Visualizing Indicators of Rootkit Infections in Memory Forensics

Stefan Vömel, Hermann Lenz

7th International Conference on
IT Security Incident Management & IT Forensics

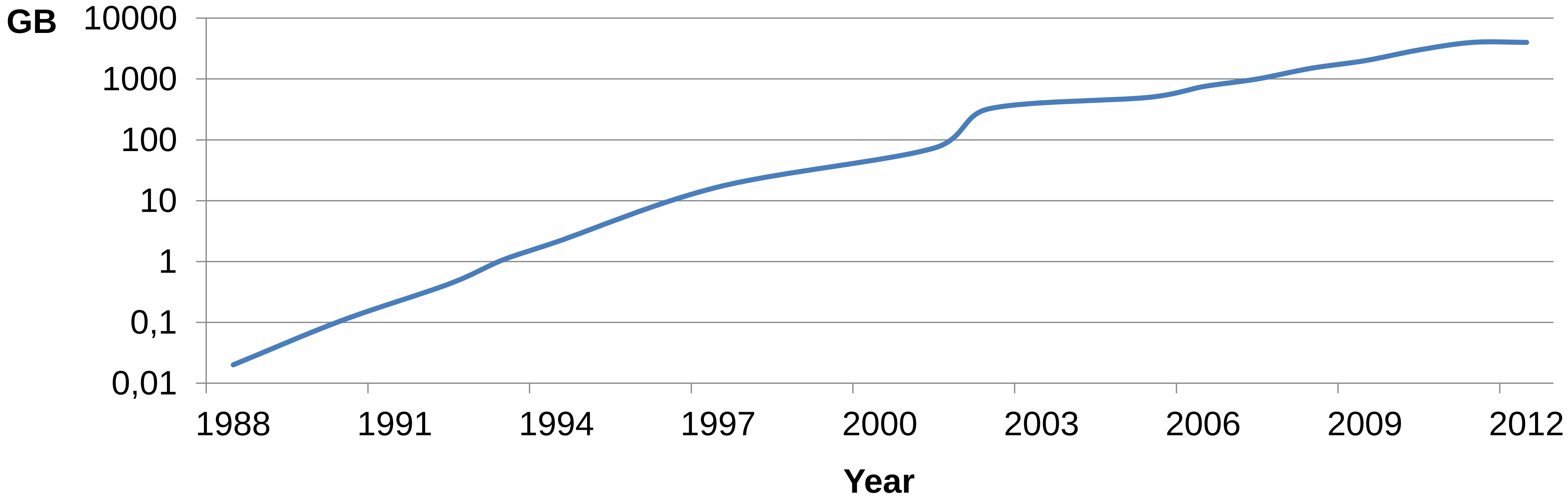
i1 - Chair for IT Security Infrastructures

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Motivation

- Traditional, hard drive-centric approaches in computer forensics have to increasingly cope with a number of challenges

- Example: Rapid growth of storage capacities



(Source: Based on http://de.wikipedia.org/wiki/Festplattenlaufwerk)

Motivation

- Further Challenges
  - Various malicious applications run solely in memory and do not leave any traces on persistent storage media any longer
  - Risk to overlook pieces of evidence if not all relevant sources of an incident are taken into consideration

ES PENINSULA NEWS

SATURDAY, MARCH 15, 2004

WORLD NEWS

r for the
Arts. 27570
rise in [illegible], on
March 20 at 2 and 4
are $38 for adults
students for the
formance and $30
nd $15 for stu-
matinee per-

SQL Slammer Worm infects 90% of vulnerable hosts within 10 minutes

Two years ago, U.S. Navy personnel and their families assigned to the Atsugi Navy base, home of the U.S.S. teacher and assistant director of the Orange County Symphony; and Mike Tiffany, has, a computer engineer. south of Tokyo — felt like home away from home, with its lush green rolling landscape and its multitude of cherry and entertained the locals for fun-filled hours. It turned out was just one of many establishm

