Windows

Secured Boot and Measured Boot: Hardening Early Boot Components against Malware

September 7, 2012

Abstract

This paper provides information about the early boot protection features for Windows operating systems. It provides guidelines for anti-malware and other security solution developers to develop security solutions to protect early boot components from malware. It assumes that the reader is familiar with developing kernel mode drivers and has some familiarity with the Trusted Platform Module.

This information applies to the following operating systems:

Windows 8
Windows Server 2012

References and resources discussed here are listed at the end of this paper.

The current version of this paper is maintained on the Web at:

Secured Boot and Measured Boot: Hardening Early Boot Components against Malware

Microsoft

Contents

Introduction

This document describes two features of the Windows operating system:

- Measured Boot
- Secured Boot with Early Launch Antimalware (ELAM)

The goal of these features is to improve the capabilities of Antimalware (AM) solutions to provide security on the Windows operating system (OS), prove client trustworthiness to remote machines, and ultimately increase user satisfaction and trust of the Windows platform and AM software.

The Measured Boot feature provides AM software with a trusted (resistant to spoofing and tampering) log of all boot components that started before AM software. AM software can use the log to determine whether components that ran before it are trustworthy versus infected with malware. The AM software on the local machine sends the log to a remote sever for evaluation. The remote server initiates remediation actions either by interacting with software on the client or through out-of-band mechanisms as appropriate. The remote server is necessary because the software running on the local client is not trustworthy if malicious components that executed before the AM software tamper with the AM software. This feature has a dependency on a Trusted Platform Module (TPM) 1.2 being present on the system.

The ELAM feature provides a Microsoft-supported mechanism for AM software to start before all other third-party components. AM drivers are initialized first and allowed to control the initialization of boot drivers, potentially not initializing unknown boot drivers. Once the boot process has initialized boot drivers and access to persistent storage is available in an efficient way, existing AM software continues to block malware from executing.

The two features can be used individually or can be combined to increase the overall effectiveness of AM software. Microsoft recommends use of both features when the hardware supports a TPM.

