



Host Integrity At Runtime & Startup (HIRS)

TCG Event Log Tool

October 2024

Users Guide

Version 3.0

Table of Contents

| | | |
|-------|--|----|
| 1 | Introduction | 1 |
| 1.1 | Purpose | 1 |
| 1.2 | Background | 1 |
| 1.2.1 | TPM Event Log | 1 |
| 1.2.2 | UEFI Boot Process | 2 |
| 2 | Installing the Event Log Tool | 3 |
| 2.1 | Installation | 3 |
| 2.2 | Event Log Location | 3 |
| 3 | Using the Event Log Tool – Linux | 3 |
| 3.1 | Parameters | 3 |
| 3.2 | Event Log Structure | 4 |
| 3.3 | Displaying Events | 4 |
| 3.3.1 | Displaying All Events | 4 |
| 3.3.2 | Displaying Only One Event | 5 |
| 3.4 | Outputting Event Log Information to a File | 5 |
| 3.5 | Displaying Information in Hex Format | 7 |
| 3.5.1 | Displaying An Event in Hex Format | 7 |
| 3.5.2 | Displaying An Event in Hex Format With Additional Context | 7 |
| 3.5.3 | Displaying Event Content in Hex Format With Additional Context | 7 |
| 3.6 | Displaying Expected PCR Values | 8 |
| 3.7 | Comparing Event Log Files | 9 |
| | Appendix A: Test Patterns | 10 |
| | Appendix B: References | 11 |

1 Introduction

1.1 Purpose

The purpose of this document is to define and support a command line application called the `tcg_eventlog_tool`. The `tcg_eventlog_tool` was created in order to inspect the Trusted Platform Module (TPM) Event Log's contents. This command tool supports the PC Client Reference Integrity Manifest (RIM) Specification^{1,2}, which specifies the use of the TPM Event Log as a Support RIM type. This tool can be used to parse and print human readable output, provide hexadecimal events which can be used as test patterns, and provide details in the case of events failing comparison.

1.2 Background

1.2.1 TPM Event Log

The TPM Event Log is defined in the TCG PC Client Platform Firmware Profile³ which is referred to as the "PFP". The Event Log file contains all the hashes that get extended into the TPM Platform Configuration Registers (PCR) during the boot cycle, as well as details about each hash and each hash's corresponding event. One can recreate the resultant PCR values by extending the values within this file, in which case the TPM PCR list may not be needed.

The project Host Integrity at Runtime and Startup (HIRS) contains a server-side application referred to as the Attestation Certificate Authority (ACA)⁴. The ACA uses the Event Log during its validation process if the firmware option is selected. The Event Log is one of the Support RIM file options for PC Client systems. This means that the Base RIM (SWID tag) file will have a hash of the Event Log in its payload for verification purposes. During the HIRS validation process, the Event Log is used to prove two things:

1. Real-time values check: The digest values found within the Event Log are used to calculate a composite hash, which the ACA compares against the expected composite hash value in the TPM Quote. Should TPM Quote verification pass, this proves that the ACA has received the correct Event Log and nothing has been altered in the Event Log since it was created (we already know the Quote is good based on the signature and nonce). Should TPM Quote verification fail, the Event Log file is needed to provide details on the individual hashes within each PCR.
2. Reference values check: Each event in the Event Log (supplied by the client) is compared against the events listed in the RIM (supplied by the OEM) to provide details in the case of events failing comparison.