The Daily W

July 20, 2001

Code Red Worm Epidemic

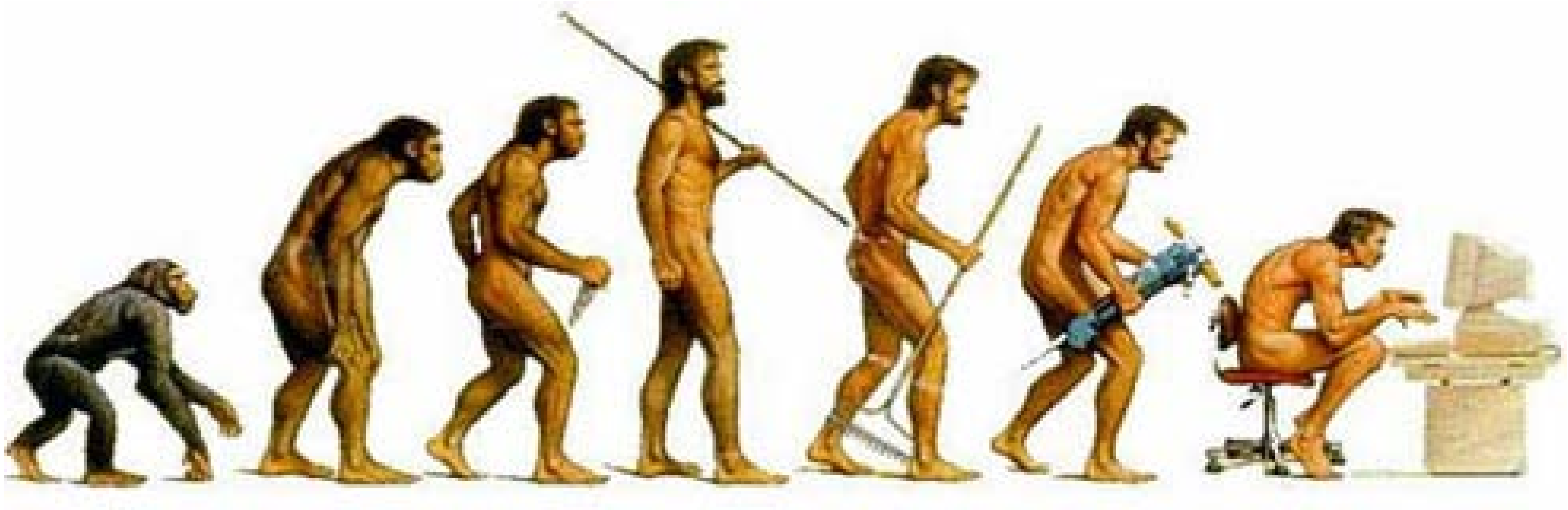
On July 19, 2001, more than 359,000 computers connected to the Internet were infected with the Code- Red (CRv2) worm in less than 14 hours. The to characterize the spread of the worm, partly due to the challenge of collecting global information about worms. Using a technique that enables global