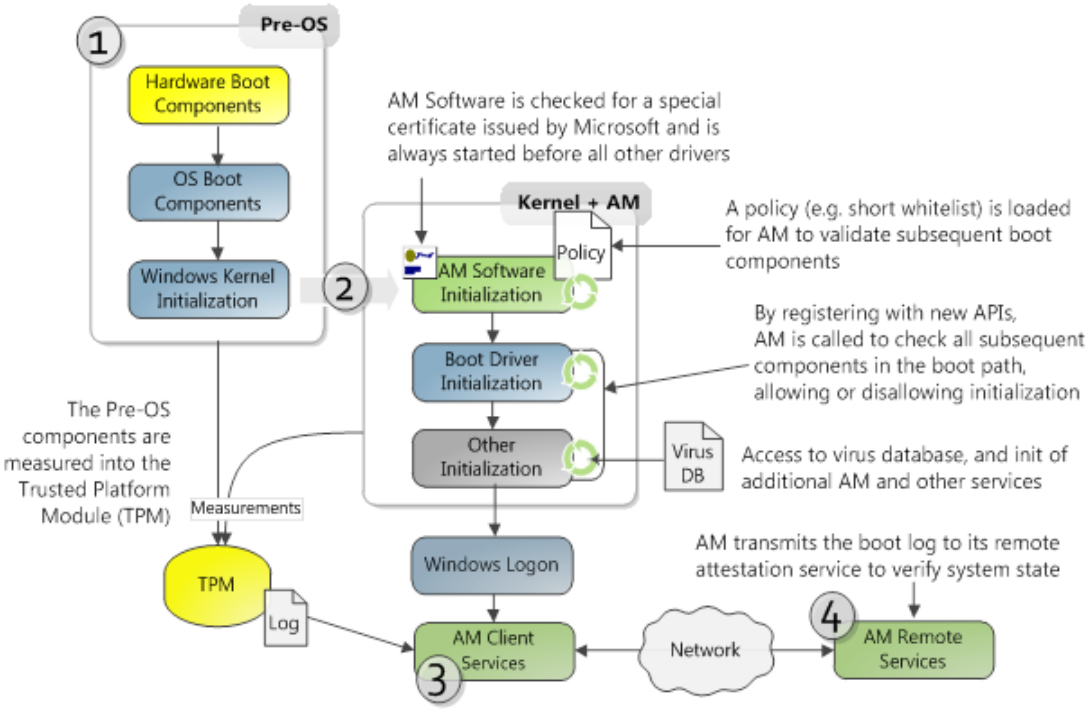

Figure 1: Measured Boot and Early AM Driver Start

Figure 1 shows both features together. Label 1 is where the boot process begins when the system is turned on. The hardware components load OS components that load and initialize the Windows kernel. Measurements are recorded in the TPM device for the components executed during the boot process. As shown in label 2, the Windows kernel then initializes ELAM software with a small policy data set, then boot drivers, and then other drivers. Before this phase, measurements are recorded in the TPM to fully reflect the OPERATING SYSTEM state when AM software will start. The ELAM software is able to enforce policy by stopping the initialization of unknown boot drivers. Later, when the file system is available, a full AM policy database can be loaded and it can be used to enforce policy by stopping the initialization of later unknown components. After Windows log on, at label 3, the AM software gathers a log containing measurements of the boot components recorded in the TPM and sends the information to a remote server for validation of the client system state. This figure provides one simple illustration of the two features. Variations of the features can provide customer value in unique ways.

Early Launch AM Software

How It Works

ELAM ensures that AM software is running and actively monitoring the system before any third-party drivers are initialized. Before each boot driver is initialized, the Windows kernel initiates a callback to the AM driver to verify the trustworthiness of the driver being initialized. The AM driver then returns Good, Bad, or Unknown. Based on administrator-defined policy, the kernel uses the returned state to determine whether to initialize the driver or to skip initialization.

Identifying an AM Driver as an Early Launch Driver

When an ELAM driver is installed, it registers itself as a boot driver and as an early launch driver using existing tools for online and offline installation through typical INF processing.

The ELAM driver must advertise itself as a boot start driver similar to all other boot start drivers. The INF sets the start type to SERVICE_BOOT_START (0), which indicates the driver should be loaded by the boot loader and initialized during kernel initialization. An ELAM driver advertises its group as *Early-Launch*.

Loading Boot Drivers

All boot drivers are loaded into memory by the Windows OS loader, winload, prior to handoff to the Windows kernel. The boot driver loading process is shown in Figure 2: Loading Boot Drivers**Error! Reference source not found.** as it relates to the overall boot sequence.

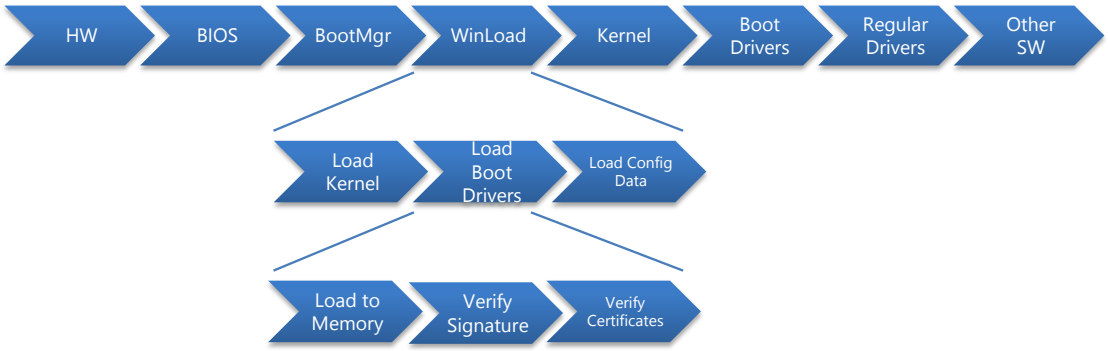

Figure 2: Loading Boot Drivers

Winload checks the signatures of all boot drivers as they are loaded into memory. On 64-bit platforms all boot drivers must have valid signatures, their certificates must chain to a trusted root certificate authority, and their certificates must contain the Codesigning enhanced key usage (EKU) extension. Signing is not required on 32-bit platforms. ELAM drivers are special in that they must be signed by Microsoft, and their signing certificates must also contain a special EKU indicating that it is an ELAM Driver. This is true for both 32-bit and 64-bit platforms.

