Software-design for Dynamic Integrity Measurement

M. Jahnke · T. Rix · K.-O. Detken (DECOIT GmbH)

A. Rein (Huawei Technologies)

Prof. Dr. Kai-Oliver Detken
DECOIT GmbH
Fahrenheitstraße 9
D-28359 Bremen
https://www.decoit.de
detken@decoit.de

Agenda

- Introduction
  - Cooperative project
  - Technologies
  - Problem definition
  - Dynamic Integrity Measurement
- DRIVE Framework
  - Kernel module / measurement
  - Reference Value Generator (RVG)
  - Verification
- Guideline example
- Conclusion

Introduction
Project

- Cooperation project between DECOIT® GmbH and Huawei Technologies: *Dynamic Runtime Integrity Verification and Evaluation (DRIVE)*
- Main goal was to use a TPM chip for dynamic runtime integrity verification
- Both companies are members of the Trusted Computing Group (TCG)
- Integrity measurements were a part from different other research projects of DECOIT ® GmbH: VOGUE, ESUKOM, SIMU etc.
- DRIVE project wants to bring the research approach to industrial product lines
- First demonstrator has been presented on the TCG Members Meeting in July

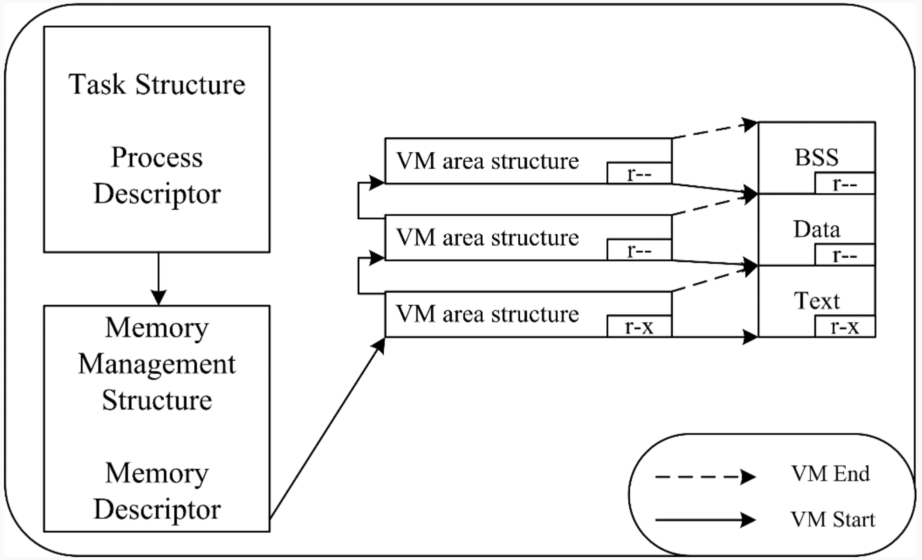
Introduction
Technologies – TPM

- The Trusted Platform Module (TPM) is a computer chip specified by the TCG
- The key material consists of
  - Endorsement Key (EK)
  - Storage Root Key (SRK)
- It implements several cryptographic functions
- Required key material is stored in a secure memory inside the TPM hardware
- The Platform Configuration Registers (PCRs) allow to store unchangeable sequences of data
- Main TPM functionalities are: Root of Trust, system integrity, integrity measurement, Trusted Boot, remote attestation

Introduction

Technologies – Memory Management

- Memory management in Linux
  - System memory (RAM) is divided into two parts at boot time:
    - Kernel space
    - User space memory
  - Process memory is divided into segments
    - Text (executable code)
    - Constant data
    - Variable data
  - Segments are further divided into pages of equal size (usually 4.096 bytes on modern architectures)
  - Pages have a set of flags indicating state and access rights



Introduction
Problem definition

- State of the art: static integrity measurement systems
  - e.g. Integrity Measurement Architecture (IMA)
  - Only verify integrity at boot (OS) or start time (processes)
- Attackers may attempt in-memory modifications on already running processes and OS components
  - Can change the behaviour of processes
  - Cannot be detected by IMA approach
- On servers, processes usually run long times before restart
  - Possible in-memory modifications are hard to spot and fix
  - Vulnerable check is only foreseen at boot time