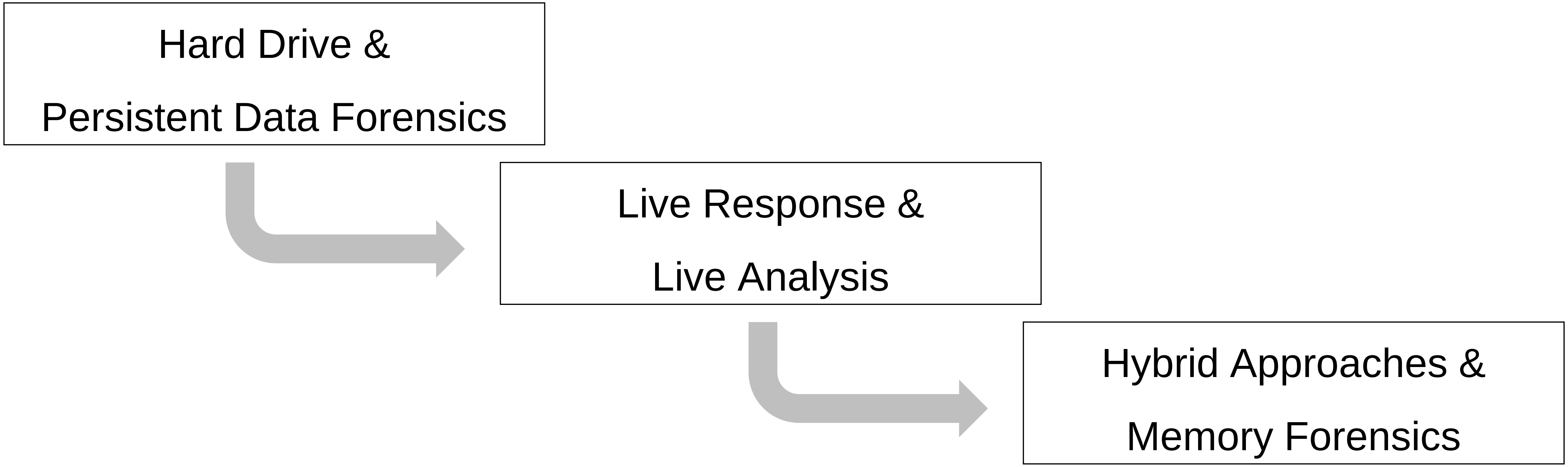
Ren
foll
imp

The
that

Motivation

- Evolution of Investigative Approaches





Hard Drive &

Persistent Data Forensics

Live Response &

Live Analysis

Hybrid Approaches &

Memory Forensics

Characteristics

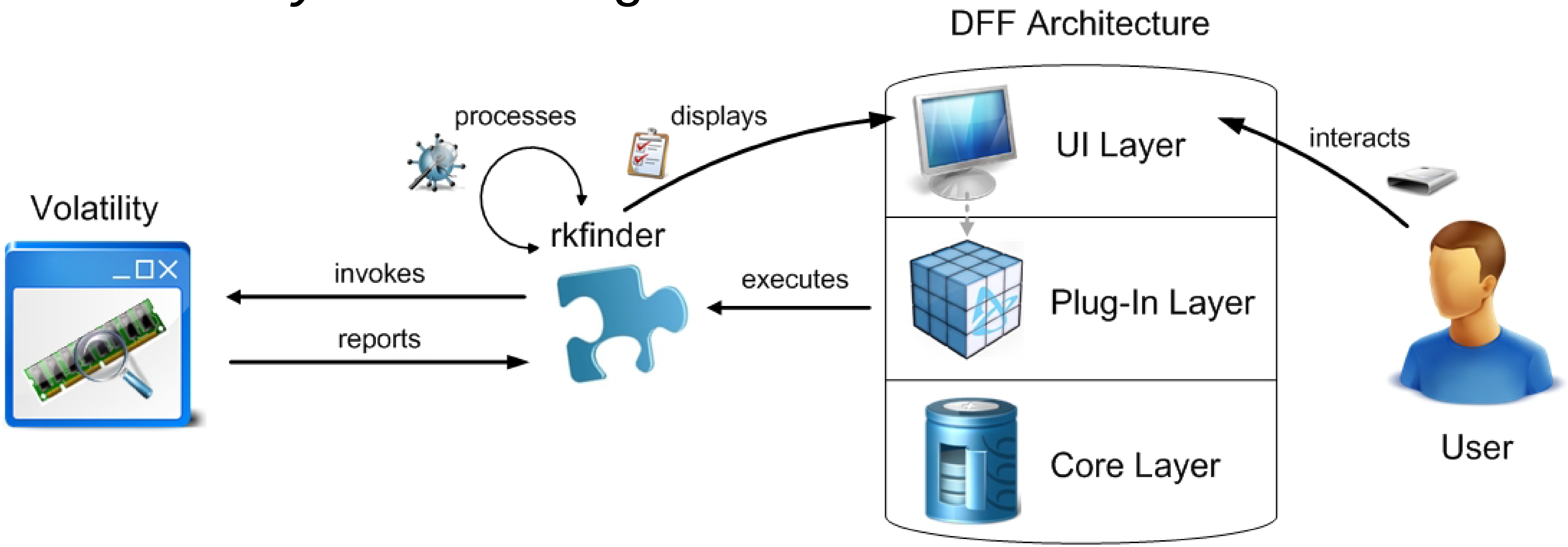
- Benefits of a Memory-Based Forensic Investigation
  - Size of memory snapshots is several magnitudes smaller than the image of a hard drive
  - Possibility of extracting state-related information, e.g., list of running processes, loaded modules, referenced files, etc.
- Problem
  - Available analysis tools mainly aim at experienced investigators
    - Report interpretation frequently requires thorough knowledge of operating system internals