Verifying Malware Signature Data

Each ISV is responsible for updating and verifying its malware signature data for its ELAM driver. It will be stored in a well-known location in the registry and will be loaded by winload at the same time as the AM driver. When the AM driver is initialized, it should verify the integrity of the malware signature data by validating a digital signature or other vendor-specific means.

Initializing the Drivers

Once the boot drivers are loaded into memory, control is handed over to the Windows kernel. The kernel first initializes the ELAM drivers, which then register for callbacks from the kernel. Through these callbacks, the AM drivers get information about each of the other boot drivers as they are about to be initialized. Using this

information, the AM drivers classify each of the other drivers as good, bad, or unknown and return that classification to the kernel. Figure 3: Boot Driver Initialization shows the driver initialization steps as they relate to the overall boot process.

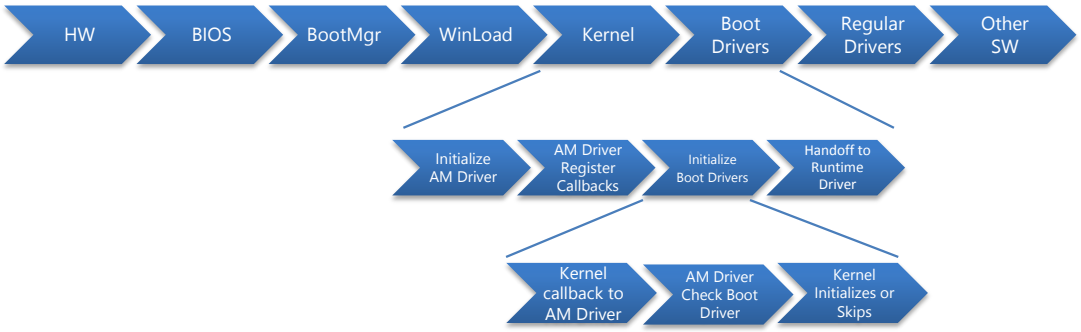

Figure 3: Boot Driver Initialization

The kernel uses the classification to determine whether to initialize the driver or to skip initialization. The boot driver initialization policy is administrator-definable. The default is to initialize good and unknown drivers, and to skip bad drivers. Once all boot drivers have been evaluated, the kernel will unload the Early Launch AM drivers.

Handoff to Runtime Driver

ELAM drivers are intended to be lightweight, performant drivers that only evaluate boot drivers. It is expected that an ISV would continue to have a runtime AM driver with a full signature database to continually monitor the system once it has started. To ensure continuation of coverage, the ELAM driver should hand off any status information to the runtime AM driver before it is unloaded. This requires that the runtime AM driver be a boot driver so it is loaded, evaluated, and started before the ELAM driver unloads.

Interface with Measured Boot

Measured Boot is not required for ELAM drivers, but in cases where measured boot is available and running, it can enhance the effectiveness of the AM driver. The AM driver itself, as well as its configuration (malware signature) data, is measured and the measurements are stored in the TPM. In addition, if the ELAM driver detects that the computer is not healthy according to the ELAM policy, it can affect the Measured Boot log so that attestation fails.

Measured Boot

How It Works

Measured Boot is intended to prove which components ran before AM software was started. By providing reliable information about the initial state of the system when AM software starts and by AM software enforcing security policy, a client machine can prove to a remote machine it is in a trustworthy state.

A high-level diagram of the Windows boot process is shown in Figure 4. This diagram omits ELAM drivers because it is not a requirement for Measured Boot.

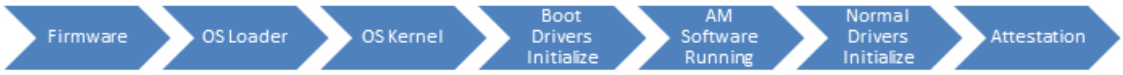

Figure 4: High-Level Windows Boot Flow

To prove which components ran before the AM software started, measurements are recorded in a TPM. A TPM is a hardware component usually soldered to the motherboard. The TPM provides a variety of security functions including storage of boot integrity measurements and mechanisms to prove the current integrity measurements. For Measured Boot, all components that run before AM software are recorded in the TPM. Successively, for each component (loaded by hardware or software) its measurement is recorded in the TPM before it is run. By design, once a measurement is recorded in the TPM, the measurement cannot be reset until the next reset of the system. This prevents a malicious component from tampering with its own measurement. Generally, successive boots of the platform result in identical measurements.

