About

Title:	Improving Digital Forensics and Incident Analysis in Production Environments by Using Virtual Machine Introspection
Student:	Benjamin Taubmann
Advisor:	Prof. Dr. Hans P. Reiser
Affiliation:	Assistant Professorship of Security in Information Systems University of Passau
Research Area:	System Security, Memory Forensics, Virtual Machine Introspection
Projects:	DINGfest (BMBF), ARADIA (DFG)

UNIVERSITÄT PASSAU

Motivation

Motivation

Why do we need digital forensics?

- Traditional crime investigation
- Incident analysis
- Malware analysis

What are the challenges?

- Higher security standards (Access)[a]
- High amounts of data (Semantic Gap)
- Performance (Information Extraction)
- Stealthiness (Tracing)

[a] https://motherboard.vice.com/en_us/article/5984jq/cops-dont-look-iphonex-face-id-unlock-elcomsoft

UNIVERSITÄT PASSAU

Types

- **Memory Forensics:** Forensics on (snapshots of) main memory to find sensitive information that is not stored on hard disk such as passwords, keys or rootkits
- **Virtual Machine Introspection:** Memory Forensics applied to running virtual machines

Advantages

- Access to raw, unencrypted data (e.g., key material)
- Isolation and forensic soundness
- Detailed tracing

Research Problems

Research Problems

A **Architecture:** How does a **generic approach** for computer forensics look like? What are the application requirements?

B **Data Acquisition:** How to gain **access** to the memory of production systems such as cloud environments or mobile devices?

C **Information Extraction:** How to **locate and extract** high level information efficiently from main memory?

D **Applications:** How to deploy and adapt VMI methods to the **requirements of real world use cases and modern computing systems**?

A How does a **generic approach** for computer forensics look like? What are the application requirements?

Requirements

UNIVERSITÄT PASSAU

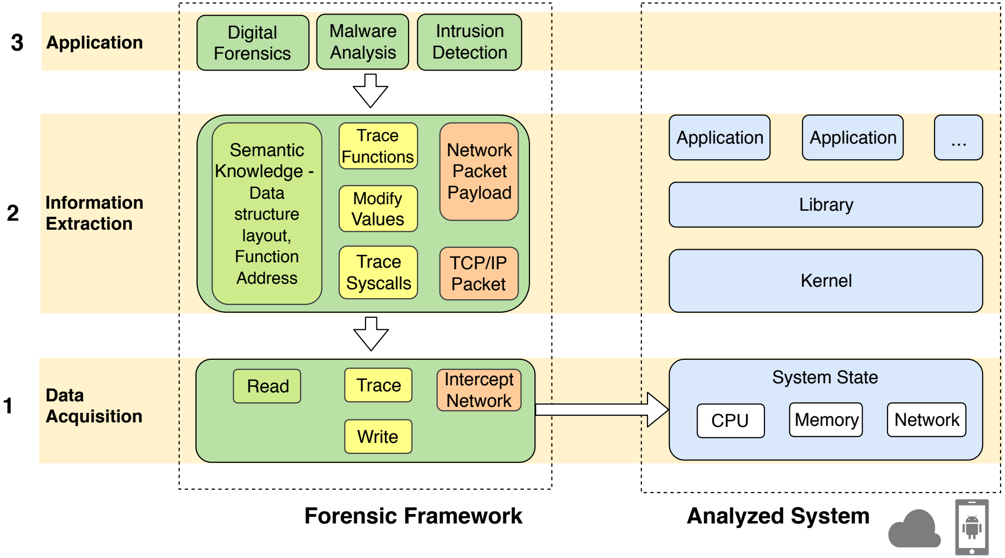
- **Off-line:** read memory and CPU registers, address translation
- **On-line:** write memory and CPU registers, control flow interception, manipulation, injection, access unmapped memory regions
- **File Access:** Read files (tmpfs, shm, encrypted fs)
- **Network Traffic Monitoring**

Application Requirements:

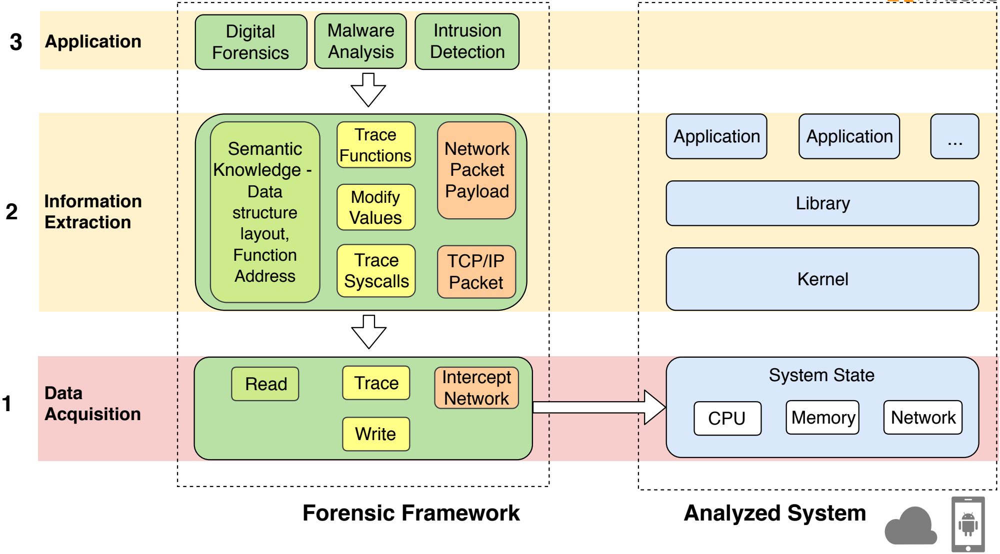
1. **Forensic Soundness:** Attackers MUST not interfere with the data acquisition process
2. **Security:** Forensic interface MUST not be a new attack surface
3. **Stealthiness:** Forensic analysis SHOULD NOT be noticeable from the analyzed system
4. **Stability:** Forensic analysis MUST NOT crash the analyzed system
5. **Platform Independence:** Forensic analysis SHOULD BE portable to other operating systems/hardware platforms
6. **Performance:** Forensic analysis SHOULD affect the performance of the analyzed system as little as possible
7. **Multiprocessor Support:** Tracing a system with multiple CPUs SHOULD be possible

Depending on the use case, some are more important than others

Architecture



Data Acquisition



UNIVERSITÄT PASSAU

B How to gain **access** to the memory of production systems such as cloud environments or mobile devices?

UNIVERSITÄT PASSAU

How to get access to the memory of production systems such as cloud environments or mobile devices?

Challenges:

- **Generic Interface** for different systems
- **Forensic Soundness:** access to raw untampered memory without using OS functions
- **Security:** do not introduce new attack surface

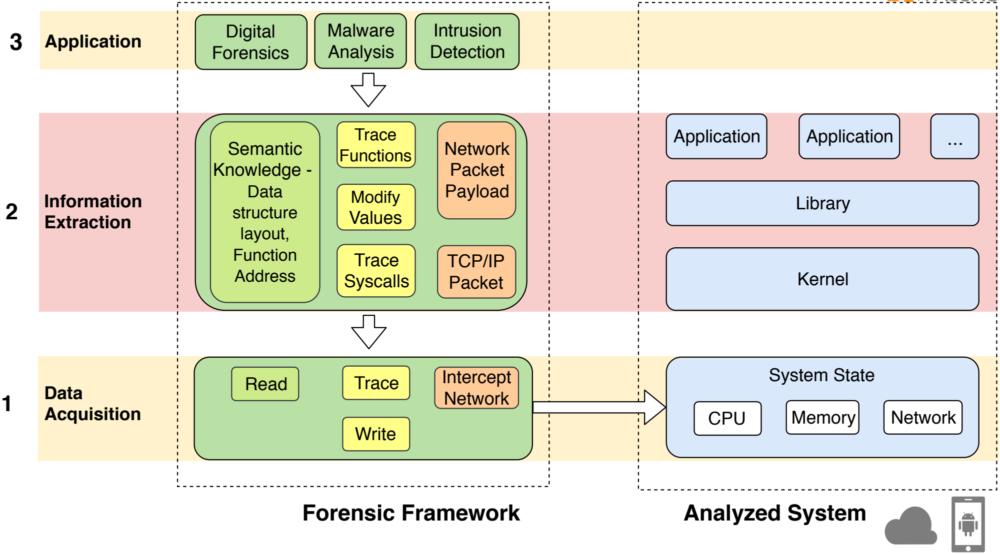
1. **Mobile Devices:**
   - **SOTA:**
     - Mobile devices have a high level of security
     - Coldboot attacks tools overwrite kernel data structures
   - **Contribution:** Minimal bare-metal application to access memory and transfer it to analysis PC[1]
2. **IaaS-based cloud computing:**
   - **SOTA:** No VMI support for cloud costumers
   - **Contribution:** Extended cloud management and the hypervisor so that cloud costumers can do VMI on their VMs[2]

[1] Taubmann, Benjamin et al. "A Lightweight Framework for Cold Boot Based Forensics on Mobile Devices." In: *ARES*. 2015.

