Bringing Forensic Readiness to Modern Computer Firmware

By:
Tobias Latzo, Florian Hantke, Lukas Kotschi and Felix Freiling

From the proceedings of

The Digital Forensic Research Conference

DFRWS EU 2021

March 29 - April 1, 2021

Tobias Latzo, Florian Hantke, Lukas Kotschi, Felix Freiling

BRINGING FORENSIC READINESS TO MODERN COMPUTER FIRMWARE

Security Research Group
Department of Computer Science
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)

MOTIVATION I

- Memory analysis is important in today's digital investigations
- It is considerable hard to acquire system's memory
  - Vömel and Freiling [1]: ***Correctness***, ***atomicity***, *and* ***integrity***: *defining criteria for forensically-sound memory acquisition*
  - Pagani et al. [2]: *Introducing the Temporal Dimension to Memory Forensics*

[1] S. Vömel and F. Freiling. *Correctness, atomicity, and integrity: defining criteria for forensically-sound memory acquisition.* Digital Investigation 9(2), 125–137. 2012.

[2] F. Pagani, O. Fedorov, and D. Balzarotti. *Introducing the Temporal Dimension to Memory Forensics*. ACM Transactions on Privacy and Security 22(2), Article 9. 2019.

MOTIVATION II

- Today's memory acquisition often makes use of OS kernel functionality
- Memory acquisition from lower layers is beneficial [3]
- System security is getting better

It's hard to deploy forensic acquisition software post-incident on lower layers without tampering with evidence.

[3] T. Latzo, R. Palutke, and F. Freiling. *A universal taxonomy and survey of forensic memory acquisition techniques.* Digital Investigation 28, 56–69. 2019.

UNIFIED EXTENSIBLE FIRMWARE INTERFACE (UEFI)

- Successor of PC-BIOS
- Boots directly into protected mode or long mode
- GUID Partition Table (GPT)
- Network stack
- Secure boot
- Reference implementation: EDK 2

Let's make use of UEFI's capabilities for memory acquisition.