¹ https://trustedcomputinggroup.org/wp-content/uploads/TCG_PCClientSpecPlat_TPM_2p0_1p04_pub.pdf

² https://trustedcomputinggroup.org/wp-content/uploads/TCG_PCClientSpecPlat_TPM_2p0_1p04_pub.pdf

³ https://trustedcomputinggroup.org/wp-content/uploads/TCG_PCClientSpecPlat_TPM_2p0_1p04_pub.pdf

⁴ https://trustedcomputinggroup.org/wp-content/uploads/TCG_PCClientSpecPlat_TPM_2p0_1p04_pub.pdf

1.2.2 UEFI Boot Process

During the boot process of a device, UEFI records hashes of firmware components and/or events to the Trusted Platform Module (TPM) PCRs⁵. Hashes normally capture firmware images, firmware configuration, expansion component firmware images, expansion component firmware configurations, and the bootloader. TPM-aware bootloaders can continue logging hashes to describe the kernel, initial file system, and any modules. Kernels, applications, and drivers can also log runtime hashes to the TPM.

Hashes are stored in the TPM's Platform Configuration Registers (PCRs) in accordance with Figure 1. Most TPMs have 24 PCRs per supported hash algorithm. TPM 1.2 supports SHA-1 (24 PCRs). TPM 2.0 supports SHA-1 and SHA-256 at the minimum (48 PCRs minimum). PCR values are computed via a series of one-way hashes where each measurement hash is appended to the current PCR value, and then the combination is hashed and becomes the new PCR value (referred to as "extending the PCR").

UEFI also records measurement hashes, along with information about these hashes, in the TPM Event Log. The TPM Event Log is an audit log that can be used for verification later, after the system boots.

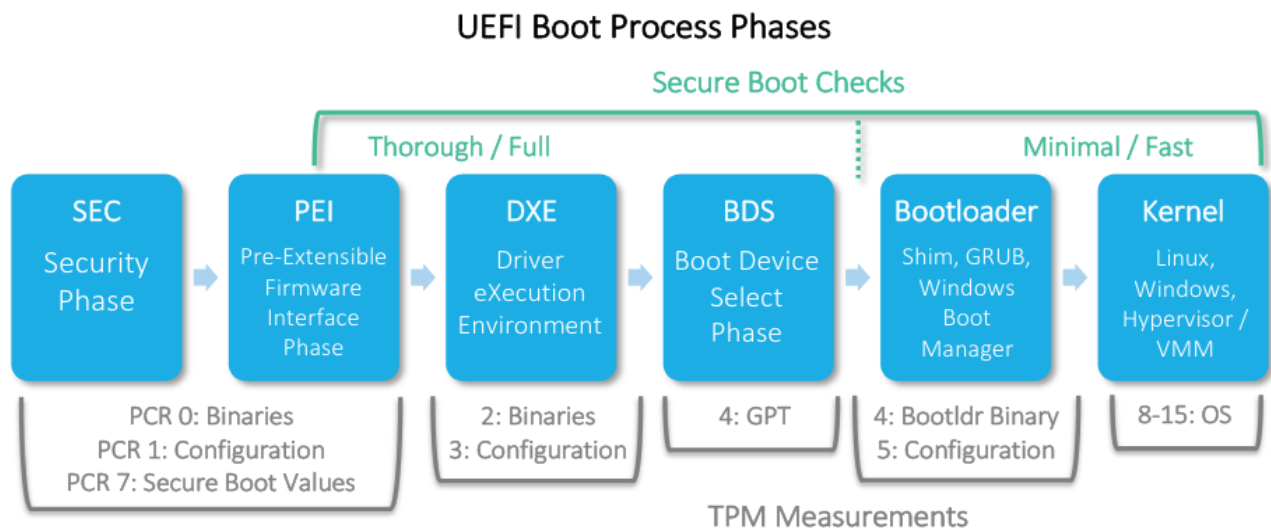


Fig. 1. The interaction of Secure Boot and TPM with UEFI boot phases.⁵

⁵ https://trustedcomputinggroup.org/wp-content/uploads/TCG_PCClientSpecPlat_TPM_2p0_1p04_pub.pdf

2 Installing the Event Log Tool

2.1 Installation

Installation instructions for the Event Log Tool are located in the tcg_eventlog_tool README:

- https://github.com/nsacyber/HIRS/blob/main/tools/tcg_eventlog_tool/README.md

2.2 Event Log Location

The default location for the TPM Event Log (Linux) is:

- `/sys/kernel/security/tpm0/` with a default name of "binary_bios_measurements"

The default location for the TPM Event Log (Windows) is:

- `C:\Windows\Logs\MeasuredBoot\`

3 Using the Event Log Tool – Linux

The tcg_eventlog_tool RPM will create a command line shortcut. This can be invoked from a command line by using:

```
elt -h
```

Invoking this command will bring up a Help page, which lists out the Event Log Tool's many uses and functions.

3.1 Parameters

`-f: --file`

Use a specific Event Log file. The following parameter **MUST** be a path and file name.

The local Event Log file will be used if this option is not present.

Note: Access to the local Event Log may require admin privileges.

`-e: --event`

Display event descriptions (including event content) in human readable form.

The following optional parameter is a single event number used to filter the output. All events will be displayed if the optional parameter is not provided.

`-ec: --contenthex`

Displays event content in eventhex format when `-event` is used.

`-ex: --eventhex`

Displays event in hex format when `-event` is used.

`-d: --diff`

Compares two TPM Event Logs and outputs a list of events of the second log that differed.

`-o: --output`

Output to a file. The following parameter **MUST** be a relative path and file name.

-p: --pcr

Output expected PCR value calculated from the TCG Log (for PCR Replay).

The following parameter MAY be a PCR number used to specify a single PCR.

No following parameters will display all PCRs.

-v: --version

Parser version.

-x: --hex

Displays an event in hex format. Use with -ec to get content.

Use -e -ec and -ex options to filter output.

All output will be human readable form if not present.

3.2 Event Log Structure

The format of the event log files is as follows:

pcrIndex: The PCR Register number, typically shown in documentation as PCR[0], where 0 is the pcrIndex.

eventType: An enumerated type found in Table 27 of the PFP. The PFP uses upper case labels to reference the events (e.g. event type 0x00000007 is labeled EV_S_CRTM_CONTENTS).

digests: This is a hash value (SHA1, SHA256 or SHA384 depending upon the log type). This may be a hash of firmware, a file, or the event itself. The coverage of the digest is dictated by Table 27 of the PFP.

eventSize: The size (in bytes) of the event data.

event: The event data as described by Table 27 of PFP.

Note: The Event# is not part of the TPM Event Log but is useful to display for identification purposes.

3.3 Displaying Events

3.3.1 Displaying All Events

In order to display all events within a specified log file, you will need an Event Log file. Typically the filetype would be a .bin. See Appendix A for links to example test patterns.

Once you have this file, you can input it into this command for results:

```
elt -f TpmLog.bin -e
```

```

tcg_eventlog_tool is opening file:TpmLog.bin

Event Log follows the "Crypto Agile" format and has 56 events:

Event# 0: Index PCR[0]
Event Type: 0x3 EV_NO_ACTION
digest (SHA-1): 0000000000000000000000000000000000000000000000000000000000000000
Event Content:
    Signature = Spec ID Event03 : Log format is Crypto Agile
    Platform Profile Specification version = 02.00 using errata version 00

Event# 1: Index PCR[0]
Event Type: 0x8 EV_S_CRTM_VERSION
digest (SHA256): 96a296d224f285c67bee93c30f8a309157f0daa35dc5b87e410b78630a09cfc7
Event Content:
0000

Event# 2: Index PCR[0]
Event Type: 0x80000008 EV_EFI_PLATFORM_FIRMWARE_BLOB
digest (SHA256): dbc7fc2d1845dfa3f87fe661a7a200a2798d1bdc55aaf0f4b5f2e9fbbca5466b
Event Content:
    Platform Firmware Blob Address = 6b207000 length = 851968

Event# 3: Index PCR[0]
Event Type: 0x80000008 EV_EFI_PLATFORM_FIRMWARE_BLOB
digest (SHA256): 60cce9bd7cc2196e9cd05853be8a564a0c8c4aec12411f8981aefaf04e82f20cf
Event Content:
    Platform Firmware Blob Address = 6afdc000 length = 2273280

```

The tool will list every event. In this example, there are 55 events in all.

