Process Implanting
A New Active Introspection Framework for Virtualization
Zhongshu Gu†, Zhui Deng†, Dongyan Xu†, Xuxian Jiang*  
†Purdue University  ‡North Carolina State University

Motivation
- In-context Execution  
- Agnostic to OS release
- Memory Footprints  
- Tamper from Malware
- Unreliable Results

Out-of-box Approach
- Transparent Monitoring  
- Tamper Resistance
- Semantic Gap  
- Coarse Grained
- OS Release Sensitive

Process Implanting Approach
- In-box Context:  
  • Dynamically implant process into VM  
  • Active introspection  
  • Reuse of existing tools
- Out-of-box Protection:  
  • Protection by underlying hypervisor  
  • Process execution awareness  
  • Stealthy to malicious process

Security Features
- Scheduled out  
  - Recover victim process' context
- Scheduled in  
  - Recover implanted process' context

Frequent Scene Restoration
- Mandatory restoration of victim process
- Covert channel communication
- Privilege escalation
- "Unkillable" flag setting

Coordination & Protection from Hypervisor
- Memory space is allocated at runtime
- Implanted program is loaded at runtime
- Address beyond boundary of physical memory

Invisible Memory Space
- "Memory space allotted for big system testing"

Conclusion
- A general purpose active VM introspection framework
- Narrowing semantic gap between guest VM and hypervisor
- Minimum impact on guest system after restoration
- Coordination and protection from hypervisor

Design Overview

In-box Context:
- Memory space is allocated at runtime
- Implanted program is loaded at runtime
- Address beyond boundary of physical memory

Out-of-box Protection:
- Protection by underlying hypervisor
- Process execution awareness
- Stealthy to malicious process

Hypervisor
Guest Operating System
Virtual Machine Monitor

Victim Process
Implanted Process
Other Process

Process Implanting Approach