Project Idea

- Idea:
  - Facilitate the memory analysis process, especially with respect to finding potentially installed malicious software
    - Automatically check system resources for consistency
    - Inconsistencies may indicate a system compromise
  - Correlate and display results in a convenient graphical user interface
    - *rkfinder* visualizes a view of the system state in a tree-like pane
    - particularly aims at users with little forensic expertise, e.g., IT personnel in smaller- and medium-sized companies

System Architecture

- Architecture of *rkfinder*
  - Written as a plug-in for the *Digital Forensics Framework* (DFF)
  - Cooperates with the memory analysis framework *Volatility* in the background

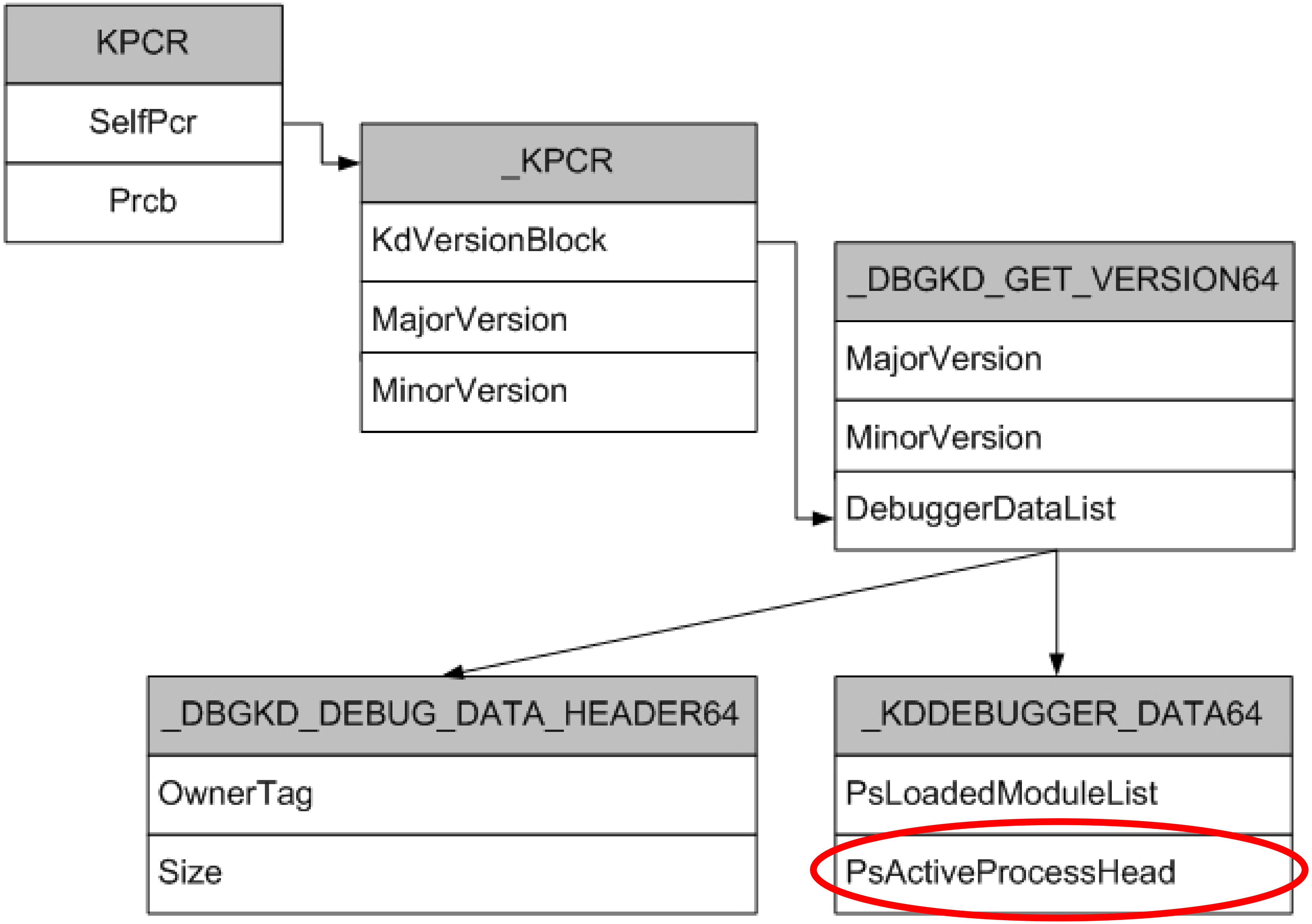


Mode of Operation

- Detection of System Inconsistencies
  - Use *cross viewing* techniques to analyze the system state from different angles
- Approach
  - Identify system objects by reconstructing a logical, *post-mortem* view of the system state
  - Identify system objects by physically scanning the memory snapshot
  - Compare all results with the output of a basic live analysis shortly after the memory snapshot has been taken

