Video surveillance: A new forensic model for the forensically sound retrieval of picture content off a memory

Stefan Kiltz, Tobias Hoppe, Jana Dittmann

Otto-von-Guericke-University Magdeburg
Faculty of Computer Science
Research Group on Multimedia and Security
Outline

• Motivation
• Our forensic model
• Practical application
  – Data gathering
  – Data investigation
• Automated graphics extraction
• Extraction of system information
• Conclusion
Motivation

- Video surveillance widely used today
- Challenge: Providing a system for small office and home users to gather and extract forensically sound picture evidence using existing technology; OpenSource, freeware and self-written programmes
- Sensor identification based on the CCD chip of the camera only ensures authenticity and is vulnerable to false positives and false negatives
- Solution: tying the picture content to the machine that contains it by extracting it off main memory (RAM) and provide proof of integrity by checksumming the memory dump as one atomic step
Motivation

- Security aspects of integrity and authenticity to be met throughout the process
- Assumptions:
  - System (HW/SW) has not been tampered with by the operator during the creation of the memory dump
  - Secure storage of the retrieved memory dump
- Suggested solution is robust against manipulation after the data gathering process (memory image)
- Presented solution is based on a self-developed model of the forensic process
Our forensic model

- Separated into *Phases, Classes of methods, Forensic data types*
- *Phases* (mutual exclusive) are used to model sequence details during a forensic investigation, not a new approach but novel phase of *strategic preparation* is included, being beneficial for the operator of an IT-system conducting a forensic investigation
- *Classes of methods* (mutual exclusive) classify forensic capabilities of software, not only dedicated forensic suites gather forensically relevant data (e.g. a database application) - ensures independence from particular software solutions
- *Forensic datatypes*, a layered approach similar to ISO/OSI model (not mutual exclusive), used to determine input and output data of forensic tools/methods, describe the forensically relevant data as a data source
Our forensic model (cont’d)

- Phases addressed in our approach:
  - Strategic preparation (SP)
  - Operational preparation (OP)
  - Data gathering (DG)
  - Data investigation (DI)

- Classes of methods in our approach:
  - Operating system (OS)
  - IT application (ITA)
  - Data Processing and Evaluation (DPE)

- Forensic datatypes in our approach
  - Hardware data ($DT_1$)
  - Raw data ($DT_2$)
  - User data ($DT_8$)
Practical application: gathering picture content on windows-based systems (1)

• Target: Gathering of picture content using a surveillance camera connected to a laptop overlooking a parking space

**Strategic preparation SP**
- win32dd.exe (*physical* memory)
- ProcMemDump.exe (*logical* memory)
- Md5deep.exe (Cryptographic sha-256 hash)
- Blat.exe (command line email)
- SecureDump.exe (Wrapper application)
- Stamper (time stamping service)

**Operational preparation OP**
- Volatility.py (process memory reconstruction)
- Irfanview (picture viewer)
Practical application: gathering screen-content on windows-based systems (2)

Data gathering (DG):

- Gathering of raw data ($DT_2$)
- Possible at two layers
  - Physical memory (win32dd.exe)
    - Equal in size to amount of RAM
    - Not necessarily consistent
  - Virtual process memory (ProcMemDump.exe)
    - Guaranteed to belong to target process
    - Including swapped-out data
    - Reduces search domain dramatically
- Subsequent hash generation for Integrity checks
Practical application: gathering screen-content on windows-based systems (3)

Virtual vs. physical memory acquisition:

- **Acquisition of logical / process memory (ProcMemDump)**
  - No/little fragmentation of contents due to virtual memory management
  - Dump also includes non-RAM contents like mapped files or content swapped out to disk
  - Smaller size
  - Collection
  - Dump does only contain data belonging to the target process
  - Dump does not contain any formerly freed pages
  - Dump file potentially inconsistent due to duration of the acquisition process

