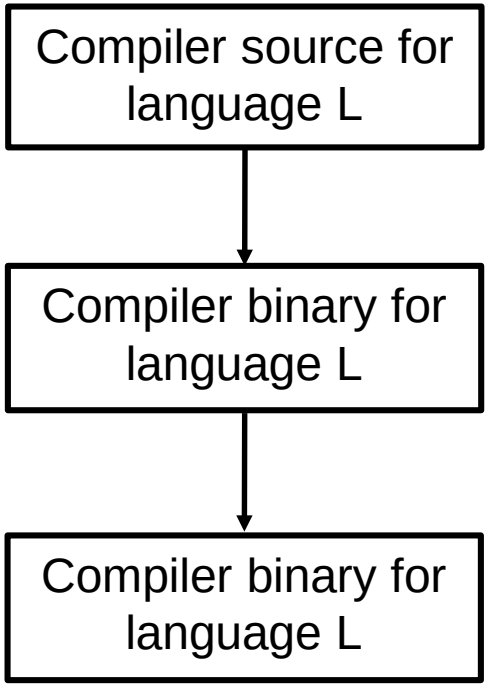


Malicious Compiler Inserts a Backdoor

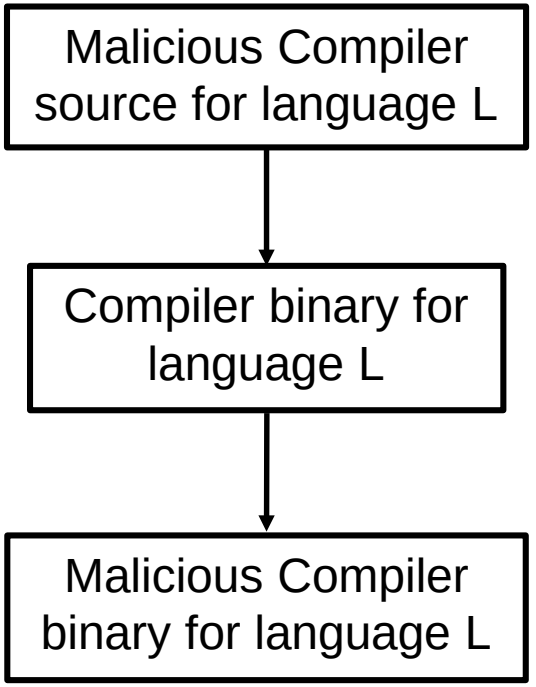
Assumption:
Malicious behavior cannot be detected in binary, but may be detectable in compiler source



