coreboot - Bug #66

rmodule_copy_payload() does not initialize unused memory

08/16/2016 06:38 PM - Trammell Hudson

Status:	New	**Start date:**	08/16/2016
Priority:	Normal	**Due date:**	
Assignee:	Aaron Durbin	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

If module->payload_size != rmodule_memory_size(module), the excess memory remains uninitialized in rmodule_copy_payload(). This prevents reproducible TPM measurements of the unpacked modules and could possibly lead to runtime bugs or security vulnerabilities.

History

#1 - 05/13/2017 09:11 PM - Nico Huber

- Assignee set to Aaron Durbin

#2 - 05/15/2017 10:02 PM - Aaron Durbin

What about rmodule_clear_bss() ? I'm confused here. Is this a theoretical issue or a do you have a rmodule where things aren't cleared?

payload_size is the on-disk usage while rmodule_memory_size is the full program (including bss).

#3 - 05/16/2017 09:44 PM - Aaron Durbin

This is from rmodule_load():

```
-------/*
------- * In order to load the module at a given address, the following steps
------- * take place:
------- *  1. Copy payload to base address.
------- *  2. Adjust relocations within the module to new base address.
------- *  3. Clear the bss segment last since the relocations live where
------- *     the bss is. If an rmodule is being loaded from its load
------- *     address the relocations need to be processed before the bss.
------- */
-------module->location = base;
-------rmodule_copy_payload(module);
-------if (rmodule_relocate(module))
------->-------return -1;
-------rmodule_clear_bss(module);
```

You want to verify the contents of the on-disk piece? I'm confused when/where you are trying to do measurements. It seems like you are doing it at the wrong place. There's not much information to go on here w.r.t. the original report.

#4 - 05/16/2017 09:54 PM - Trammell Hudson

I'm not sure about module_clear_bss() and will need to look into it. Right now I'm doing the measurement in cbfs_load_and_decompress().

The one module that shows the problem is the SMM code. However, I note that cbfs_prog_stage_load() appears to memset() the extra memory, so perhaps I need to relocate my measurement.

#5 - 05/16/2017 10:03 PM - Aaron Durbin

You are making your measurement at the wrong place for the rmodules. You need to hook into the full loading path for each type of thing loaded. prog_segment_loaded(..., SEG_FINAL) are the final contents including memory location and size. You could certainly hook into that in some form. That would give you a better view of the contents of memory.