UEberForensIcs

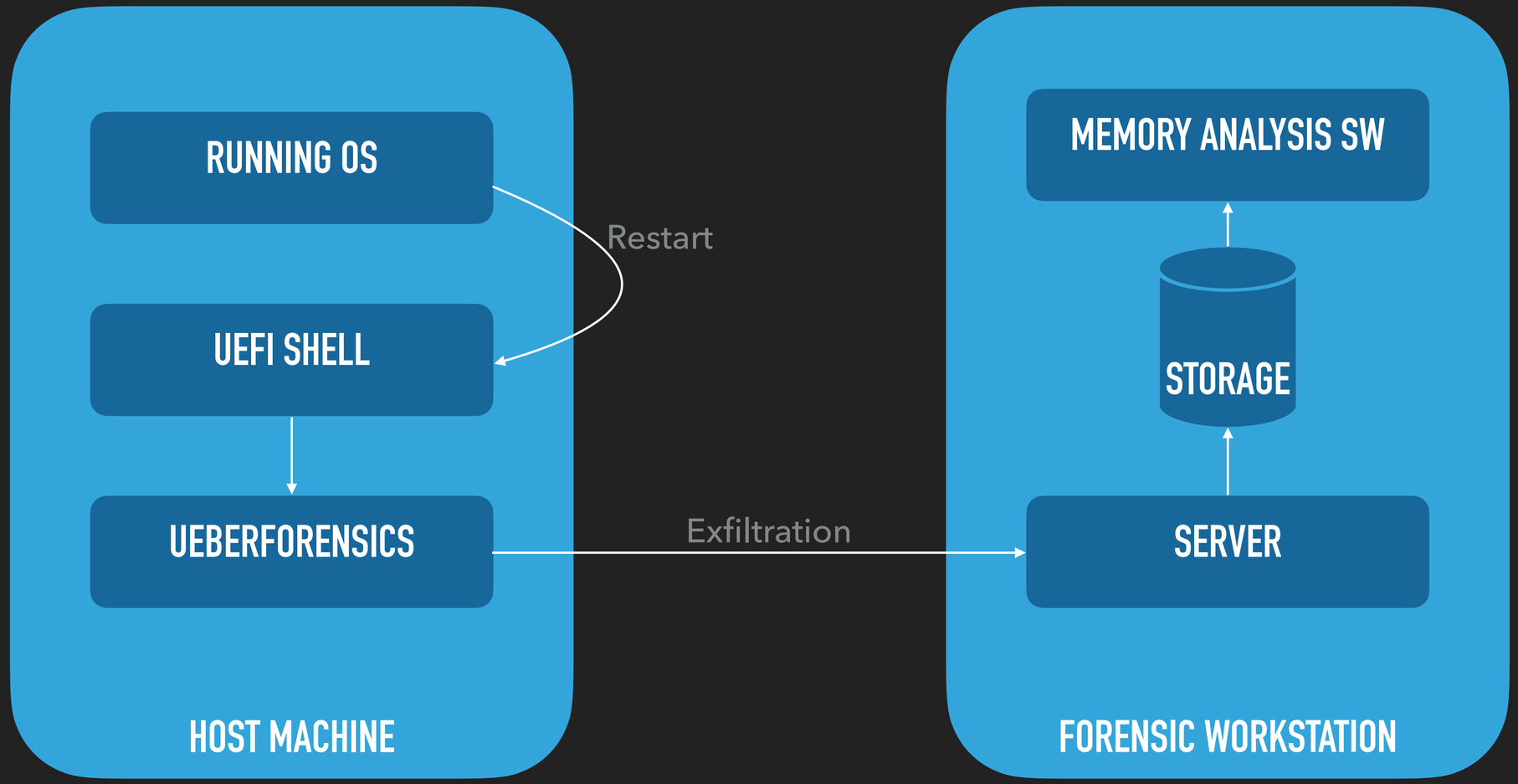
UEFI built-in memory forensics

IDEA

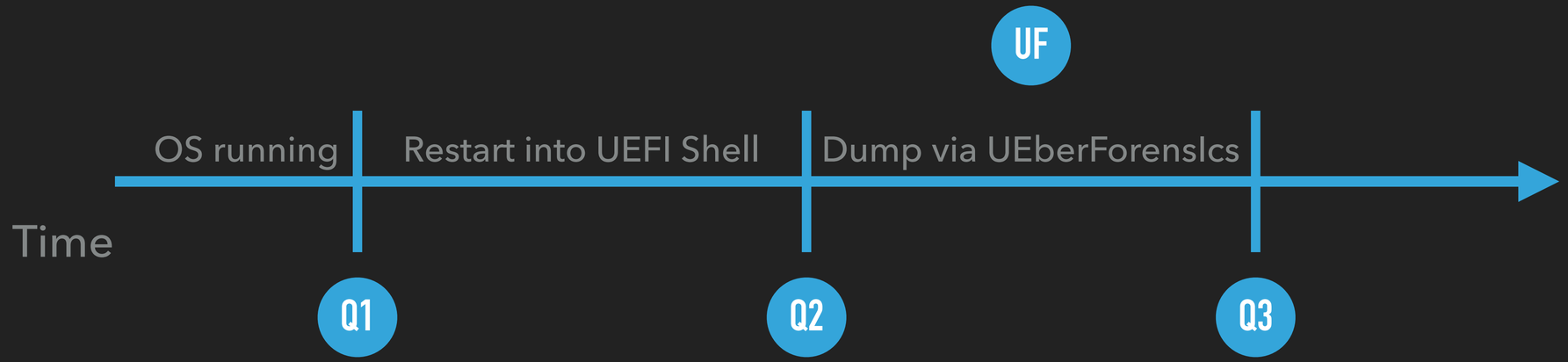
- Similar to cold boot [4]
  - restart computer
  - open UEFI Shell and dump over the network
- OS-independent
- Running processes are immediately stopped (atomicity)
- Anti-forensics software cannot tamper with evidence

[4] J. A. Halderman, S. D. Schoen, N. Heninger , W. Clarkson, W. Paul, J. A. Calandrino, A. J. Feldman, J. Appelbaum, and E. W. Felten. *Lest we remember: Cold boot attacks on encryption keys*. Proceedings of the 17th USENIX Security Symposium, July 28-August 1, 2008, San Jose, CA, USA. USENIX Association, pp. 45–60. 2008.