[2] Taubmann, Benjamin, Noelle Rakotondravony, and Hans P. Reiser. "CloudPhylactor: Harnessing Mandatory Access Control for Virtual Machine Introspection in Cloud Data Centers." In: *IEEE TrustCom-16*. 2016.

C How to **locate and extract** high level information efficiently from main memory?

Information Extraction



Data Extraction

How and when to locate and extract high level information efficiently from main memory?

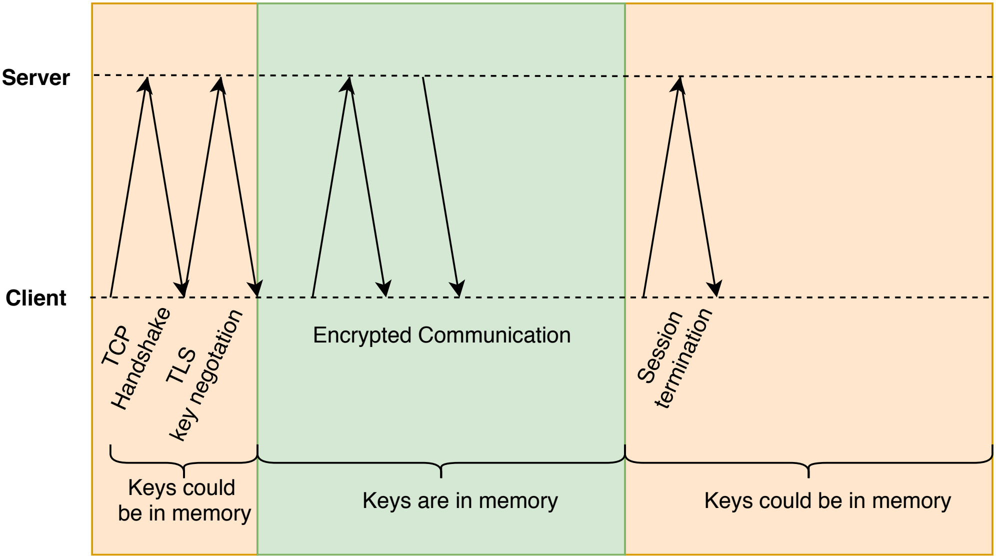
Example: Extracting sessions keys from memory in order to decrypt TLS encrypted network communication of

- Malware (in virtual machines)
- Persons using chat applications (mobile devices)

Requirements

- No modification of the application
- No modification of the network traffic
- Without knowing application logic
- Support of *perfect forward secrecy (PFS)*

Motivating Example



a **When to extract data?**

- state based (e.g., from network traffic)
- control flow based (e.g., when functions are called)
- time based (e.g., every second)

b **How to locate information?**

- the data (regular expression, entropy, etc.)
- the data structures storing the data (offset in data structures, type of data structure)
- the control flow (a function that directly accesses data)

c **How to get semantic knowledge?**

- From source code/debugging information
- By regenerating from main memory
- By regenerating from CPU instructions

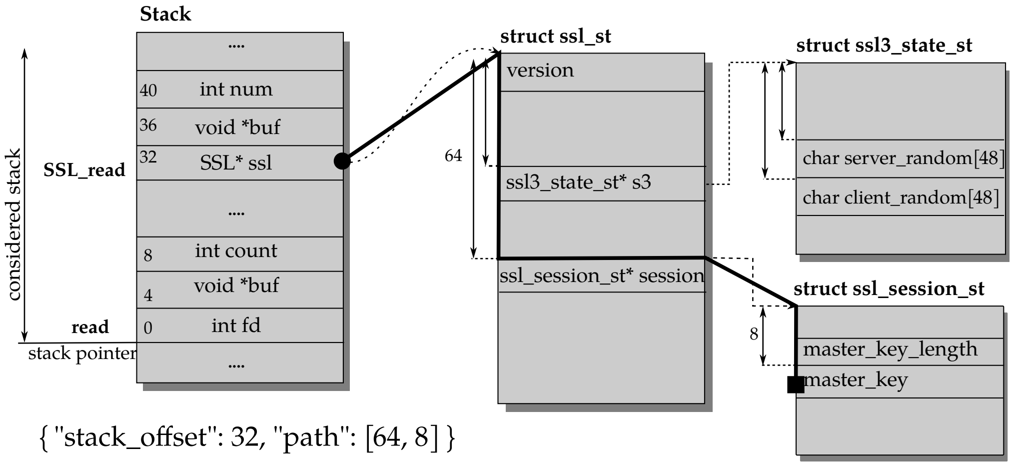


Figure: The contents on the stack when the `read` function is called by the `SSL_read` of OpenSSL function. The path from the starting point – the SSL pointer (black dot) to the MS (black square) – is marked bold and the corresponding. The computed path and the offset on the stack are noted on the bottom left side.