3.3.2 Displaying Only One Event

If you would like to display only one event from an Event Log, you can use:

```
elt -f TpmLog.bin -e 1
```

For this example, Event #1 was used.

```

tcg_eventlog_tool is opening file:TpmLog.bin

Event Log follows the "Crypto Agile" format and has 56 events:

Event# 1: Index PCR[0]
Event Type: 0x8 EV_S_CRTM_VERSION
digest (SHA256): 96a296d224f285c67bee93c30f8a309157f0daa35dc5b87e410b78630a09cfc7
Event Content:
0000

```

3.4 Outputting Event Log Information to a File

If you would like to output information from the `tcg_eventlog_tool` to an external file for use later, you can use the `-o` option as below:

```
elt -f TpmLog.bin -p 0 -o example.txt
```

In this case, the query information about the TpmLog.bin file was saved to a new text file named example.txt.

Using `cat example.txt` shows that the information queried above from `elt -f TpmLog.bin -p 0` was saved to the example.txt file that was created:

```
tcg_eventlog_tool is opening file:TpmLog.bin
Expected Platform Configuration Register (PCR) values derived from the Event Log:

pcr 0 = 5ef6c69a589a96b5ade6a09e960eb341e6f68a8239df66be34e5e991ddde97a8

----- End PCR Values -----
```



```

tcg_eventlog_tool is opening file:TpmLog.bin

Event Log follows the "Crypto Agile" format and has 56 events:

Event# 1: Index PCR[0]
Event Type: 0x8 EV_S_CRTM_VERSION
digest (SHA256): 96a296d224f285c67bee93c30f8a309157f0daa35dc5b87e410b78630a09cfc7
Event Content:
0000
Event content (Hex) (2 bytes): 0000

```

3.6 Displaying Expected PCR Values

If you would like to view all expected PCR Values of an Event Log, you can use the -p option as below:

```
elt -f TpmLog.bin -p
```

```

tcg_eventlog_tool is opening file:TpmLog.bin
Expected Platform Configuration Register (PCR) values derived from the Event Log:

pcr 0 = 5ef6c69a589a96b5ade6a09e960eb341e6f68a8239df66be34e5e991dddde97a8
pcr 1 = 0f16d93fe0cbe7114fd9fefeb1d98a0802b184b6077f05275269aa90ebb8a993
pcr 2 = 966eb0b055e5b656f81c08ed1b2107cdea5740f321382d07a0eade7d014addee
pcr 3 = 3d458cfe55cc03ealf443f1562beec8df51c75e14a9fcf9a7234a13f198e7969
pcr 4 = c919e77702cb066016b575c008659ba7d758b0b4c3f9df29658e1770699823d1
pcr 5 = 45f6dd68feb493ec2f371f2fbd2f904181a20e9491102304f239745f6fdleaf6
pcr 6 = 3d458cfe55cc03ealf443f1562beec8df51c75e14a9fcf9a7234a13f198e7969
pcr 7 = 65caf8dd1e0ea7a6347b635d2b379c93b9a1351edc2afc3ecda700e534eb3068
pcr 8 = 0000000000000000000000000000000000000000000000000000000000000000
pcr 9 = 0000000000000000000000000000000000000000000000000000000000000000
pcr 10 = 0000000000000000000000000000000000000000000000000000000000000000
pcr 11 = 0000000000000000000000000000000000000000000000000000000000000000
pcr 12 = 0000000000000000000000000000000000000000000000000000000000000000
pcr 13 = 0000000000000000000000000000000000000000000000000000000000000000
pcr 14 = 0000000000000000000000000000000000000000000000000000000000000000
pcr 15 = 0000000000000000000000000000000000000000000000000000000000000000
pcr 16 = 0000000000000000000000000000000000000000000000000000000000000000
pcr 17 = ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
pcr 18 = ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
pcr 19 = ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
pcr 20 = ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
pcr 21 = ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
pcr 22 = ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
pcr 23 = 0000000000000000000000000000000000000000000000000000000000000000

----- End PCR Values -----