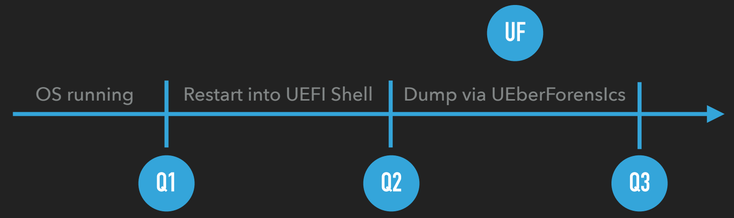
ARCHITECTURE

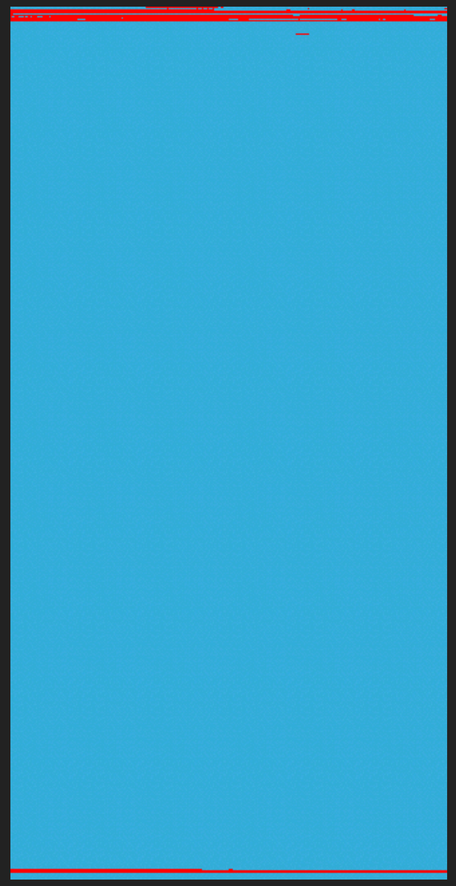


EVALUATION I

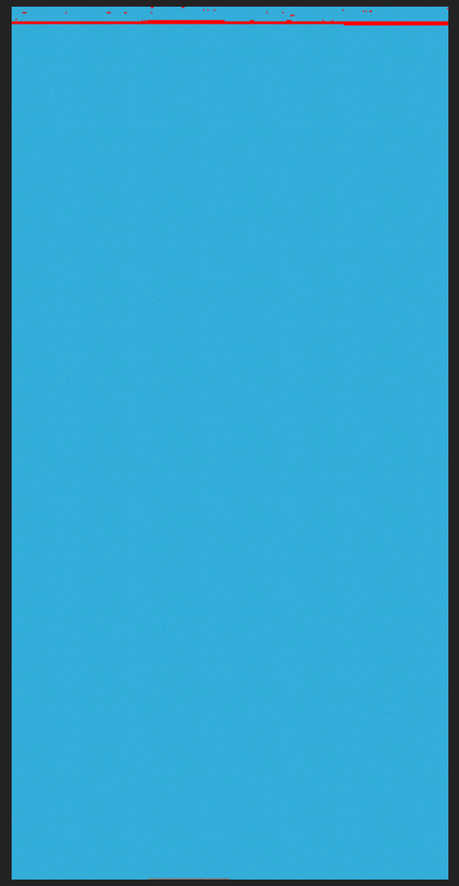


EVALUATION II

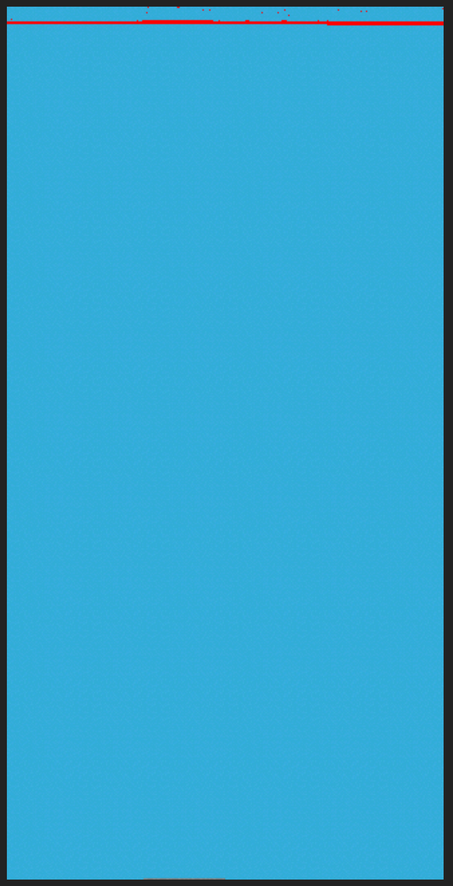




diff(Q1, Q2): 24.6 MiB (1.2%)



diff(Q2, UF): 4.9 MiB (0.2%)



