# Forensische Analyse des Windows-Arbeitsspeichers.


Andreas Schuster
Deutsche Telekom AG
Konzernsicherheit
andreas.schuster@telekom.de

# Forensische Analyse des Windows-Arbeitsspeichers.
## Agenda.

1. Einführung
2. Datensicherung
3. Auswertung
   - 3.1. Stand der Technik
   - 3.2. Suche nach Prozessen und Threads
4. Anwendungsbeispiele
5. Fazit

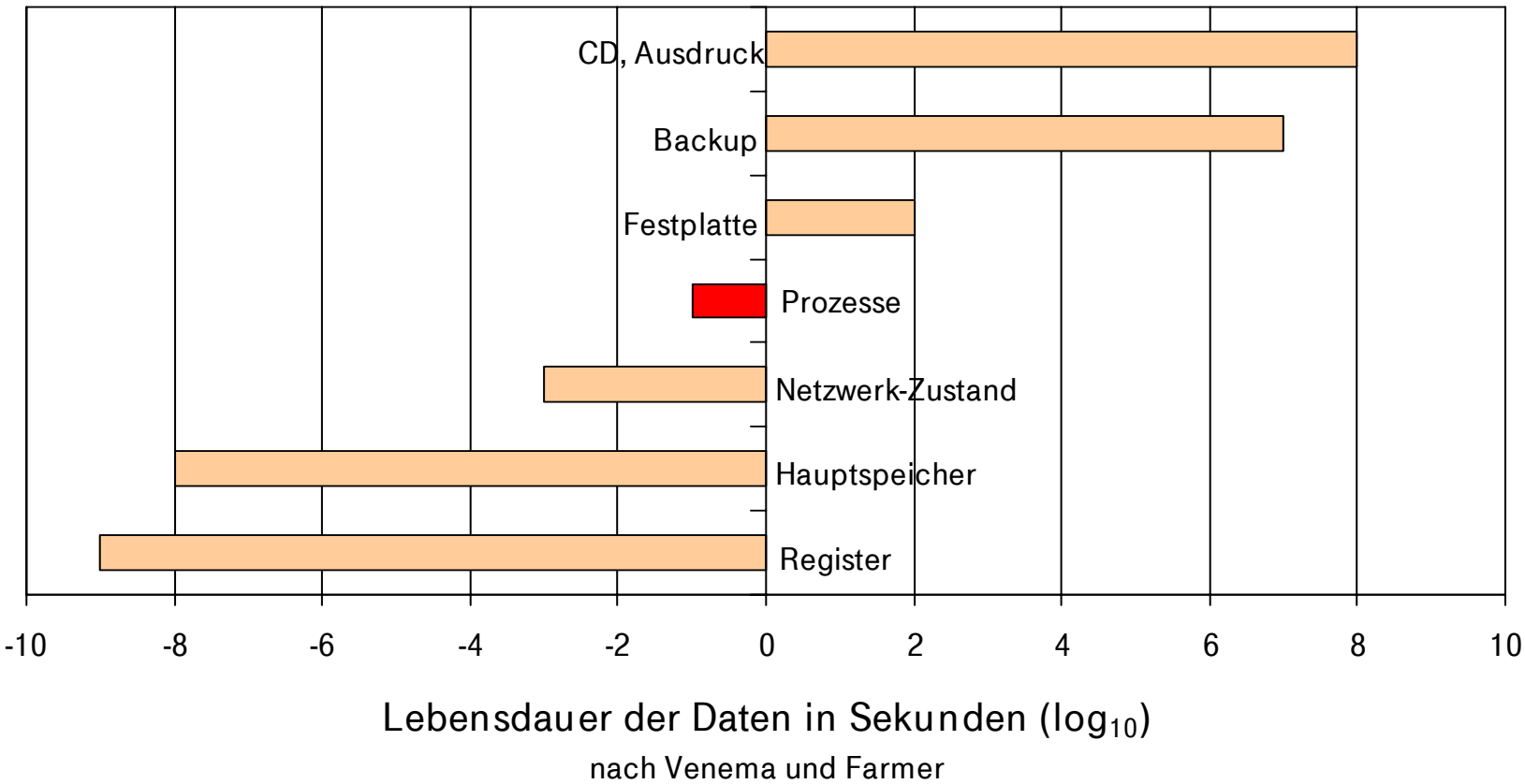
# Einleitung.
## Ziele einer forensischen Untersuchung.

