Towards Reliable Rootkit Detection in Live Response

IMF, Stuttgart, September 2007

Felix C. Freiling
University of Mannheim, Germany

Bastian Schwittay
Symantec (Deutschland) GmbH, Germany

Motivation

- Traditional forensics use **dead** analysis
- Live Response captures data from **live** systems
- **Rootkits** change the behaviour of live systems

- **Goal**: Increase credibility of live response
- **Subgoal**: Sound methods for reliable rootkit detection during Live Reponse

Agenda

- Motivation
- Background
  - Live Response
  - Windows Rootkits
- Detection Experiments
- Results and Recommendations
- Summary and Discussion

Live Response

- Volatile data is lost when powering down a computer
  - Running processes
  - Open network ports
  - Kernel modules loaded
  - RAM contents
  - …
- Live Response includes all techniques that capture data from running systems

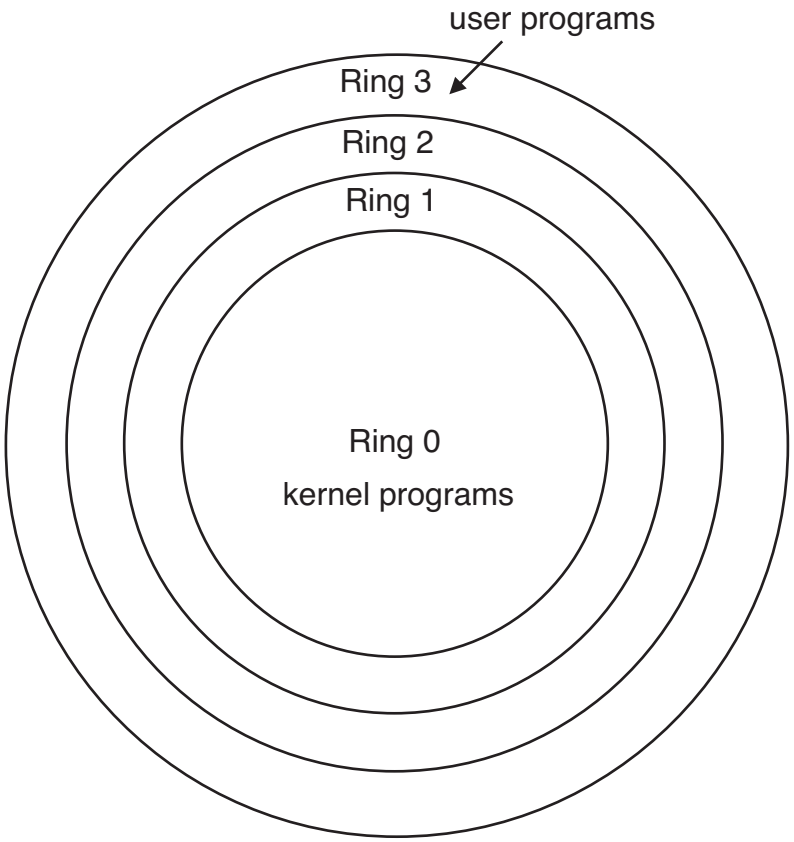
Live Response Dilemma

- **Dilemma**: Live Response techniques alter the running system‘s state
- Acceptable only if alterations are well-understood
- Tradeoff between value of captured information and integrity of the evidence

Windows Rootkits

- "a set of programs and code that allows a permanent and undetectable presence on a computer" [Hoglund and Butler]
- Ultimate attacker's tool
- Stealth techniques to alter a running system:
  - Hide processes, files, drivers, ports etc.
  - Optionally include backdoors or keyloggers