On a PC Client platform, the TPM has at least 24 different registers for storing integrity measurements called Platform Configuration Registers (PCR). Different measurements events are stored in different PCRs. For example, the system firmware generally records all the measurements in the PCRs numbered 0 through 7, whereas the PCRs numbered 8 through 15 hold measurements made by an operating system.

When enabled, the Measured Boot feature will record integrity measurements which include the OS kernel components and all boot drivers, including third-party drivers. If AM software starts as a boot driver, the OS state when the AM software starts will be accurately reflected in measurements recorded in the TPM's PCR measurements. Optionally, the AM software can use the Windows TPM driver to record additional measurements in the TPM. This allows AM software to continue boot measurements farther if desired.

AM software can use the TPM to provide evidence about its current PCR measurement values. The client can send the measurements to a remote machine. This process is called attestation. The remote machine can evaluate the PCR measurement values and determine if the values represent a trustworthy OS state including the launch of trustworthy AM software.

Measured Boot with Conventional BIOS

Figure 5 shows the potential boot process from system power on for a conventional BIOS system. The arrows illustrate how the typical boot process passes control from component to component. The initial component in the top left is the Core Root of Trust for Measurement (C-RTM) and it is the firmware component which starts the measurement chain by recording itself and the Power On Self-Test (POST) component before passing control to it. The TPM PCRs used for each measurement are listed in brackets. Information about the firmware components and the TPM measurements made by hardware are documented in the Trusted Computing Group (TCG) PC Client Specification Implementation for Conventional BIOS. Once the OPERATING SYSTEM

kernel starts, it initializes ELAM drivers, the TPM driver and its dependent drivers, and then other boot drivers. The AM software runtime that enforces policy for Measured Boot must start as a boot-critical driver because if it starts later, the measurements the OPERATING SYSTEM placed in the TPM will not fully represent the OPERATING SYSTEM state when the AM runtime software was started. Eventually normal drivers are initialized and the OPERATING SYSTEM continues the initialization. Optionally, the AM runtime could extend additional measurements into the TPM PCR 15 using the Microsoft TPM Driver. The legend shows what types of components are measured. The figure does not show attestation which would occur later. The figure does include the ELAM Driver for completeness, but use of it is not mandatory as the Measure Boot feature primarily depends on the AM Runtime component initialized in Phase 3 to enforce AM policy.