Self-Compiler

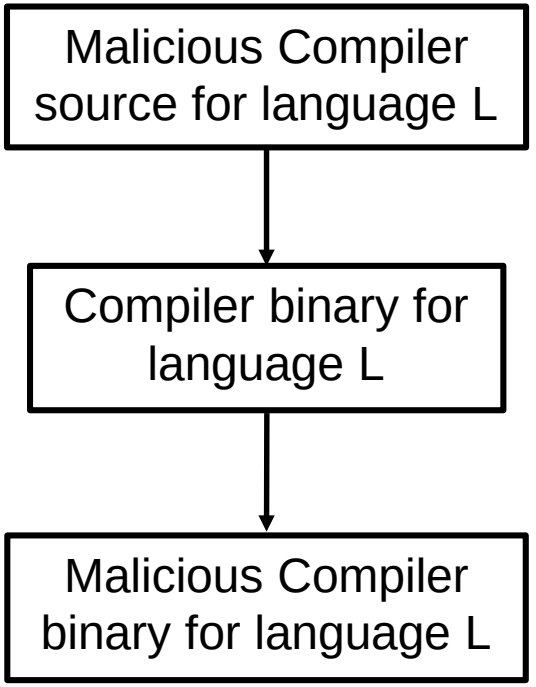


Malicious Self-Compiler in Binary and Source



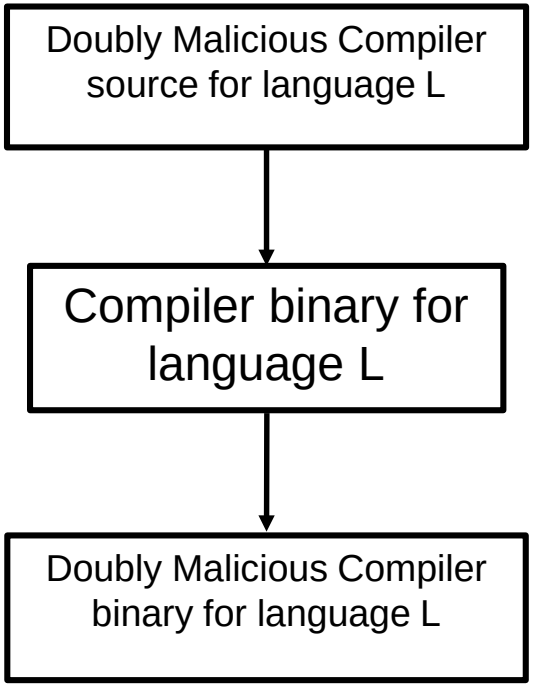
Malicious Self-Compiler in Binary and Source