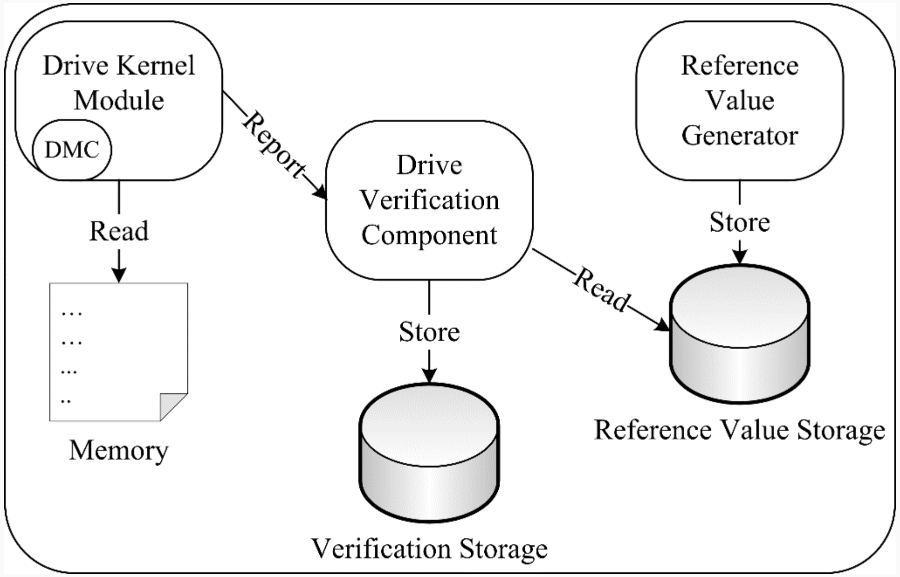
Introduction
Dynamic Integrity Measurement

- Solution: Dynamic Integrity Measurement (DIM)
- Integrity measurements at runtime
  - Operating System (Kernel + Loadable Kernel Module – LKM)
  - Processes (executables and dynamic libraries)
- Measurement of static memory segments
  - Executable code (machine code)
  - Constant data segments, including
    - Constants (e.g.: `const int i = 42`)
    - String literals (e.g.: `printf("string")`)
- Dynamic areas, such as heap and stack, are not monitored

DRIVE Framework

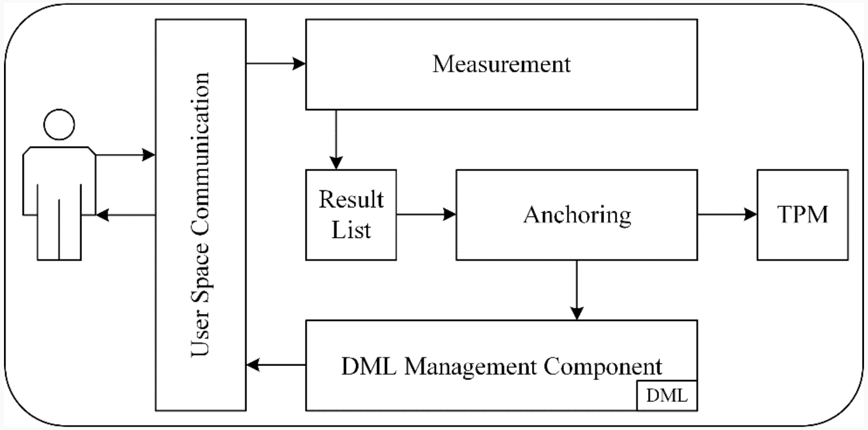
Overview

- DRIVE consists of three separate components
  - Measurement, including the DRIVE kernel module and System State Report (SSR) Generator
  - Reference Value Generator
  - Verification
- Verification Storage
  - Stores measurements and verification results
  - Can be used for further analysis and forensics



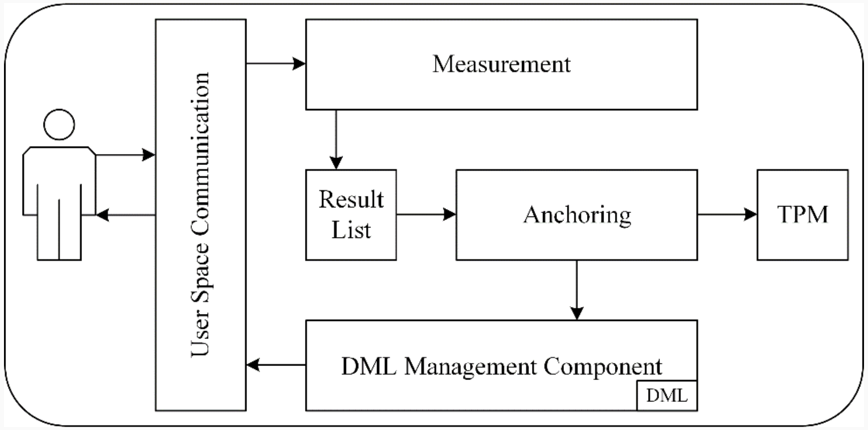
DRIVE Framework
DRIVE Kernel Module

- The DRIVE Kernel Module (DKM) is implemented as a Linux loadable kernel module (LKM)
- It provides a framework to access and measure user and kernel space memory
- It consists of five major modules
  - User space communication
  - Measurement
  - Anchoring
  - Trusted Platform Module (TPM)
  - Dynamic Measurement List (DML)



DRIVE Framework
Measurement

- DRIVE Kernel Module (DKM)
  - Identify required memory segments and calculate hash digests
  - Collect additional metadata (e.g. page access rights)
  - Anchor the results with the hardware trust anchor (TPM PCR)
  - Add the measurements to the DML
  - Entries can only be inserted into and read from the DML
- SSR Generator retrieves the DML contents from the DKM
- Transmission of the SSR to the verification system