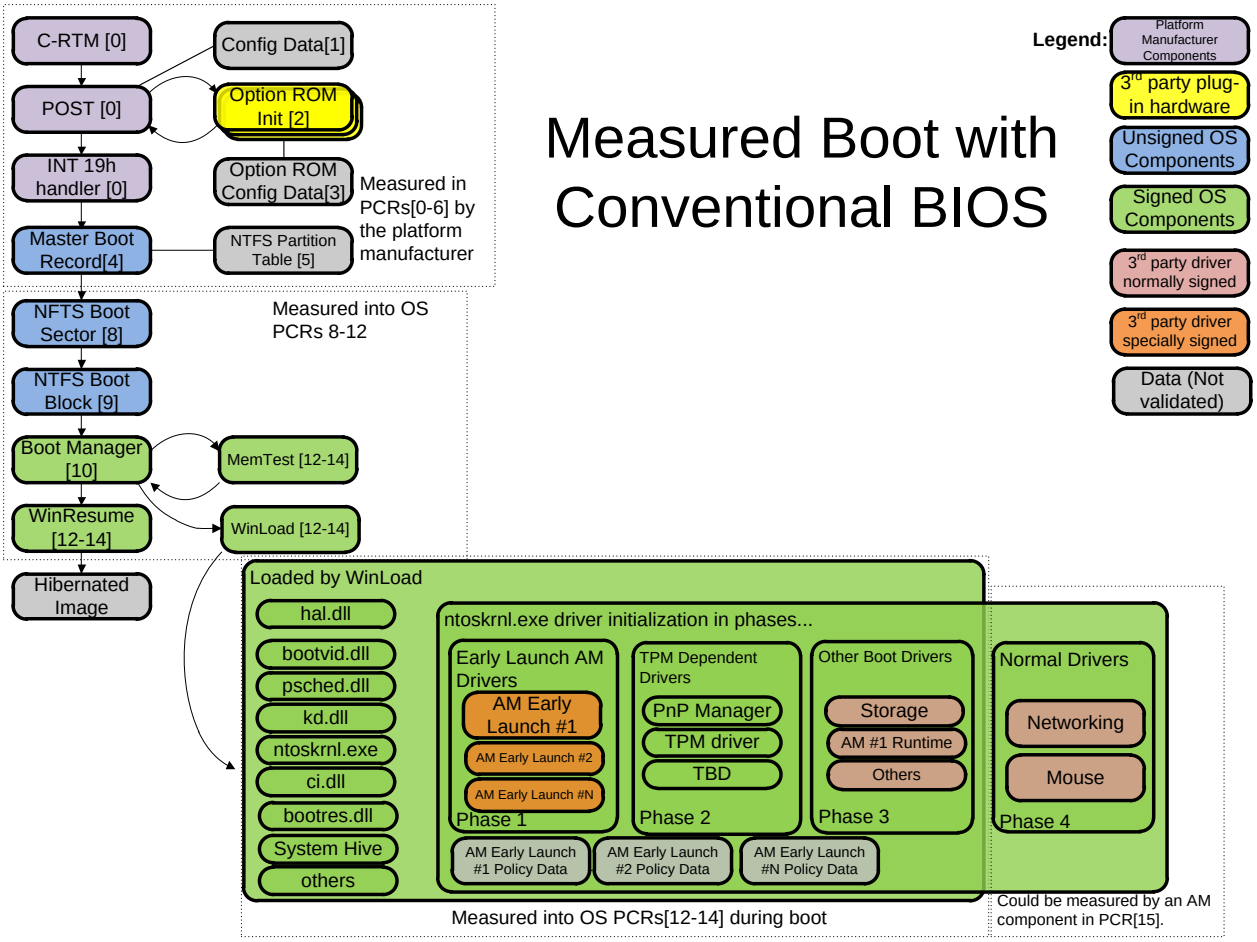

Figure 5: Measured Boot with Conventional BIOS

Measured Boot with Unified Extensible Firmware Interface (UEFI)

Figure 6 shows the potential boot process for UEFI platforms. The operating system portion of the flow is similar to conventional BIOS platforms, but the first OS component is Boot Manager. Information about the firmware components and the

TPM measurements made by hardware is documented in the TCG [UEFI Platform Specification.]