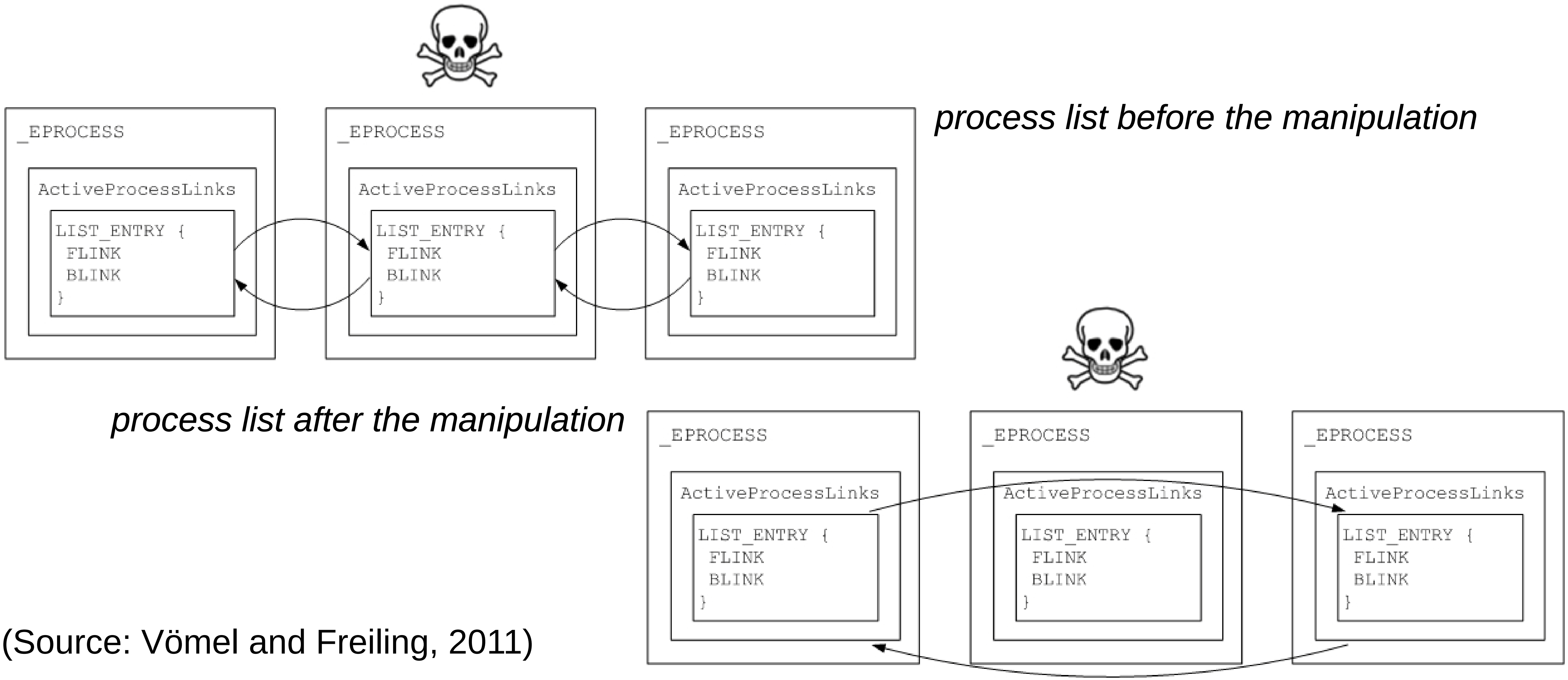
Mode of Operation

- Example: Reconstruction of the Process List



Mode of Operation

- Logical Manipulation of the Process List
    - can be revealed by matching the system state with the results of a physical memory snapshot scan