Source code analysis will reveal malicious behavior

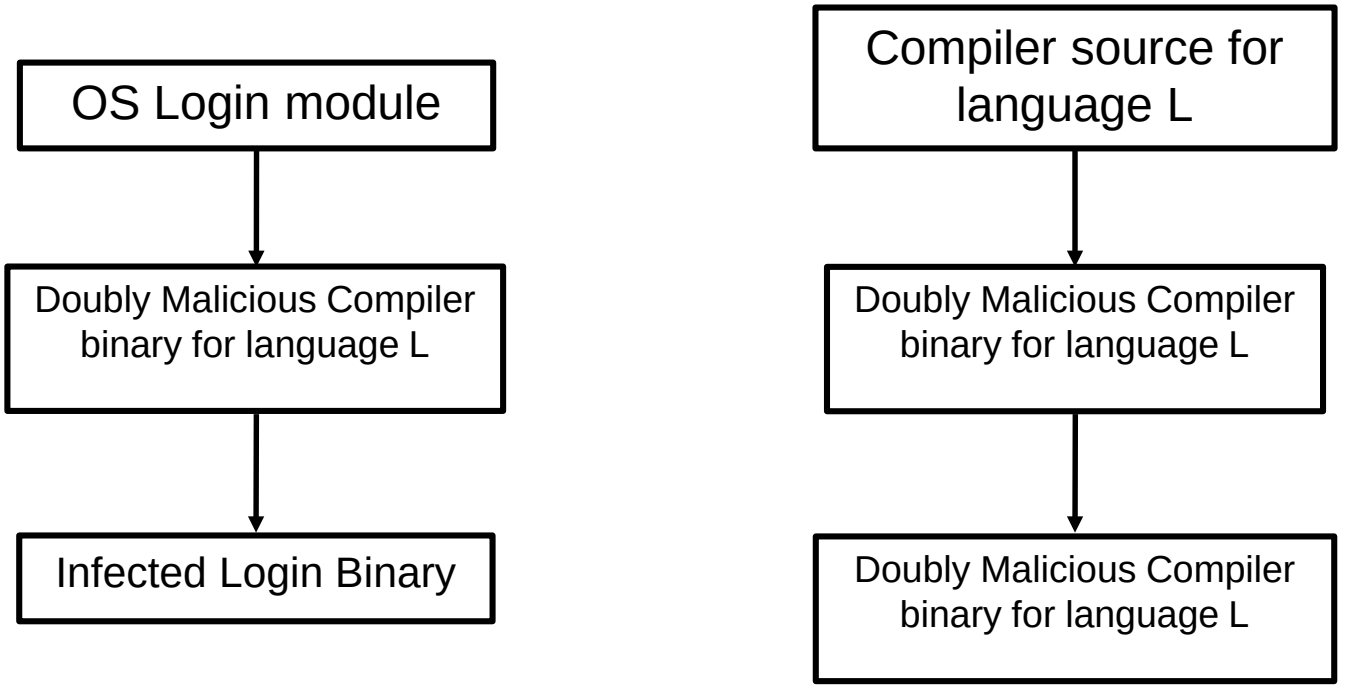


Doubly Malicious Self-Compiler in Binary and Source

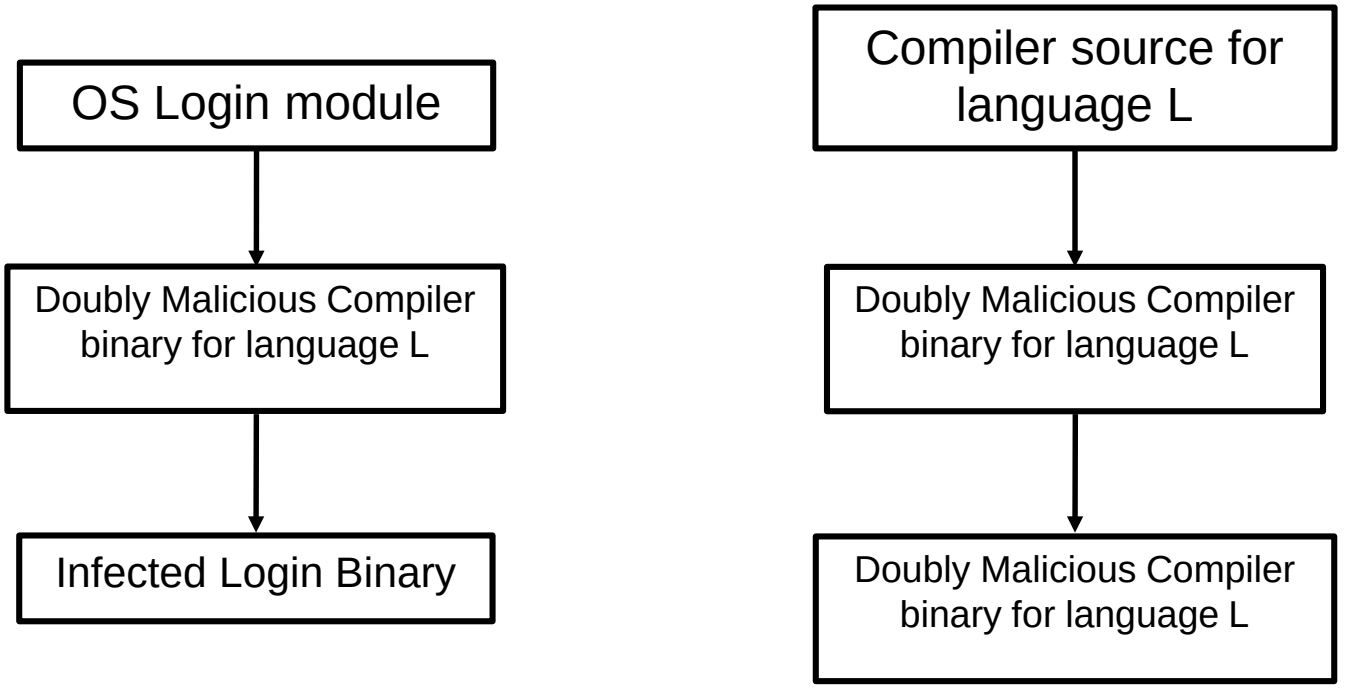
Source code analysis will reveal doubly malicious behavior



Doubly Malicious Complier Binary Behavior

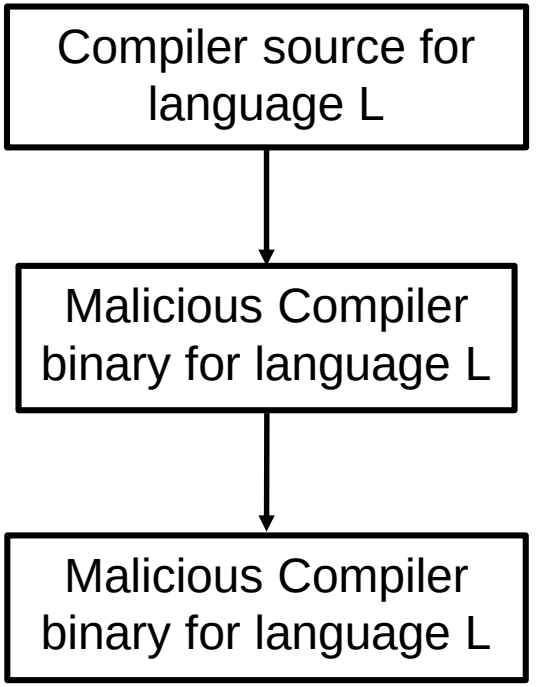


Doubly Malicious Complier Binary Behavior



No trace of malicious behavior in source code

Malicious Self-Compiler in Binary but not in Source



partial countermeasure

Wheeler, D.A., Countering trusting trust through diverse double-compiling, 21st Annual Computer Security Applications Conference, pp.13-48, 5-9 Dec. 2005.