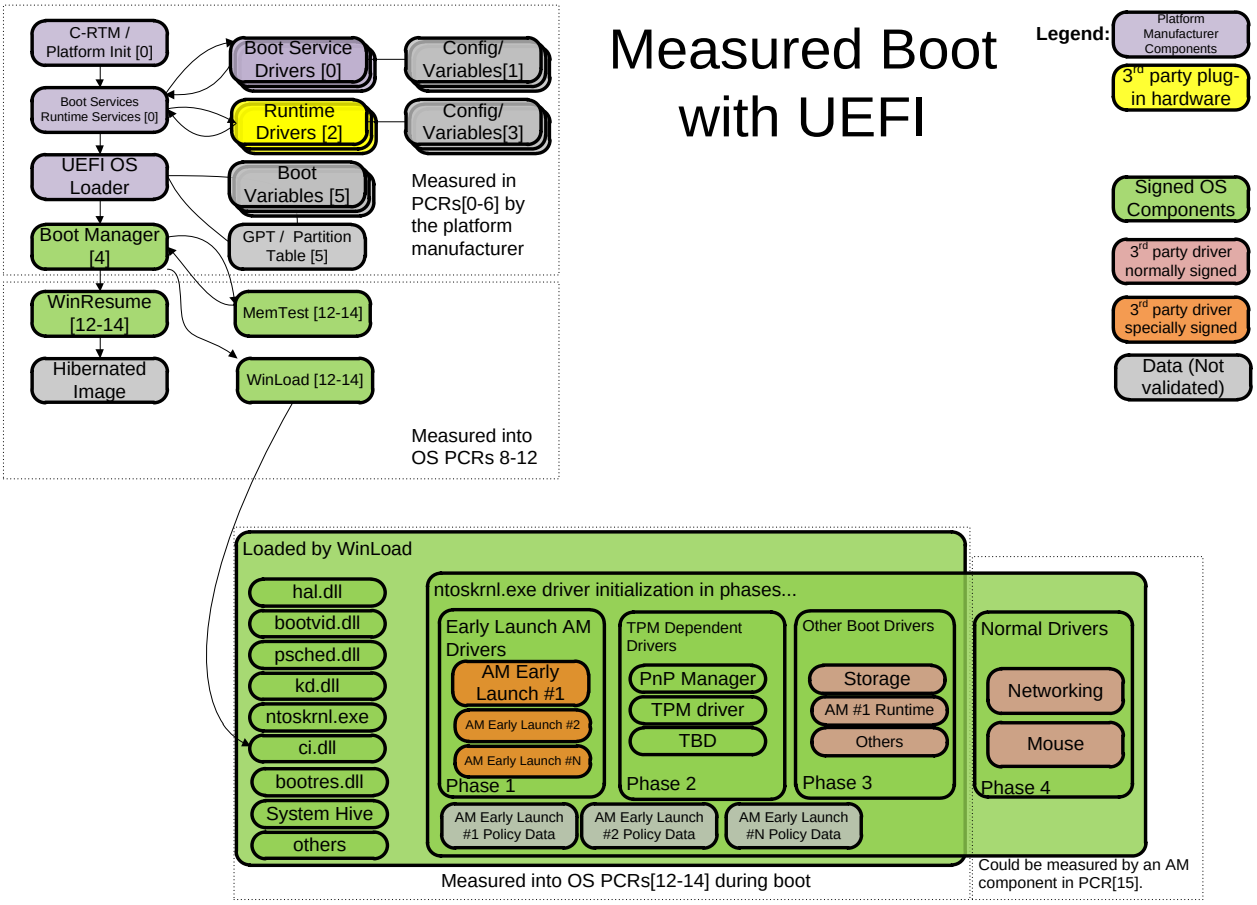

Figure 6: Measure Boot with UEFI

AM Software Adding Measurements

AM software or third-party applications can add more measurements using PCR value 15. Because multiple third-party tools can extend measurements in PCR 15, a special structure is used for the measurements so applications can check whether a measurement was made by their component or a different component.

Attestation

Attestation is a client machine providing evidence about its state to a remote computer. The Measured Boot feature allows a client to send TPM PCR values to a remote computer in a trusted way. Client software communicates with a remote machine to get a nonce value. The client software uses Measured Boot interfaces to get a Quote, which is a signed statement of the client's TPM PCR values that includes the nonce. The nonce helps prevent Quote replay attacks. The key used to sign the Quote is called an Attestation Identity Key (AIK). The AIK is unique per client and the server should verify that the AIK used to sign the Quote is associated with the client to prevent attacks when a Quote from a machine other than the client is provided to the remote server.

The remote server understands the PCR values inside a Quote it receives from a client if it has seen the same PCR values previously. If the PCR values are not familiar to the remote verifier, it might need additional information about what component measurements resulted in the PCR values. This information is contained in a log. The log lists different components that were measured into each PCR and the measurement value associated with each. The remote server can confirm the measurements listed in the log match the TPM PCR values in the Quote. If the measurements listed in the log do not match the TPM PCR values, the client is untrustworthy. If the measurements listed in the log do match the TPM PCR values, the remote server should evaluate each measurement in the log and determine if it is the measurement of a trustworthy component.

If the AM software enforces policy that maintains the state of the client indefinitely, attestation might only be necessary once for the client per a boot. If the AM software enforces policy that extends additional measurements in the TPM over time which represent changes to the state of the client, attestation might need to occur more frequently.

Provisioning

Measured boot depends on the following provisioning actions:

1. Making the TPM ready for use.
2. Establishing an AIK the server associates with the client.
3. Turning on the boot measurements.
4. Installing an AM software component as a boot driver which enforces policy.
5. Installing client components that perform attestation with a remote server.

Resources

PC Client Specification Implementation for Conventional BIOS

http://www.trustedcomputinggroup.org/resources/pc_client_work_group_specific_implementation_specification_for_conventional_bios_specification_version_12

TCG UEFI Platform Specification

http://www.trustedcomputinggroup.org/resources/tcg_efi_platform_specification_version_120_revision_10