- **Acquisition of physical / main memory (win32dd)**
  - Also contains contents of freed pages until overwritten
  - Dump does also contain information from the system and other processes
  - High fragmentation complicating resource identification
  - Low-frequented pages may not be included (requires swapfile analysis)
  - Dump will be inconsistent due to duration of the acquisition process and inherent changed imposed by its program code
Practical application: securing the data gathering process

- Self-developed wrapper application SecureDump:
  - Waits for predefined keystroke, initiates the memory dump creation calling win32dd
  - Calls sha256 to calculate the hash sum
  - Initiates logical memory collection using ProcMemDump
  - Calls sha256 to calculate the hash sum
  - Executes msinfo32 to create hardware inventory
  - Calls sha256 to calculate the hash sum
  - Sends logfile (checksummed with sha256) of the process using blat via e-mail to digital timestamping service stamper

- One atomic step ensuring integrity and supports authenticity evaluation mechanisms
Practical application: securing the data gathering process

- Reply mail from the *stamper* service is sent to multiple recipients
- Stamper uses PKI to ensure authenticity and integrity
- Resulting mail contains the hash sums, sender address, confirmation of the time of reception
- Modification of the dump file, the inventory file or reports can be easily identified
- Assumption: digital timestamping service is trustworthy
Practical application: gathering screen-content on windows-based systems (4)

Data investigation (DI):
- Identification of graphics content ($DT_8$) within raw data ($DT_2$)
- Support tool: Freeware picture viewer *IrfanView*
- Rename dump file extension to “raw” to open as raw image
- **A priori knowledge** about graphics format
  - Example (illustrated on the next slides):
    - Surveillance camera picture width of 640 pixels
    - Colour depth of 24 bpp
    - The dump file has a **file size of 120 MB**
      $\Rightarrow$ Set image height **70,000 pixels** ($70000 \times 640 \times 3 = 128.17 \text{ MB} > 120 \text{ MB}$)
      to achieve that the entire dump is included in the visualisation
Illustration of manual picture identification
In most cases it is necessary to **iteratively adjust** the other settings like colour order or bytes to skip to achieve an appropriate visualisation.
Automated graphics extraction

- Identification of graphics content ($DT_8$) within raw data ($DT_2$)
- Utilise prior knowledge about the raw data structure
- E.g. real-world true colour pictures $\Rightarrow$ multi channel representation. For example 24 Bit = 1 Byte per Pixel (colour order RGB, BGR etc.)
- Assumption: Correlation of 3-Byte groups not evident in arbitrary data
- Detection approach: Calculation of channel entropies among 4096 pixels
  $\Rightarrow$ Calculation of the variance among the 3 channel entropies for each frame
Practical application: gathering screen-content on windows-based systems (5)

Graphics extraction from physical memory dumps:
- The previous examples were performed on user-space dumps.
- Basically, this principle is also applicable to physical memory dumps.
- Virtual address space reconstruction using the Volatility Framework.
- Practical problems: Memory page fragmentation in kernel-space
  - Typical page size on Windows machines: 4096 Bytes
  - Larger amounts are not necessarily allocated in adjacent order.

Fragments of the Desktop picture (DT8) in physical memory (DT2) using desktop width of 1280 pixels

Fragments of the target picture (DT8) in main memory data

1280 px

(...)

(...)

Informatik 2009

17

2009/09/28
Extracting system information

- Necessary to tie memory dump to the recording machine -> authenticity
- ethernet address (mac) and laptop mainboard serial number as unique properties in memory dump ($DT_1$)
- Separated processes using the volatility framework
- Loaded into the HxD hexadecimal editor
- Found mac address in the system process (PID 4)
- Found serial number in svhost process (PID 1096)
Extracting system information
Conclusion and further work

• Forensically sound acquisition of main memory data with regards to *integrity* and *authenticity* using physical and logical memory dumps
• Data investigation to extract picture content off the memory dump (physical and logical) applying image processing techniques
• First steps towards automated preselection of image content
• Further research focuses physical to logical address space conversion and on automating the picture detection process further