Trustworthy Measurements of a Linux Kernel and Layered Attestation via the seL4

Michael Neises
m811n155@ku.edu
neisesmichael@gmail.com

University of Kansas

September 17, 2023

Introduction & Motivation

1. Forked the "vm_introspect" camkes app
2. Wrote a suite of measurements over unmodified linux 6.1.y
3. Integrated those measurements into a system for remote attestation with KU-SLDG's am-cakeml
4. Three seL4 threads: VM, Inspector, Attestation Manager

---

1. One Principle of RA is "trustworthy mechanism"
2. Previous solutions were either vulnerable or cost hardware

System.map

1. I learned about system.map by studying the Volatility project.
2. It contains a table mapping symbols to virtual addresses
3. Symbol : Flag : Virtual Address

```
ffff800008edd000 T swapper_pg_dir
ffff800008b40000 D __start_rodata
ffff800008e75b40 R __start___ksymtab
```

4. (T)ext, (D)ata, (R)ead-only Data, etc

Types of Addresses

1. Kernel Virtual Address (Linear)
2. Kernel Virtual Address (Non-Linear)

```
bool is_linear_map_address(uint64_t vaddr){
  return ((vaddr ^ PAGE_OFFSET) < (PAGE_END - PAGE_OFFSET));
}
uint64_t kernel_virt_to_phys(uint64_t virtaddr)
{
    uint64_t ret;
    if(is_linear_map_address(virtaddr))
    {
        ret = (virtaddr & ~PAGE_OFFSET);
    }
    else
    {
        ret = (virtaddr - KIMAGE_VADDR);
    }
    return ret;
}
```

Measurement: Kernel Read-only Data

1. Determine the range of pages using System.map

```
ffff800008b40000 D __start_rodata
ffff800008e4ec10 D __start_ro_after_init
ffff800008e6f7b8 D __end_ro_after_init
ffff800008ed2000 D __end_rodata
```

2. We want those RO pages that are *not* RO after init.
3. Digest every such page somehow using SHA512

Measurement: Kernel Modules

1. A symbol in System.map refers to the list_head of the linked list of kernel modules. That list_head does not itself belong to a kernel module.

```
ffff800009466f90 D modules
```

```
struct list_head {
        struct list_head *next, *prev;
};
```

2. However, these addresses are not like those previous.
3. We can't translate them using the prior style of translation.

Translation Table Walk

1. It's not easy to find helpful prose on this subject.
2. This symbol points to the top-level "page global directory"

   ```
   ffff800008edd000 T swapper_pg_dir
   ```

3. ARM Architecture Reference Manual with *just the right* index
4. A virtual address is interpreted as a sequence of offsets
5. They are offsets into their PGD, PUD, PMD, and PTE.
6. The Page Table Entry contains our paydata.

Table D8-10 4KB granule translation table properties at each lookup level

Lookup level	Index into translation table	Maximum entries in table	Contents of translation table entries	Additional requirements
-1	- IA[51:48]	- 16	Lookup level not supported Table descriptors	Effective value of TCR_ELx.DS is 0 Effective value of TCR_ELx.DS is 1
0	IA[47:39]	512	Table descriptors Table descriptors and Block descriptors	Effective value of TCR_ELx.DS is 0 Effective value of TCR_ELx.DS is 1
1	IA[38:30]	512	Table descriptors and Block descriptors	-
2	IA[29:21]	512	Table descriptors and Block descriptors	-
3	IA[20:12]	512	Page descriptors	-

Measurement: Kernel Modules (final)

1. Now we can retrieve the kernel modules from their linked list.
2. Let's perform a measurement of their read-only data.
3. Every module has a module_layout that specifies the count and location of that module's RO pages.

```
struct module_layout {
    void *base; (the actual code + data)
    uint size; (total size)
    uint text_size; (size of exe code)
    uint ro_size; (size of (text+rodata))
    uint ro_after_init_size; (if exists, else ro_size)
};
```

Measurement: Tasks

1. The init_task symbol points to the swapper task.

   ```
   ffff8000093d3e40 D init_task
   ```

2. Swapper is the root of the process tree. Traverse.
3. Simple data like PID and Parent ID were easy to collect.
4. But the paydata is in VMAs, and those are not so easy
5. Before July 2022, VMAs were held in a linked list
6. Since Linux 6.1, VMAs are managed by Maple Trees

The Maple Tree

1. A balanced tree designed for storing non-overlapping ranges.
2. It replaces three data structures without performance penalty.
3. It's used only for memory management.
4. Its binary was particularly hard to read.

```
The Maple Tree squeezes various bits in at various
points which aren't necessarily obvious.  Usually,
this is done by observing that pointers are N-byte
aligned and thus the bottom log_2(N) bits are
available for use.
```

Measurement: Tasks (final)

1. Collect the task paydata via the leaves of the task's maple tree
2. The measurement reduces the process tree to name, PID, paydata, and parent/child relations

```
Task init recognized:
    const char init[] = "2B11AF6A0FF649922C6A837D487EE
    6601DE5FEA477976614A3CB9C5DDA3EDC6C023FE88D86E8940
    14FE64DF378FD336893B719D2A0B00369C28405B9B0557FD3";
    HexToByteString(&init,
     &digests[DIGEST_NUM_BYTES*(numDigests++)]);
```

End of Measurements

1. Each measurement has a corresponding appraisal function
2. Basically the digests are compared to known digests
3. Final result is passed to the AM running as an seL4 thread.
4. That includes a measurement of the AM running in the VM.

Future Work, Repo Links, Final Thoughts

1. Study kernel exploits
2. Try to fool the measurement suite
3. Port this work to a microcomputer (ODROID XU4)

1. https://github.com/ku-sldg/attarch/tree/main
2. https://github.com/ku-sldg/am-cakeml/tree/master