process list before the manipulation

process list after the manipulation

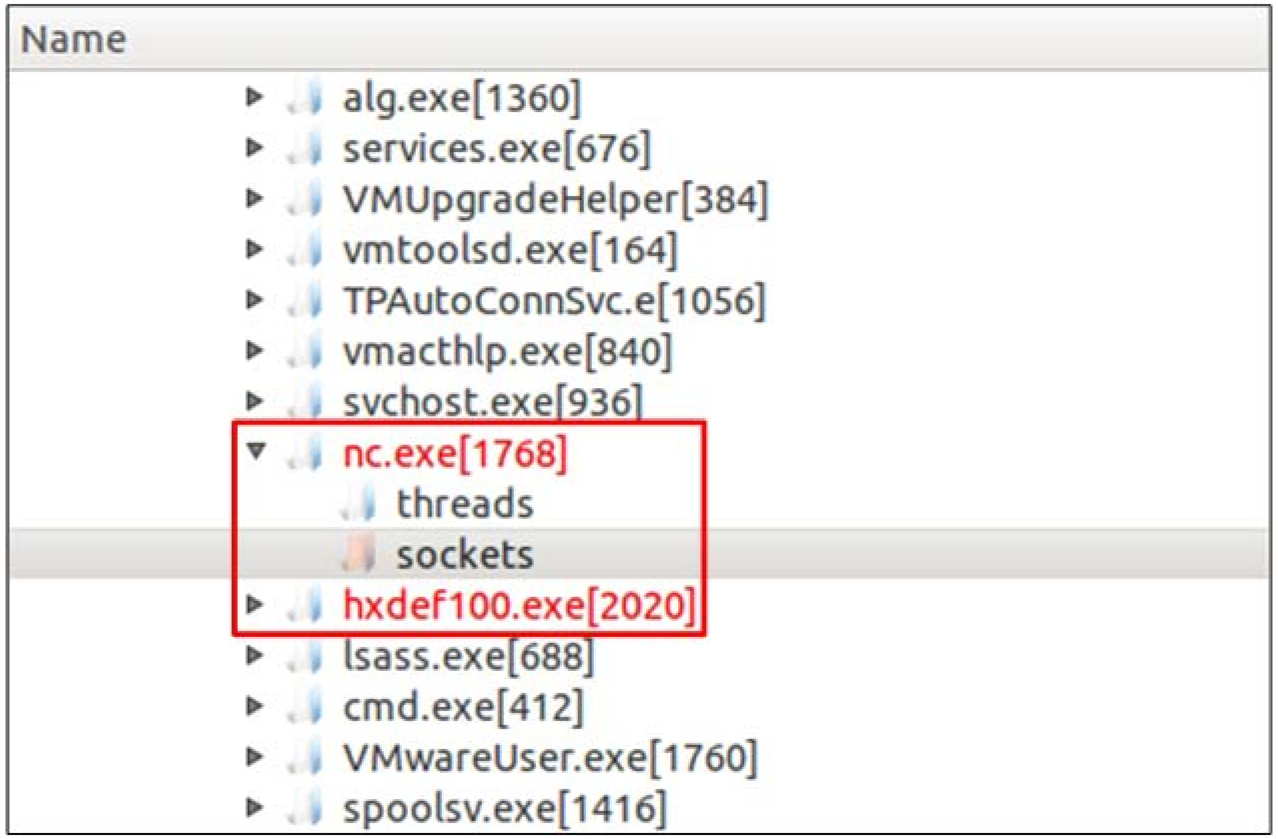
(Source: Vömel and Freiling, 2011)

Mode of Operation

- Detection Capabilities
  - With the help of the *cross viewing* approach, the following system manipulations can be detected:
    - Hidden processes, threads, and network connections
    - Installed hooks and notification routines
    - Maliciously inserted libraries
    - Maliciously injected code
    - Rogue system services