Fragen in der digitalen Forensik:

nach A. Geschonneck (2004)

- **Wer** hat
- **was**
- **wo**
- **wann** → Start- und Endzeiten
- **womit** → Prozesse und Threads
- **wie** und
- **weshalb** getan?

# Einleitung.
## Prozessinformationen sind flüchtige Daten.



# Enleitung.
## Incident-Response Toolkits.

Programme:

- Dienstprogramme des Betriebssystems aus vertrauenswürdiger Quelle.
- Foundstone (z.B. FPort)
- Sysinternals (z.B.Handles,PsList)
- Windows Forensic Toolchest. http://www.foolmoon.net/security/wft/

Kritik:

- Ausführung im Userland, hohe Anfälligkeit gegenüber Rootkits.
- Sicherung und Auswertung der Daten in einem Arbeitsschritt.
- Großteil der Veränderung des Arbeitsspeichers wäre vermeidbar!

# Datensicherung.
## Ein Abbild des physischen Arbeitsspeichers.

Kopie von \\.\Device\PhysicalMemory mit "dd":

- Standardverfahren der Literatur.
- Ohne Zukunft: ab Windows 2003 SP 1 kein Zugriff mehr.

Livedump mit Sysinternals LiveKd und Debugger:

- In-Vivo, erfordert aber Symboldateien und Administratorrechte.
- Weiterverarbeitung mit Microsofts Debuggern ist möglich.

Crashdump:

- Post-mortem, liefert deshalb ein "scharfes" Abbild.
- Weiterverarbeitung mit Microsofts Debuggern ist möglich.
- Konfiguration siehe http://support.microsoft.com/kb/244139/en-us.

# Auswertung.
## Stand der Technik – Enumeration von Listen.

Prinzip:

- Objekte des Kernels sind durch Listen vielfach verbunden, Kernel-Variablen zeigen auf Anfänge der Listen.

Programme:

- Windows Memory Forensic Toolkit (M. Burdach, 2005)
- MemParser (C. Betz, 2005)
- kntlist (G. M. Garner und R.-J. Mora, 2005)

Kritik:

- Alle Programme sind (in unterschiedlichem Maße) anfällig gegenüber Direct Kernel Object Manipulation (DKOM).
- Beendete Prozesse und Threads werden prinzipbedingt nicht erkannt.

# Auswertung.
## Suche nach Objekten.

Analogie zur Untersuchung von Dateisystemen:

- Wie stellt man gelöschte Dateien wieder her?
- Suche nach (Meta-) Daten.

Anforderungen:

- Die Kriterien müssen spezifisch sein, um false positives zu vermeiden.
- Die Kriterien müssen sich auf essentielle Informationen stützen, um eine Manipulation durch den Täter zu vermeiden.

Beispiele ungeeigneter Kriterien:

- ASCIIZ-String mit Programm-Namen.
- Accounting-Informationen (IO-Volumina, Zeitstempel).

# Auswertung.
## Suche nach Objekten.

DISPATCHER _HEADER:

- UChar Type
  3 = PROCESS
  6 = THREAD
  ...
- UChar Absolute
  immer 0?
- UChar Size
  konstant für Type und Betriebssystemversion.
- UChar Inserted
  immer 0?

# Auswertung.
## Suche nach Objekten.

- DISPATCHER_HEADER an mehreren Stellen der gesuchten Struktur.
- Alignment: Strukturen sind auf Grenzen von 8 Byte ausgerichtet.

Speziell für Prozesse:

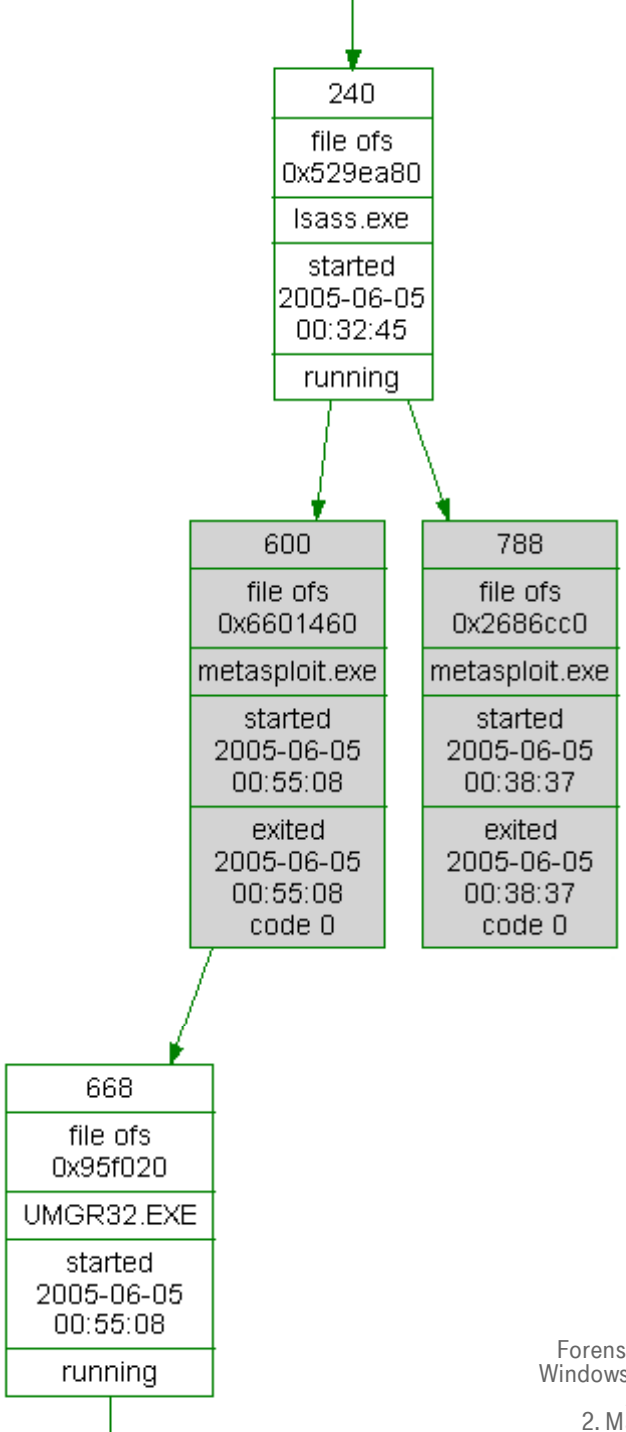
- PageDirectoryBaseAddress != 0.
- PageDirectoryBaseAddress zeigt auf Beginn einer Speicherseite.
- ThreadListHead.{Flink | Blink} zeigt in den Speicherbereich des Kernels.

Speziell für Threads:

- ThreadsProcess zeigt in den Speicherbereich des Kernels.
- StartAddress != 0
  Ausnahme: Idle-Thread (PID 0).

# Anwendungsbeispiele.
## Aufklärung eines Angriffs.

- Ungewöhnlich:
  Das Local Security Authority Subsystem (LSASS) hat 2 Kindsprozesse erzeugt.
- Die Kindsprozesse tragen den Namen metasploit.exe, ein Hinweis auf den gleichnamigen Exploit-Baukasten http://www.metasploit.org/
- Aber:
  Namen sind nicht vertrauenswürdig.