diff(UF, Q3): 5.8 MiB (0.3%)

EVALUATION III

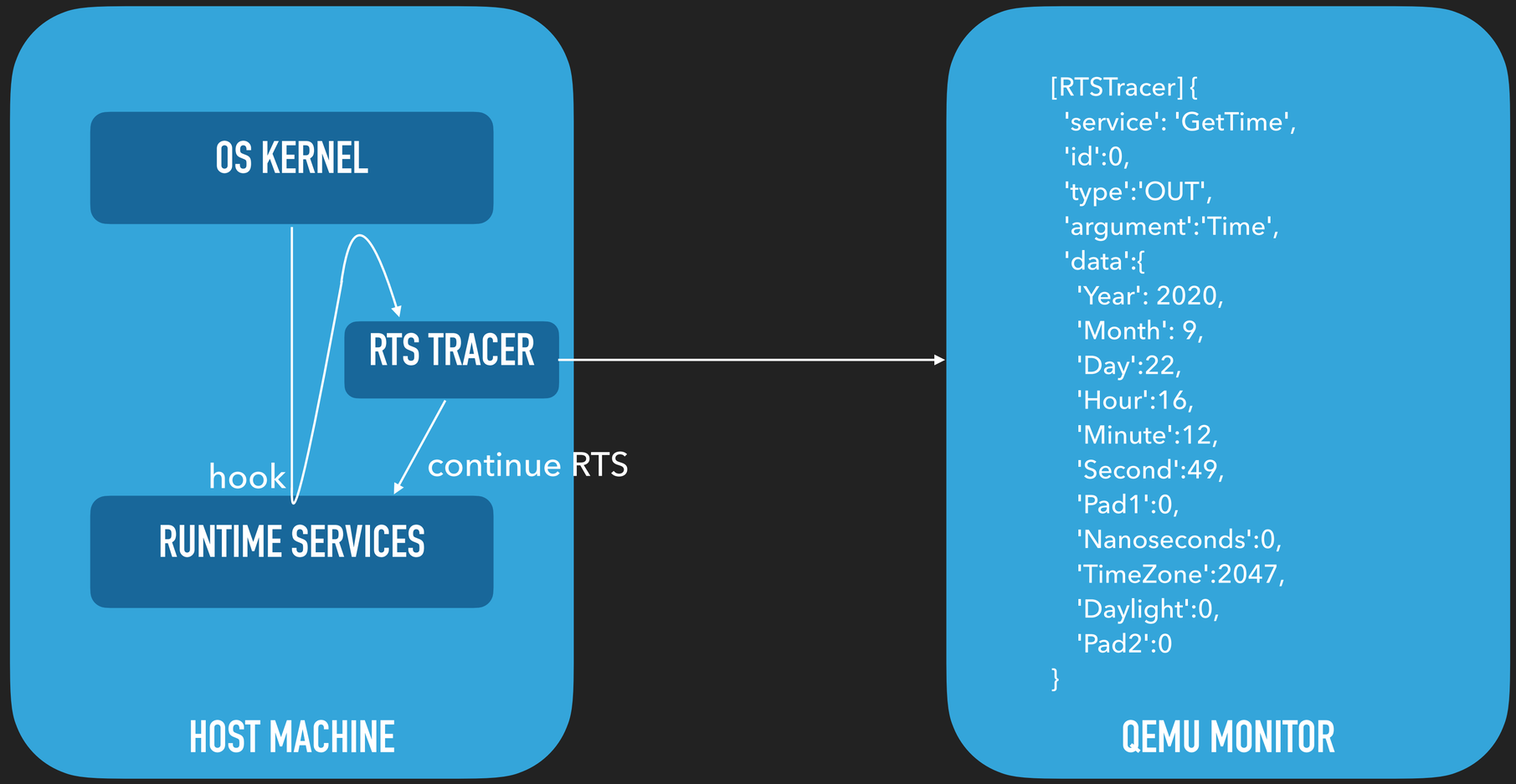
- Correctness
  - Relatively small differences probably caused by acquisition
- Atomicity
  - Similar as cold boot
- Integrity
  - Memory regions of the firmware are known