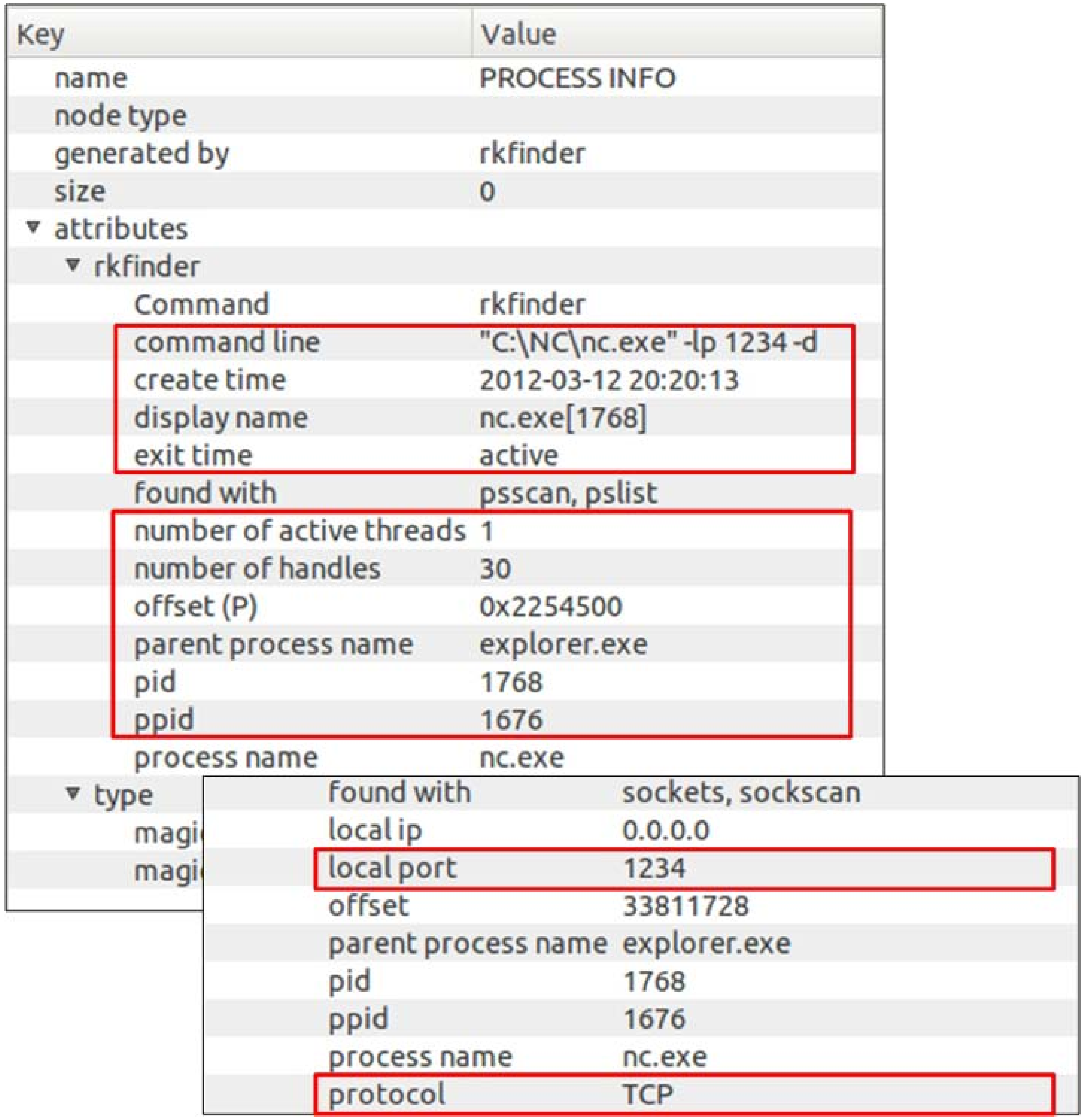
Mode of Operation

- Example: Detection of Hidden Processes



Name

- ▸ alg.exe[1360]
- ▸ services.exe[676]
- ▸ VMUpgradeHelper[384]
- ▸ vmtoolsd.exe[164]
- ▸ TPAutoConnSvc.e[1056]
- ▸ vmacthlp.exe[840]
- ▸ svchost.exe[936]
- ▾ nc.exe[1768]
  - threads
  - sockets
- ▸ hxdef100.exe[2020]
- ▸ lsass.exe[688]
- ▸ cmd.exe[412]
- ▸ VMwareUser.exe[1760]
- ▸ spoolsv.exe[1416]