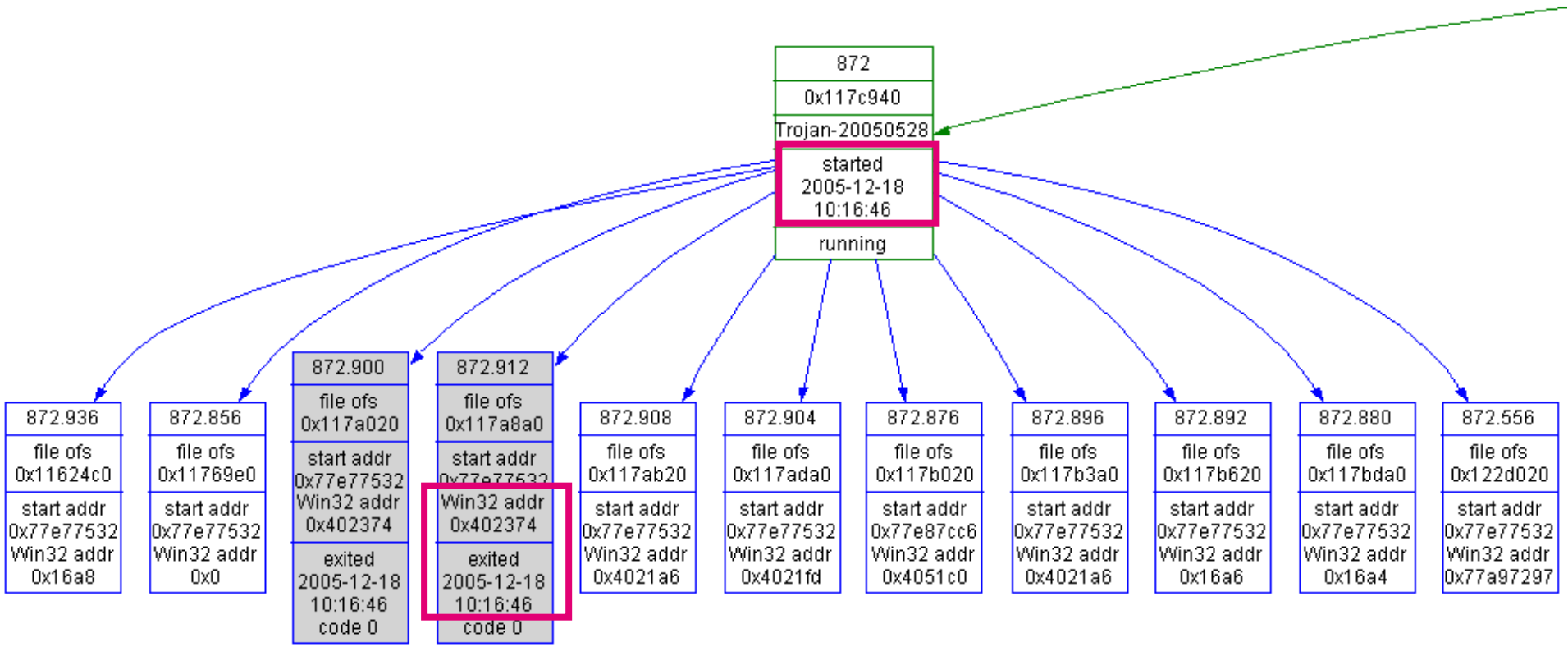
# Anwendungsbeispiele.
## Datierung einzelner Aktionen.

Die Kontrollstruktur für Threads (ETHREAD) ermöglicht die Verknüpfung

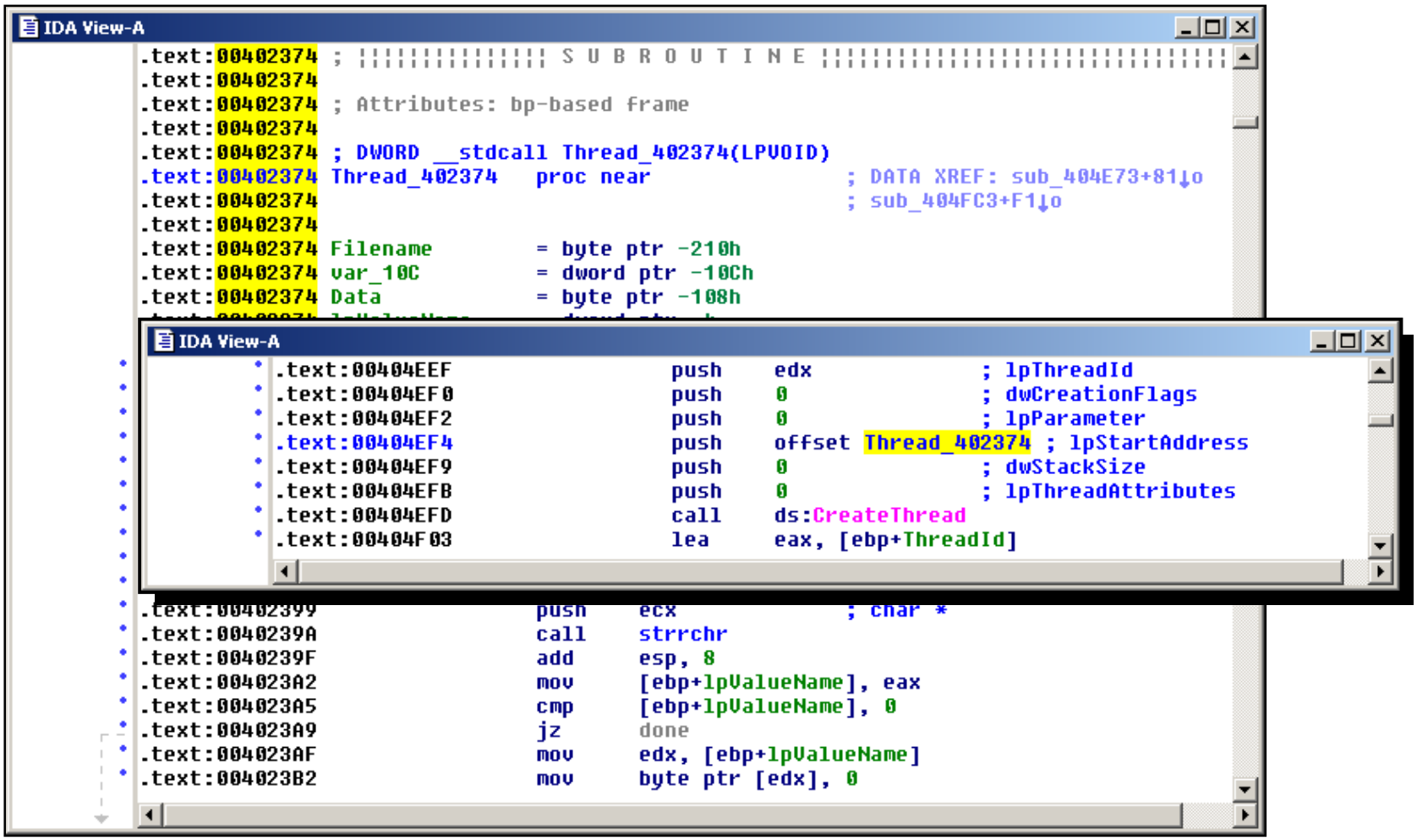
- der (Start-) Adresse einer Routine
- mit dem Endzeitpunkt ihrer Ausführung.