```

3.7 Comparing Event Log Files

If you would like to compare Event Log files to see where certain events may have failed comparison, you can use this command:

```
elt -d TpmLog.bin TPMLog_Altered.bin -p
```

The two files being compared in this example are TpmLog.bin and TPMLog_Altered.bin.

```
tcg_eventlog_tool is opening file:TpmLog.bin
tcg_eventlog_tool is opening file:TPMLog_Altered.bin

Event Log TpmLog.bin did NOT match EventLog TPMLog_Altered.bin
There were 2 event mismatches:

Event# 6: Index PCR[0]
Event Type: 0x1 EV_POST_CODE
digest (SHA256): b35c7afc52ea5f813320b3f269ce2cae3899f718ddc1fa5bc2b19b8f2ec16088
Event Content:
BAD DATA

Event# 25: Index PCR[1]
Event Type: 0x80000002 EV_EFI_VARIABLE_BOOT
digest (SHA256): 6c362820e63da000a221476fcc2d509041ed6ad16e9f040e7869albba452446a
Event Content:
UEFI Variable Name:BootOrder
UEFI_GUID = 8be4df61-93ca-11d2-aa0d-00e098032b8c : EFI_Global_Variable
UEFI Variable Contents =>
  BootOrder = Boot 0004Boot 0003Boot 0002Boot 0000Boot 0001Boot 0005
```

As you can see above, the Event Logs had 2 event mismatches. Since a mismatch has occurred, this could mean that the digest values within the Event Log are not verifiable and may have been tampered with.

Appendix A: Test Patterns

TPM Event Log (*TpmLog.bin*):

https://github.com/nsacyber/HIRS/blob/main/tools/tcg_rim_tool/src/test/resources/TpmLog.bin

TPM Event Log (*TPMLog_Altered.bin*):

https://github.com/nsacyber/HIRS/blob/main/tools/tcg_rim_tool/src/test/resources/TPMLog_Altered.bin

Appendix B: References

- [1] Trusted Computing Group. (2024, April 26). *TCG PC Client Reference Integrity Manifest Specification, Vs 1.1 Rev 1.0*. Retrieved from Trusted Computing Group: <https://trustedcomputinggroup.org/resource/tcg-pc-client-reference-integrity-manifest-specification/>
- [2] Trusted Computing Group. (2024, April 26). *TCG Reference Integrity Manifest (RIM) Information Model, Vs 1.1 Rev 1.0*. Retrieved from Trusted Computing Group: <https://trustedcomputinggroup.org/resource/tcg-reference-integrity-manifest-rim-information-model/>
- [3] Trusted Computing Group. (2023, Dec 4). *TCG PC Client Platform Firmware Profile Specification, Vs 1.06 Rev 52*. Retrieved from Trusted Computing Group: <https://trustedcomputinggroup.org/resource/pc-client-specific-platform-firmware-profile-specification/>
- [4] NSACyber HIRS Group. (2024, Aug 1). *ACA User Guide*. Retrieved from Github HIRS: <https://github.com/nsacyber/HIRS/tree/main?tab=readme-ov-file#quick-links>
- [5] Cybersecurity Requirements Center. (2019, June). *Boot Security Modes and Recommendations*. Retrieved from National Security Agency: <https://media.defense.gov/2019/Jul/16/2002158058/-1/-1/0/CSI-BOOT-SECURITY-MODES-AND-RECOMMENDATIONS.PDF>