Host Integrity At Runtime & Startup (HIRS)

TCG Event Log Tool

10 June 2021

Users Guide

Version 2.1

Table of Contents

Table of Contents

1 Introduction

1.1 Purpose

The purpose of this document is to support and define a command line application called the tcg_eventlog_tool. The tcg_eventlog_tool was created in order to support the PC Client RIM Specification which utilizes the TPM Event Log as a Support RIM type and for inspection of the TPM Event Log's contents. This command tool can be used to parse and print human readable output, provide hexadecimal events which can be used as test patterns, and to compare event logs for providing details on what events may have miscompared.

1.2 Background

1.2.1 TPM Event Log

The TPM Event log is defined in the TCG PC Client Platform Firmware Profile[1] which is referred to as the "PFP". The Event log contains all the hashes that get extended into the TPM PCRs values during the boot cycle, so one can recreate the resultant PCRs by extending the values within this file, therefore TPM PCR List may not be needed. This file will be needed to show what each PCR covers and to provide details should TPM Quote verification fail.

For provisioning, the TCG 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 it in its payload for verification purposes.

1. The digest values found within the logs can be used to calculate the expected values in the TPM Quote.
2. The events in the RIM can be used to compare against the log provided by the client to detail which event caused the mis-compare.

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

1.2.2 UEFI Boot Process

UEFI can record hashes of firmware components to the Trusted Platform Module (TPM) in the TPM Event Log. The TPM must be both activated and enabled for hashes to be written. 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 log runtime hashes to the TPM too.

Hashes are stored in the TPM's Platform Configuration Registers (PCR) 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, then the combination is hashed and becomes the new PCR value. Measurement hashes are recorded in an audit log for verification later.

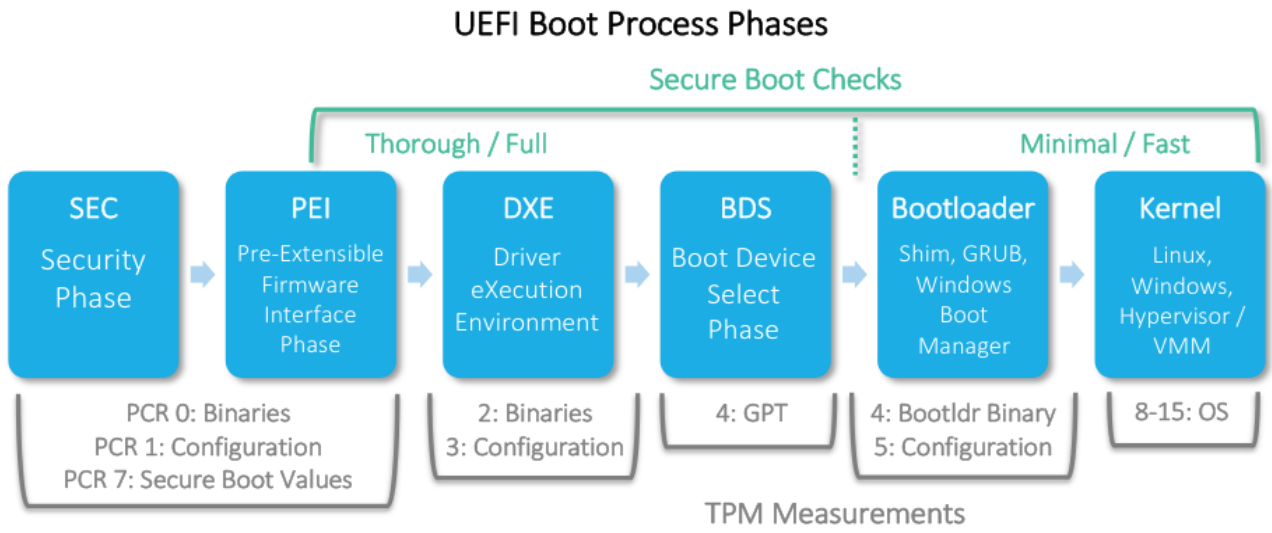

Fig. 1. *The interaction of Secure Boot and TPM with UEFI boot phases.*[2]

[2] https://media.defense.gov/2019/Jul/16/2002158058/-1/-1/0/CSI-BOOT-SECURITY-MODES-AND-RECOMMENDATIONS.PDF

2 Installing the Event Log Tool

To install the Event Log Tool, access the releases page on the HIRS GitHub repository. Make sure to download the RPM files which apply to the latest release. Currently installation packages are only available for Centos 7.

2.1 Centos 7 Installation

To install this tool, use the following command from the directory where the Event Log Tool is placed:

```
sudo yum localinstall tcg_eventlog_tool-X.X.X-1.i386.rpm
```

Where X.X.X is the latest version of the tcg_eventlog_tool package.

Note: Once installed, the tcg_eventlog_tool can be run from any directory in Linux.

2.2 Using the Event Log Tool with Windows

Currently, there is no installation package for the tcg_eventlog_tool for Windows. However, it can be invoked using Java:

To run the tcg_eventlog_tool through a Command Shell:

Navigate to the tcg_eventlog_tool folder and invoke using java –jar option to the tcg_eventlog_tool jar file:

```
java -jar build\libs\tools\tcg_eventlog_tool-1.0.jar -h
```

Another way you could invoke:

```
java -jar build\libs\tools\tcg_eventlog_tool-1.0.jar -f
C:\Windows\Logs\MeasuredBoot\0000000059-0000000000.log -e
```

2.3 Requirements

Note that a TCG Event Log will only be populated on a given device if the device:

1. Utilizes TCG compliant UEFI Firmware.
2. Has a TPM 1.2 or 2.0 that has been activated prior to the current boot.
3. Has a TCG aware OS.