UNIVERSITÄT PASSAU

Decryption of TLS based communication

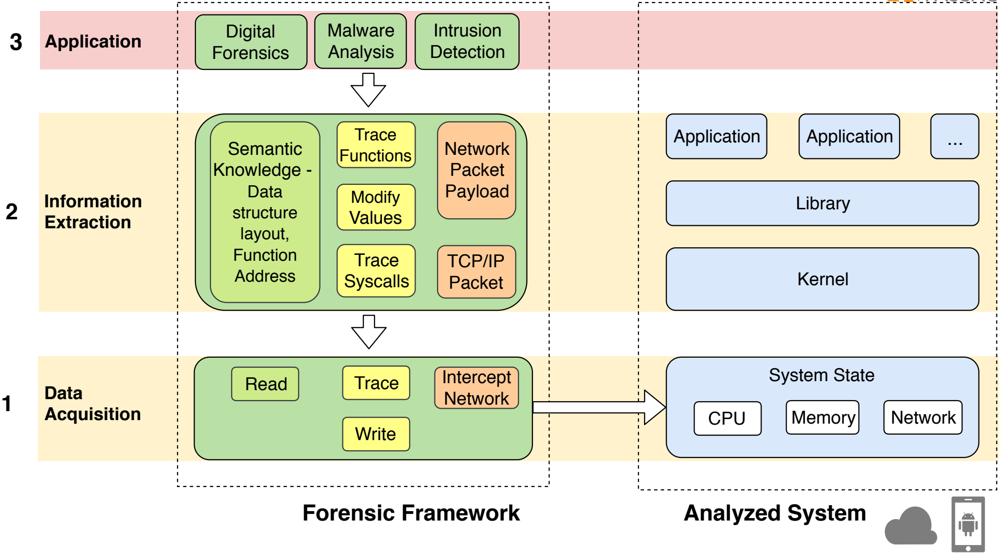
- **SOTA:** MitM based Proxy solutions
- **Contributions:**
  - Approach to extract TLS session keys from main memory of virtual machines[3]
  - Derive semantic knowledge about data structures from memory snapshots[4]
  - Improve performance of key extraction by intercepting the control flow of Android applications

[3] Taubmann, Benjamin et al. "TLSkex: Harnessing virtual machine introspection for decrypting TLS communication." In: *DFRWS EU*. 2016.

[4] Taubmann, Benjamin, Omar Al Abduljaleel, and Hans P. Reiser. "DroidKex: Fast Extraction of Ephemeral TLS Keys from the Memory of Android Apps." In: *DFRWS USA*. 2018.

D How to deploy and adapt VMI methods to the **requirements of real world use cases and modern computing systems**?

Applications and Requirements



How to deploy and adapt VMI methods to the requirements of real world use cases and modern computing systems?

Advantages:

- Stealthiness
- Isolation
- Forensic Soundness

Challenges:

- **Overhead:** VMI-based tracing can be slow
- **Level of detail:** Extraction of more information slows down the process
- **Large amount of information:** many logs

Intrusion Detection System

- **SOTA:** VMI-based tracing is too slow for production environments
- **Contribution:**
  - **Trade-off** between detailed tracing and performance: lightweight tracing to detect intrusions, heavyweight tracing for incident analysis[5][6]

Honeypots

- **SOTA:** SSH Honeypots are easy to detect
- **Contribution:** Implementation of a stealthy VMI-based honeypot[7]