Der Speicherplatz für den Startzeitpunkt kann während der Ausführung des Threads für andere Zwecke verwendet werden – Datenverlust!

# Anwendungsbeispiele.
## Datierung einzelner Aktionen.



# Anwendungsbeispiele.
## Datierung einzelner Aktionen.



# Anwendungsbeispiele.
## Stimmen aus dem Jenseits.

Testlauf mit einem Speicherabbild der DFRWS Memory Analysis Challenge:

| No. | Type | PID | Time created | Offset | Remarks |
|---|---|---|---|---|---|
| 7 | Proc | 176 | 2005-06-05 00:32:44 | 0x0001045d60 | winlogon.exe |
| 8 | Proc | 176 | 2005-06-04 23:36:31 | 0x0001048140 | winlogon.exe |
| 9 | Proc | 164 | 2005-06-03 01:25:54 | 0x000104ca00 | winlogon.exe |
| 10 | Proc | 180 | 2005-06-05 00:32:43 | 0x0001286480 | csrss.exe |
| 11 | Proc | 168 | 2005-06-03 01:25:53 | 0x0001297b40 | csrss.exe |

kntlist bestimmt aus der Variablen KeBootTime den Startzeitpunkt zu 2005-06-05 00:32:27

Einige Spuren stammen wahrscheinlich von einem früheren Systemstart.

Aber: Das ist ein seltener Einzelfall!

# Fazit.
## Ein Vergleich mit listenbasierten Verfahren.

- Die Qualität der Datensicherung ist entscheidend für den Erfolg der Untersuchung.
- Die Suche identifiziert auch Objekte, die listenbasierte Verfahren nicht finden können.
- Listenbasierte Verfahren, insbesondere kntlist, liefern jedoch detailliertere Informationen.

- **Optimal ist die Kombination beider Verfahren.**

# Fazit.
## Weitere Materialien.

- Hintergrundinformationen.
- Anwendungsbeispiele.
- Definitionen von EPROCESS / ETHREAD für diverse Versionen von Windows 2000 bis zu Longhorn / Vista.
- PTfinder:
  Proof-of-Concept Code für Windows 2000,
  findet Prozesse und Threads in Speicherabbildern
  (dd, Crashdump, VMware VMSS).

- http://computer.forensikblog.de/

# Vielen Dank für Ihre Aufmerksamkeit.


Andreas Schuster
Deutsche Telekom AG
Konzernsicherheit
andreas.schuster@telekom.de