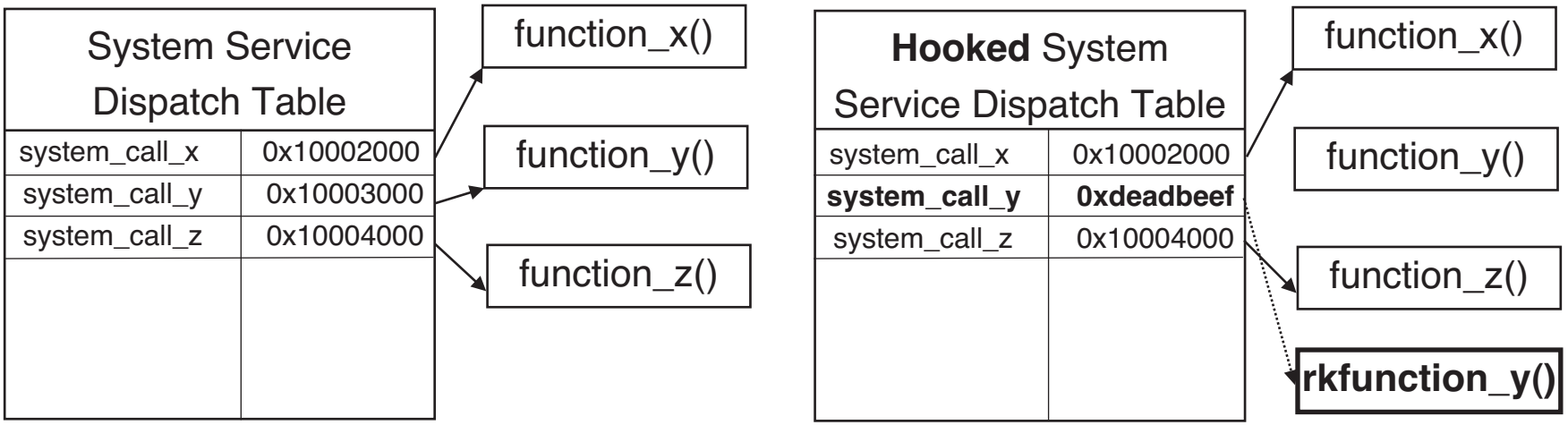
Windows Internals



- Ring 0 software has full system priviliges
- *Kernel rootkits* run in Ring 0
- Live Response tools mostly run in Ring 3
- In the presence of a rootkit, Live Response tools can not capture accurate data!