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

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

The default location for the TCG 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 TCG 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 would be the pcrIndex.

eventType: An enumerated type found in Table 9 of the PFP. The PFP uses upper case labels to reference the events (e.g. event type 0x00000007 is labelled EV_S_CRTM_CONTENTS).

digests: This is a hash value (SHA1 or SHA256 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 9 of the PFP.

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

event: The event data as described by Table 9 of PFP.

Note: The Event# is not part of the TCG 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. This would be a .bin.

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): 0000000000000000000000000000000000000000
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): 60cce9bd7cc2196e9cd05853be8a564a0c8c4aec12411f8981aefa04e82f20cf
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 like so:

```
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 -----------------
```

3.5 Displaying Information in Hex Format

3.5.1 Displaying Only An Event in Hex Format

If you would like to display an event from the tcg_eventlog_tool in a Hex Format, you can use the –x option like this:

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

In this example, Event #1 is transcribed into Hex Format:

```
0000000008000000000000000000000000000000000000000000000000000000000000000000000000
0020000000000000000000000000000000000000000000000000000000000000000000000000000000
```

3.5.2 Displaying Only An Event in Hex Format With Additional Context

If you would like to display an event in hex format with additional context but no content information, you can use the –ex option like this:

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

In this example, Event #1 is transcribed into Hex Format:

```
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 (Hex no Content) (78 bytes): 0000000008000000000000000000000000000000000000000
0000000000000000000000000000000200000000000000000000000000000000000000000000000000
000000000000000000000
```

3.5.3 Displaying Event Content in Hex Format With Additional Context

If you would like to display an event with content information in hex format with additional context, you can use the –ec option like this:

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

In this example, Event #1 and it's content has been transcribed into Hex Format:

```
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 like so:

`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 = 5ef6c69a589a96b5ade6a09e960eb341e6f68a8239df66be34e5e991ddde97a8
 pcr 1 = 0f16d93fe0cbe7114fd9fefeb1d98a0802b184b6077f05275269aa90ebb8a993
 pcr 2 = 966eb0b055e5b656f81c08ed1b2107cdea5740f321382d07a0eade7d014addee
 pcr 3 = 3d458cfe55cc03ea1f443f1562beec8df51c75e14a9fcf9a7234a13f198e7969
 pcr 4 = c919e77702cb066016b575c008659ba7d758b0b4c3f9df29658e1770699823d1
 pcr 5 = 45f6dd68feb493ec2f371f2fbd2f904181a20e9491102304f239745f6fd1eaf6
 pcr 6 = 3d458cfe55cc03ea1f443f1562beec8df51c75e14a9fcf9a7234a13f198e7969
 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 = ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
 pcr 18 = ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
 pcr 19 = ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
 pcr 20 = ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
 pcr 21 = ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
 pcr 22 = ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
 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 miscompared, 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): 6c362820e63da000a221476fcc2d509041ed6ad16e9f040e7869a1bba452446a
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.

- See Appendix A for more information and test patterns.

Appendix A: Test Patterns

Event Log Files

TCG TPM Event Log *(TpmLog.bin)* :
https://github.com/nsacyber/HIRS/blob/master/tools/tcg_rim_tool/src/test/resources/TpmLog.bin

TCG TPM Event Log (*TPMLog_Altered.bin*) :

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

Appendix B: Building the Event Log Tool

B.1 Windows

1. Building the Event Log Tool

Several options exist for building on Windows 10:

1. Windows Command Shell (CMD.exe):
    - Navigate to the tcg_eventlog_tool folder and run the Windows Gradle wrapper:

➢ `gradlew.bat clean build`

2. Windows PowerShell with Windows Subsystem for Linux enabled:
    - Navigate to the tcg_eventlog_tool folder and run the Linux Gradle wrapper:

➢ `./gradlew clean build`

In both cases the tcg_eventlog_tool-X.X.jar file should have been placed in the build\libs\tools\ (Windows) or build/libs/tools/ (Linux) folder.

B.2 Linux

1. Building the Event Log Tool

To build this tool, navigate to the tcg_eventlog_tool directory and use the following command:

```
./gradlew clean build
```

Appendix C: Packaging the Event Log Tool

C.1 Windows

1. **Packaging the Event Log Tool**

Currently, only an installation file for Linux RPM is supported.

C.2 Linux

1. **Packaging the Event Log Tool**

To create an RPM on a Linux device, use the following command in the tcg_eventlog_tool directory:

```
./gradlew buildRPM
```

Appendix D: References

[1] Trusted Computing Group. (2020, November 12). *TCG Reference Integrity Manifest (RIM) Information Model.* Retrieved from Trusted Computing Group: https://trustedcomputinggroup.org/wp-content/uploads/TCG_RIM_Model_v1p01_r0p16_pub.pdf

[2] Trusted Computing Group. (2020, March 31). *TCG PC Client Reference Integrity Manifest Specification.* Retrieved from Trusted Computing Group: https://trustedcomputinggroup.org/wp-content/uploads/TCG_PC_Client_RIM_r0p15_15june2020.pdf

[3] Trusted Computing Group. (2019, December 4). *TCG PC Client Platform Firmware Integrity Measurement.* Retrieved from Trusted Computing Group: https://trustedcomputinggroup.org/wp-content/uploads/TCG_PC_Client-FIM_v1r24_3feb20.pdf

[4] Trusted Computing Group. (2019, June 3). *TCG PC Client Platform Firmware Profile Specification.* Retrieved from Trusted Computing Group: https://trustedcomputinggroup.org/wp-content/uploads/TCG_PCClientSpecPlat_TPM_2p0_1p04_pub.pdf

[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