DRIVE Framework
Reference Value Generator

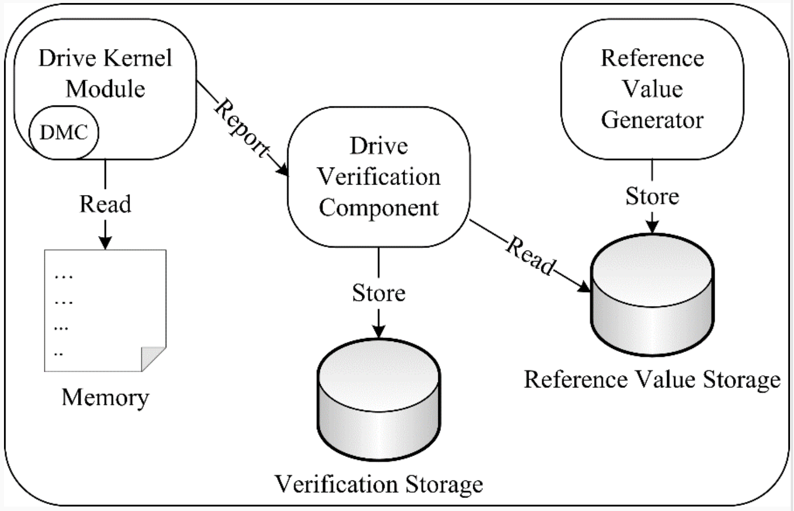
- The Reference Value Generator (RVG) produces the opposite of the measurement results, required by the verification
- Calculation of reference values
  - Requires static configuration data of the target system
  - Utilizes trustworthy binary source files (ELF files): these must match the ones present on the target system
  - Concrete results values or hash digests
  - If required, static metadata from the ELF file: some measurements require additional calculations performed by the verification
- Reference values are stored in the Reference Value Storage

DRIVE Framework
Verification

- Decoding and verification of SSR contents
- Comparison of measured and reference values
  - Direct comparison of values or hash digests
  - Additional calculations may be necessary
- On-demand calculation of hash digests
  - Requires runtime information from the target system
  - Requires static ELF information from the RVG
  - Apply these values to known binary memory images
  - Calculate hash digests of the modified image

Guidelines
Concept

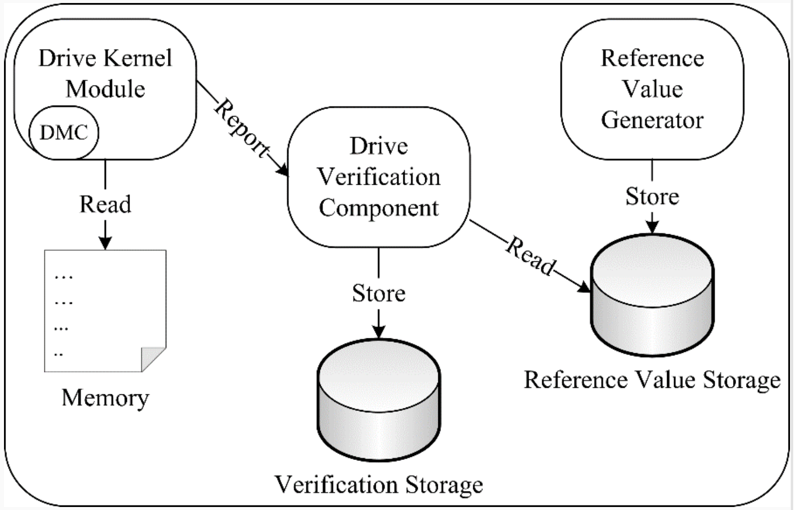
- Guidelines contain the actual business logic to perform measurements
- Modular measurement and verification logic
  - Encapsulation
  - Prevention of side-effects
- A set always contains three different guidelines
  - Measurement
  - Reference Value Generation
  - Verification
- Addition and removal of sets is easy
- Sets must not have dependencies on each other



DRIVE Kernel Module (DKM) Framework

Guidelines
Example

- Measurement (DKM):
  - Read memory segment and calculate hash digests
  - Read page access flags
  - Count pages with unexpected sets of flags (e.g.: expected: r-x → present: rwx)
- Reference Value Generation (RVG):
  - Calculate reference has digest from an ELF file
  - Static configuration of expected page access flags
- Verification
  - Compare hash digests from DKM and RVG
  - Compare page access flags with values from RVG
  - Test the page counter to be 0 (valid result)
  - All other results are invalid



DRIVE Kernel Module (DKM) Framework

Conclusions

- A modular framework for Dynamic Integrity Measurements on Linux kernels has been developed
- Integrity of the measurement results is guaranteed by utilizing TPM features
- Changes in static memory segments can be detected
- The guideline concept allows modular and independent code to perform the actual measurement and verification steps
- Guideline code is highly platform-dependent: it must be adjusted for other kernel versions or system architectures!
- The results were implemented as a demonstrator that was presented at the July TCG Members Meeting

Thank you for your attention!

DECOIT GmbH
Fahrenheitstraße 9
D-28359 Bremen

https://www.decoit.de
info@decoit.de