UEFI Runtime Services

IDEA

- UEFI Runtime Services (RTS)
  - can be used by the OS
  - *GetTime, SetTime, GetVariable, SetVariable, ResetSystem, UpdateCapsule*
- RTS can be called during OS runtime by the kernel
  - run with kernel privileges (in contrast to the SMM)
- Code is not part of the OS, OS-independent
- Hook existing RTS and acquire memory from there
  - Problem: Data exfiltration is difficult and forgeable
  - RTS Tracer shows that RTS hooking is possible

ARCHITECTURE



EVALUATION I

- Which RTS are called and how often?
- Trace RTS for different scenarios:
  - Boot: Start machine
  - Login: Boot + login
  - Working: Login + 1h of work
  - Hour: Login + 1h of idling
  - Switch: Login + user switch
  - Reboot: Login + Reboot + Login

EVALUATION II

Runtime Service	Boot	Login	Working	Hour	Switch	Reboot
GetTime	46	46	46	46	46	92
GetVariable	754	786	786	786	850	1617
SetVariable	110	110	110	110	110	165
GetNextVariable Name	499	499	499	499	499	1067
ConvertPointer	91	91	91	91	91	182

Conclusion and Future Work

CONCLUSION

- UEberForensIcs brings forensic memory acquisition to modern computer firmware
  - Good in terms of correctness, atomicity, and integrity
- UEFI RTS Tracer
  - Firmware code is still callable when the OS is running
  - No periodic calls, RTS call must be enforced by specific events

FUTURE WORK

- Run on bare metal
- Injection of actual memory acquisition software from the RTS
  - Special OS
  - Forensic Hypervisor [5,6]
  - Exfiltration?
- Protect from unauthorized access

[6] L. Martignoni, A. Fattori, R. Paleari, and L. Cavallaro. *Live and Trustworthy Forensic Analysis of Commodity Production Systems*. 13th International Symposium on Research on Recent Advances in Intrusion Detection (RAID 2010). 2010.
[5] R. Palutke, S. Ruderich, M. Wild, and F. Freiling. *HyperLeech: Stealthy System Virtualization with Minimal Target Impact through DMA-Based Hypervisor Injection*. 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020). 2020.

QUESTIONS?