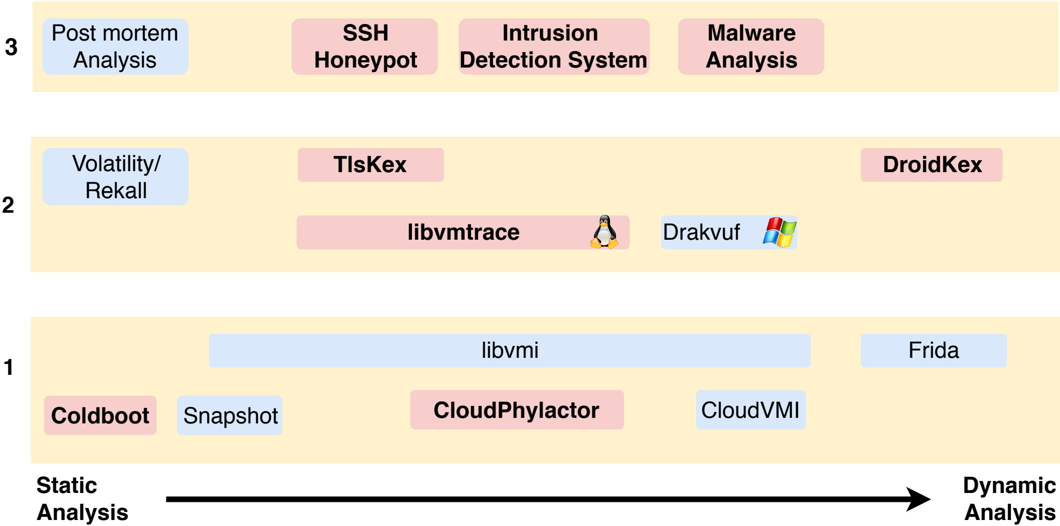
[5] Andres Fischer et al. "CloudIDEA: A Malware Defense Architecture for Cloud Data Centers." In: *C&TC 2015*. 2015.

[6] F. Menges, F. Böhm, M. Vielberth, A. Puchta, B. Taubmann, N. Rakotondravony, T. Latzo. "Introducing DINGfest: An architecture for next generation SIEM systems." In: *GI Sicherheit 2018 (Short Paperbt)*.

[7] Stewart Sentanoe, Taubmann, Benjamin, and Hans P. Reiser. "Sarracenia: Enhancing the Performance and Stealthiness of SSH Honeypots using Virtual Machine Introspection." In: *NordSec 2018*. 2018.

Summary

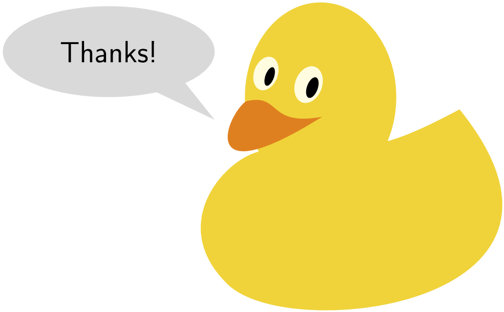
Contributions (bold red)



Conclusion

We showed:

- that TLS connections of virtual machines can be dencrypted
- that SSH sessions of a virtual machine can be monitored
- how VMI can be used in cloud environments and on mobile phones


Thanks!

Publications

[1] Taubmann, Benjamin, Manuel Huber, Lukas Heim, Georg Sigl, and Hans P. Reiser. "A Lightweight Framework for Cold Boot Based Forensics on Mobile Devices." In: *ARES*. 2015.

[2] Taubmann, Benjamin, Noelle Rakotondravony, and Hans P. Reiser. "CloudPhylactor: Harnessing Mandatory Access Control for Virtual Machine Introspection in Cloud Data Centers." In: *IEEE TrustCom-16*. 2016.

[3] Taubmann, Benjamin, Christoph Frädrich, Dominik Dusold, and Hans P. Reiser. "TLSkex: Harnessing virtual machine introspection for decrypting TLS communication." In: *DFRWS EU*. 2016.

[4] Taubmann, Benjamin, Omar Al Abduljaleel, and Hans P. Reiser. "DroidKex: Fast Extraction of Ephemeral TLS Keys from the Memory of Android Apps." In: *DFRWS USA*. 2018.

[5] Andres Fischer, Thomas Kittel, Bojan Kolosnjaji, Tamas K Lengyel, Waseem Mandarawi, Hans P Reiser, Taubmann, Benjamin, Eva Weishäupl, Hermann de Meer, Tilo Müller, and Mykola Protsenko. "CloudIDEA: A Malware Defense Architecture for Cloud Data Centers." In: *C&TC 2015*. 2015.

[10] F. Menges, F. Böhm, M. Vielberth, A. Puchta, B. Taubmann, N. Rakotondravony, T. Latzo. "Introducing DINGfest: An architecture for next generation SIEM systems." In: *GI Sicherheit 2018 (Short Paperbt)*.

[0] Stewart Sentanoe, Taubmann, Benjamin, and Hans P. Reiser. "Sarracenia: Enhancing the Performance and Stealthiness of SSH Honeypots using Virtual Machine Introspection." In: *NordSec 2018*. 2018.

[6] Taubmann, Benjamin and Bojan Kolosnjaji. "Architecture for Resource-Aware VMI-based Cloud Malware Analysis." In: *SHCIS'17*. 2017.