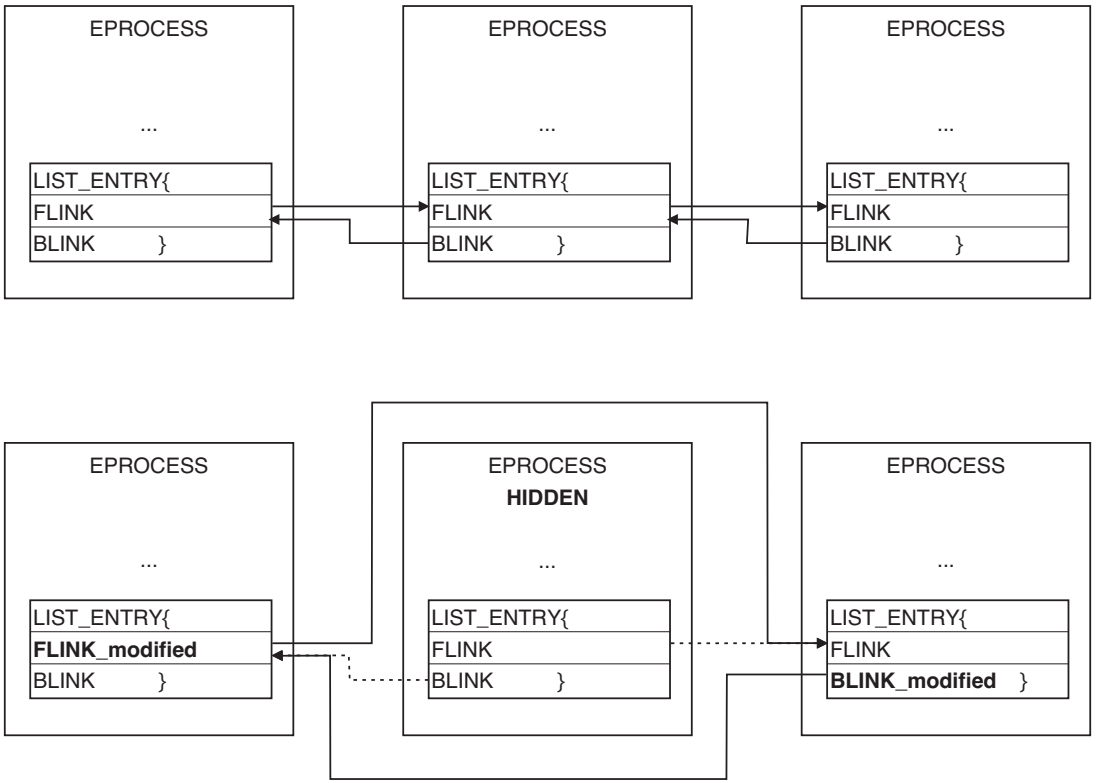
Rootkit Techniques: Hooking

- „Hooking“ – alter the execution path of applications
- Example: SSDT Hooking



Rootkit Techniques: DKOM

- „Direct Kernel Object Manipulation“
- Alter in-memory list of processes



Detection Tools

- Use heuristics to discover inconsistencies in kernel
  - Discover hooks
  - Discover hidden kernel objects
- Some tools use cross view detection:
  - Compare output of API functions with results from parsing internal data structures

Agenda

- Motivation
- Background
  - Live Response
  - Windows Rootkits
- Detection Experiments
- Results and Recommendations
- Summary and Discussion

Experimental Setup (1/2)

- 11 different publicly available rootkits
  - Available from rootkit.org
- 12 different rootkit detectors
  - Using different heuristics
- 6 well-known Live Response tools
  - pslist, fport, netstat, psservice, find, regdmp
- 4 different flavors of Windows
  - Windows 2000 SP4
  - Windows XP (no updates)
  - Windows XP SP2
  - Windows 2003 Server SP1

Experimental Setup (2/2)

- For each flavor of Windows do
  - For each rootkit do
    - If the rootkit offers file hiding capabilities create a hidden file
    - If the rootkit offers process hiding capabilities create a hidden process
    - If the rootkit ...
    - For each rootkit detector do
      - Check what hidden objects are detected
      - Revert system into original (infected) state
- Possible detection results:
  - No hidden objects detected
  - Some but not all hidden objects detected
  - All hidden objects detected

Results (1/3)

- Severe compatibility problems with rootkits
  - "Best" platform was Windows XP SP2
- Also some compatibility problems with detectors

Results (2/3)

	Rootkits								
Detectors	AFX	FU	FUTo	HxDef	Klog	NtIll.	Vanq.	phide	HPHM
Darkspy	2	2	2	2	–	2	2	2	2
Flister	2	–	–	0	–	2	2	–	–
Blacklight	2	2	0	2	–	2	2	2	2
IceSword	2	2	2	2	–	2	2	2	2
modgreper	–	–	–	1	–	–	–	–	–
RKDetector	2	2	0	0	–	2	0	2	2
RKHA	–	–	–	–	–	–	–	–	2
RKRevealer	2	–	–	2	–	0	2	–	–
SVV	2	–	–	2	–	1	2	–	1
UnhackMe	2	0	0	2	–	1	2	0	2
VICE	2	–	–	2	–	–	2	–	2

0 = no detection, 1 = partial detection, 2 = complete detection, - = incompatible

Results (3/3)

- Using the live reponse tools **none** of the hidden objects (files, processes, ports etc.) were detected

- Rootkit detection is necessary in live response!

Recommendations

- As of June 2006, the combination of the following three rootkit detectors offers complete detection:
  - Blacklight
  - IceSword
  - System Virginity Verifier (SVV)
- Good individual detection rate
- Redundancy in detection
- Different approaches result in resilience against implementation-specific attacks

Methodology

- Experiments should be repeated and documented regularly
- Result in recommendation of rootkit detectors
- Examiners use this combination of rootkit detectors
- If no rootkit is found, hypothesis that a known rootkit was installed during live response can be refuted

Summary

- Live response is becoming an integral part of incident response and digital forensics
- Rootkits subvert systems at a very low level, fooling classic live response tools
- Reliable rootkit detection is needed
- Proposed methodology combines different detection tools to achieve reliability

- What about Virtualization rootkits like Rutkowska's BluePill?

References

- Greg Hoglund and James Butler: Rootkits - Subverting the Windows Kernel. Addison-Wesley, 2005.
- Bastian Schwittay: Towards automating analysis in computer forensics. Diplomarbeit, RWTH Aachen, Department of Computer Science, 2006.
  `http://pi1.informatik.uni-mannheim.de/filepool/theses/diplomarbeit-2006-schwittay.pdf`