Key	Value
name	PROCESS INFO
node type	
generated by	rkfinder
size	0
▾ attributes	
▾ rkfinder	
Command	rkfinder
command line	"C:\NC\nc.exe" -lp 1234 -d
create time	2012-03-12 20:20:13
display name	nc.exe[1768]
exit time	active
found with	psscan, pslist
number of active threads	1
number of handles	30
offset (P)	0x2254500
parent process name	explorer.exe
pid	1768
ppid	1676
process name	nc.exe
▾ type	
magi	
magi	

found with	sockets, sockscan
local ip	0.0.0.0
local port	1234
offset	33811728
parent process name	explorer.exe
pid	1768
ppid	1676
process name	nc.exe
protocol	TCP

Evaluation

- The performance of *rkfinder* was evaluated in a preliminary study
  - Systems were infected with 6 rootkits that are commonly found “in the wild”
  - Rootkits were configured to hide specific processes and other system resources, e.g., network sockets or system services
  - A memory snapshot of the infected system was taken and analyzed by rkfinder on a trusted workstation
  - Objective: Identify and highlight all rootkit-related system manipulations

Evaluation

- Overview of the Evaluation

Rootkit	Type	Supports Process Hiding	Supports Registry Key Hiding	Supports Socket Hiding	Supports Service Hiding	Supports Driver Hiding
BH-Rootkit-Nt	Kernel-Level	√	-	√	-	-
FU	Kernel-Level	√	-	-	-	√
FUTo	Kernel-Level	√	-	-	-	√
Hacker Defender	User-Level	√	√	√	√	√
NTIllusion	Library-Level	√	√	√	-	-
Vanquish	Library-Level	√	√	-	√	-

Evaluation

- Performance Results for *rkfinder*

Rootkit	Type	Process Detection	Registry Key Detection	Socket Detection	Service Detection	Driver Detection
BH-Rootkit-Nt	Kernel-Level	√	n/a	√	n/a	n/a
FU	Kernel-Level	√	n/a	n/a	n/a	-
FUTo	Kernel-Level	√	n/a	n/a	n/a	-
Hacker Defender	User-Level	√	-	√	√	-
NTIllusion	Library-Level	√	-	√	n/a	n/a
Vanquish	Library-Level	√	-	n/a	√	n/a

Evaluation

- Detection Rates of *rkfinder*

Rootkit	Employed by	Detection Rate
Kernel-Level Process and Network Manipulation	FU, FUTo	2/2
Hooking	BH-Rootkit-Nt, Hacker Defender, NTIllusion, Vanquish	4/4
Library Injection	NTIllusion, Vanquish	1/2
Code Injection	NTIllusion, Vanquish	2/2
Service Manipulation	Hacker Defender, Vanquish	2/2

Evaluation

- Weaknesses and Limitations of the Plug-In
  - Not all highlighted objects necessarily indicate a system threat
    - e.g., function hooks are frequently installed by legitimate security applications as well
  - Certain consistency checks may be subverted with anti-forensic techniques
    - False negatives may tempt users to get a false sense of the system state and the level of security
  - Not all types of rootkits can be discovered (e.g., virtualized rootkits such as *Blue Pill*)

Future Research

- Opportunities for Future Research
  - Extend the study and include more modern and sophisticated malware species in the evaluation
    - Integrated *Yara* malware classification utility can be used to distinguish families of malicious software
  - Add support for analyzing the Windows registry
    - e.g., examine well-known *run* keys that are frequently used to automatically start malware at boot time
  - Include certain heuristics to increase the detection quality
    - e.g., parent-child hierarchy, list of access privileges, etc.

Summary

- Summary and Conclusion
  - *rkfinder* permits examining forensic memory snapshots upon traces of potential rootkits
    - System inconsistencies that possibly indicate a system infection are identified by using a cross view approach
    - Suspicious objects are automatically highlighted in a graphical user interface
  - The plug-in particularly aims at users with only little forensic expertise
    - More sophisticated cases may still require the help and support of experienced investigators though

In case of any questions, please feel free to contact:

Stefan Vömel

http://www1.informatik.uni-erlangen.de/
stefan.voemel@cs.fau.de

i1 Chair for IT Security Infrastructures

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT