Linux en UEFI met Secure Boot

Hans Lunsing

De vorige maal heb ik u kennis laten maken met UEFI, de Unified Extensible Firmware Interface, als opvolger van het aloude BIOS. Daarbij ging ik in het bijzonder in op enkele zaken waarin de UEFI zichtbaar laat blijken dat hij er is, namelijk de EFI-systeempartitie op de harde schijf, de GUID-partitietabel (GPT) als opvolger van de oude MSDOS-partitietabel, en Secure Boot dat alleen gewaarmerkte besturingssystemen en drivers laat opstarten.Nu komt aan de orde hoe het Linux-besturingssysteem omgaat met deze nieuwe UEFI-systemen, en met Secure Boot in het bijzonder en welke gereedschappen Linux ervoor biedt.

UEFI en GPT

Linux en Grub

Grub, de standaard Linux-bootloader, ondersteunt EFI sinds 2008, terwijl de Linux-kernel EFI zelfs al ondersteunt sinds 2000, samen met de EFI-Linux-bootloader (Elilo). Elilo wordt sinds vorig jaar niet meer verder ontwikkeld omdat Grub zijn taken heeft overgenomen. De Linux-kernel kan sinds versie 3.3.0 zelfs worden opgezet met de mogelijkheid zelf als EFI-bootloader te fungeren. Hij kan dan direct van de UEFI-shell of het opstartmenu worden gestart. In een niet-UEFI-omgeving, zoals het oude BIOS, functioneert zo'n kernel als vanouds. Grub en Linux kunnen ook zonder UEFI van een GUID-partitietabel booten.

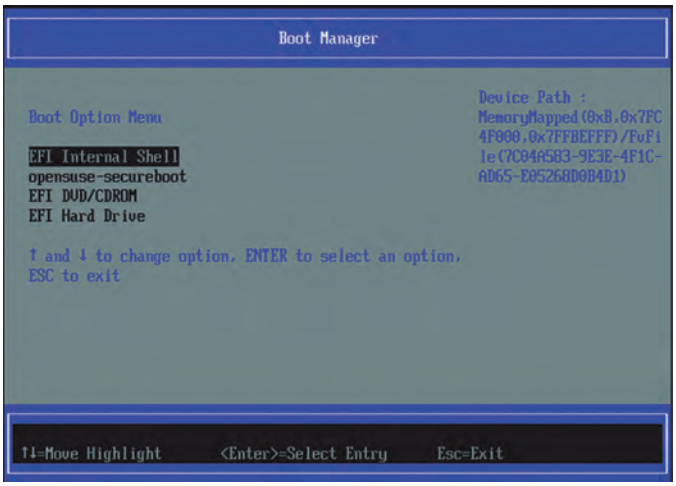
Omdat een GPT geen ruimte biedt voor de tweedefase-opstartcode van Grub moet daarvoor een speciale partitie, de BIOS Boot Partition, worden aangemaakt. Deze heeft geen bestandssysteem en kan vrij klein zijn. Het minimum is 31 KiB, maar het is aan te raden hem met het oog op toekomstige uitbreidingen groter te maken, zeg 1 MiB. Het type van de partitie dient '0xEF02' ofwel 'bios_grub' te zijn. De GUID van de partitie behoort `21686148-6449-6e6f-744e656564454649` te zijn. Wanneer deze GUID in het vereiste formaat (little endian) naar de GPT wordt geschreven vormt hij de ASCII-string 'Hah!IdontneedEFI', ofwel 'Hah! Ik heb EFI niet nodig'. Opneming van de tweedefasecode in een aparte partitie zonder bestandssysteem heeft het voordeel dat hij niet zomaar door andere software kan worden overschreven en niet door een bestandssysteem van plaats kan worden veranderd.

In een UEFI-Linux-systeem wordt de EFI System Partition (ESP) gekoppeld aan de map `/boot/efi`. Linux slaat z'n runtime informatie over EFI op in de map `/sys/firmware/efi`. Een kernelmodule (driver) met de naam 'efivarfs'[1] maakt het mogelijk dat de EFI-variabelen in een eigen bestandssysteem met de naam 'efivarfs' in de map `/sys/firmware/efi/efivars` kunnen worden geraadpleegd. Als Grub bij zijn installatie de EFI-variabelen in deze map vindt, voert hij een UEFI-installatie uit. Anders wordt het een gewone BIOS-installatie. De EFI-variabelen kunnen we ook raadplegen in de map `/sys/firmware/efi/vars`, waar ze zijn opgenomen in het standaard door de kernel verzorgde bestandssysteem 'sysfs'.

Bootmanagers

Om van verschillende opslagmedia en besturingssystemen te kunnen opstarten, komt een bootmanager van pas. Een UEFI-installatie beschikt over een eigen bootmanager die bij het opstarten kan worden opgeroepen door het indrukken van een toets (verschillend per implementatie).

Afbeelding 1 toont het bootmanagermenu in VirtualBox (zie volgende paragraaf) met openSUSE.



Afbeelding 1: Boot Manager van VirtualBox met openSUSE

Er zijn ook alternatieve bootmanagers voor UEFI beschikbaar Een handige UEFI-bootmanager is **rEFInd** (www.rodsbooks.com/refind/). Hij detecteert welke EFI-bootloaders u hebt geïnstalleerd en presenteert op basis daarvan een mooi grafisch menu met opstartkeuzes (afb. 2).

Afb. 2: Een rEFInd-menu met verschillende besturingssystemen

Een andere UEFI-bootmanager is het simpele maar effectieve Gummiboot (freedesktop.org/wiki/Software/gummiboot/). Het toont de verschillende opstartkeuzes (afb. 3)

Afbeelding 3: Een gummiboot menu

Programma's

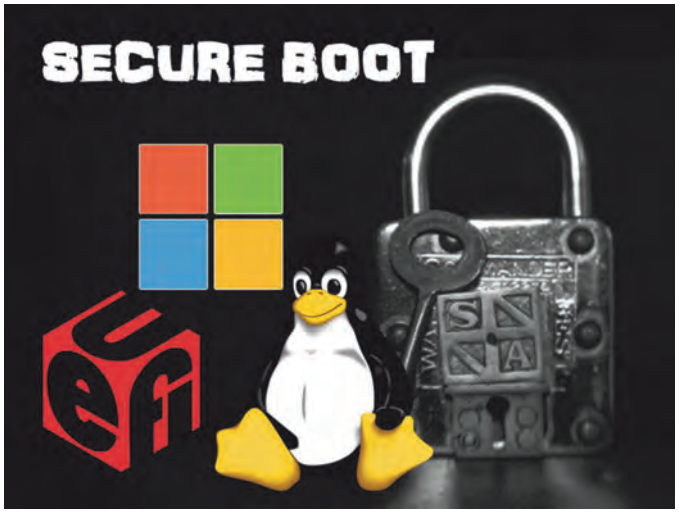
VirtualBox, een programma dat een virtueel systeem biedt waarop besturingssystemen kunnen worden geïnstalleerd, gebruikt standaard virtuele BIOS-firmware, maar heeft ook virtuele EFI-firmware ter beschikking. Deze is experimenteel, heeft geen Secure Boot, en werkt alleen voor OS X en Linux-gastsystemen, niet voor Windows[2]. Zo kun je je gemakkelijk enigszins vertrouwd maken met UEFI, de EFI-shell, en de EFI-systeempartitie, ook al heb je zelf geen UEFI-computer.

Wanneer men bij het opstarten van zo'n virtueel UEFI-systeem een willekeurige toets een aantal keren indrukt, verschijnt een menu met onder meer de keuzes 'Boot Manager' en 'Boot Maintenance Manager'[3]. Met die laatste kunnen instellingen van de EFI-bootmanager worden bekeken en gewijzigd. Zo kunnen onder meer menukeuzes aan het opstartmenu worden toegevoegd danwel eruit worden verwijderd, kan de opstartvolgorde worden veranderd en kan worden vastgelegd welk systeem de volgende keer wordt opgestart. Vanuit de 'Boot Manager' kan een van de aanwezige opstartbare opslagmedia, een van de opgenomen besturingssystemen of de EFI-shell worden gestart.

Bij een UEFI-installatie van Linux behoort het programma *efibootmgr*. Het doet ongeveer hetzelfde als de EFI-Boot Maintenance Manager, maar nu vanuit Linux. Ook voor dit programma moet de kernelmodule 'efivarfs' geladen zijn. Hoe dit programma moet worden gebruikt, krijgt u te zien door de opdracht 'man efibootmgr' te geven.

Naast 'efibootmgr' bieden bepaalde distributies nog programma's om UEFI-variabelen te bekijken en te bewerken, en om digitale handtekeningen te manipuleren, zoals de 'efitools' van de Linux Foundation. Deze komen verderop nog ter sprake.

Linux-programma's voor het bewerken van partities, zoals *parted* en zijn grafische evenknie *gparted*, kunnen goed met GPT overweg. De command-line programma's *fdisk, cfdisk* en *sfdisk* ondersteunen GPT in sommige distributies wel en in andere weer niet. In plaats van *fdisk c.s.* kunnen *gdisk, cgdisk* en *sgdisk* worden gebruikt. Deze laatste kunt u beslist niet voor schijven met MBR/MSDOS-partities gebruiken, tenzij u het oude partitieschema naar het GPT-partitieschema wilt converteren. Let wel dat de data op die partities dan verloren kunnen gaan!



Secure Boot

Secure Boot is een UEFI-protocol dat verhindert dat besturingssystemen, bootloaders of drivers worden geladen die niet met een geldige digitale handtekening zijn ondertekend. Zo krijgt malware geen kans in het opstartproces van de computer.

Sleutels

Hoe zat het ook alweer? Er is een platformsleutel (Platform Key, ofwel PK), die door de platformeigenaar, in principe de eigenaar van de hardware, wordt beheerd. De PK wordt normaliter door de fabrikant van de hardware al naar de firmware geschreven (ook al pleit de Linux Foundation ervoor dat aan de platformeigenaar over te laten). Dat kan een PK zijn van de hardwarefabrikant zelf, maar het kan ook een PK zijn van de leverancier van het besturingssysteem, meestal Microsoft[4]. Daarnaast is er een verzameling Key-Exchange Keys (KEK's), die door de fabrikant (OEM) en de leveranciers van besturingssystemen worden beheerd. Deze KEK's bestaan uit een tweetal sleutels: een openbare en een private. Alleen de sleutelbeheerder kent de private sleutel. Iedereen kan echter een KEK installeren in de UEFI 'signature database', omdat daarvoor alleen de openbare sleutel nodig is, samen met de PK. Dit onderscheid tussen openbare en private sleutel is van groot belang, omdat het de platformeigenaar in staat stelt te besluiten welke sleutel hij vertrouwt en kan installeren, terwijl de KEK-beheerder er zeker van kan zijn dat het besturingssysteem of de driver veilig kan opstarten.

Microsoft

Secure Boot maakt sinds versie 2.2 van november 2010 deel uit van de UEFI-specificatie. In 2011 kondigde Microsoft aan dat computers alleen voor Windows 8 konden worden gecertificeerd als ze een UEFI-BIOS hadden waarin Secure Boot met digitale handtekening van Microsoft was geactiveerd. Dat hield in dat andere besturingssystemen, waaronder Linux (maar ook Windows 7), niet zo maar op zo'n computer kunnen worden geïnstalleerd. Snel kwam dan ook de beschuldiging dat Microsoft het op deze manier onmogelijk maakte om een alternatief besturingssysteem naast of in plaats van Windows te installeren.

De soep wordt echter nooit zo heet gegeten als zij wordt opgediend. Want wat is het geval? Secure Boot kan (meestal) worden uitgeschakeld. Bovendien kan de UEFI in BIOS-compatibiteitsmodus worden gezet, in welk geval Secure Boot ook niet werkt. Niettemin zagen Linux-ontwikkelaars wel degelijk de voordelen in van Secure Boot.

De Linux Foundation

De Linux Foundation (www.linuxfoundation.org), die de verdere ontwikkeling van de Linux-kernel ondersteunt en de werkgever is van Linus Torvalds, de geestelijke vader van Linux, is deelnemer geworden van het UEFI-Forum (www.uefi.org) om zo invloed uit te oefenen op de verdere ontwikkeling van UEFI en met name Secure Boot. De stichting heeft een document gepubliceerd met de titel 'Making UEFI Secure Boot Work With Open Platforms' (URL[5]) waarin uiteen wordt gezet hoe Secure Boot zou moeten worden geïmplementeerd om goed met open systemen zoals Linux samen te kunnen werken.

Daarnaast heeft de Linux Foundation een UEFI Secure Boot Systeem voor open-source software (URL2[6]) ontwikkeld. Dit bestaat uit een kleine pre-bootloader, ondertekend met een Microsoft-sleutel, die Linux of een ander besturingssysteem of een bootloader zoals Grub kan opstarten. Daarbij kan worden gedacht aan een CD/DVD-installer, een LiveCD-distributie of een geïnstalleerd besturingssysteem van welke distributie dan ook, die de pre-bootloader wil gebruiken. In geval het op te starten systeem zelf niet netjes ondertekend is, vraagt de pre-bootloader de gebruiker ter wille van de veiligheid om toestemming. De pre-bootloader behoort onder de naam 'Loader' tot een pakket van door de Linux Foundation ontwikkelde EFI-tools met de naam 'efitools'. Met programma's uit dit pakket kan de UEFI-'signature database' worden uitgelezen en gemanipuleerd. Enkele Linux-distributies, waaronder Ubuntu, hebben het pakket in hun software-repositories. Voor Fedora en openSUSE wordt het geleverd in een van de OBS (openSUSE Build System)-repositories.

Linux-distributies

Ook de Linux-fabrikanten Canonical (Ubuntu), Red Hat (Red Hat en Fedora) en SUSE (SUSE en openSUSE) nemen deel aan het UEFI-Forum.

Ubuntu (en afgeleiden zoals Linux Mint) ondersteunt UEFI sinds versie 11.10 en Secure Boot vanaf versie 12.04.2. Het doet dat met een zgn. *shim*-loader (een *shim* is een tussenstuk, plug of wig) met de naam 'shimx64.efi'. Deze is ge-

waarmerkt met een Microsoft-sleutel en verifiëert bij het starten van de bootloader Grub ('grubx64.efi') of deze met Canonical's eigen sleutel is ondertekend. Hij laat toe dat ook niet gewaarmerkte kernels worden geladen. Immers, een gewaarmerkte kernel beveiligt alleen het draaiende systeem en niet de pre-boot-toestand waarvoor Secure Boot nu juist bescherming biedt. Zo kunnen gebruikers gewoon op maat gemaakte kernelmodules, zoals de niet-vrije videodrivers van AMD en NVIDIA, gebruiken of zelfs hun eigen kernels bouwen zonder het systeem te moeten herconfigureren.

openSUSE ondersteunt UEFI sinds versie 12.2 (september 2012) en Secure Boot vanaf versie 12.3 (maart 2013). Ook openSUSE gebruikt een shim-loader 'shim.efi' met een ondertekening van Microsoft, die de bootloader Grub genaamd 'grub.efi' start en daarbij verifiëert of Grub ondertekend is met een openSUSE-certificaat. Grub op zijn beurt start dan de Linux-kernel en verifiëert dat ook deze met een openSUSE-certificaat is ondertekend. Dat verschilt van de manier waarop Ubuntu het doet, en het betekent dat openSUSE's Grub niet zonder meer niet- of andersondertekende kernels kan starten. Om toch andersondertekende bootloaders of kernels te kunnen starten, biedt de shim-loader de mogelijkheid andere certificaten te importeren. Wanneer de shim-loader wordt gevraagd een programma te laden dat geen bekende ondertekening heeft, wordt het EFI-programma 'MokManager' aangeroepen, waarmee het mogelijk is certificaten in de signature database te importeren. Linux-kernelmodules worden, in tegenstelling tot de kernel zelf, zonder verificatie geladen.

In openSUSE 12.3 was ondersteuning van Secure Boot nog experimenteel. YaST was niet in staat om vast te stellen of Secure Boot was ingeschakeld, reden waarom het bij installatie mogelijk was handmatig ondersteuning voor Secure Boot in te schakelen. Alleen dan werd de shim-loader geïnstalleerd. In latere versies is ondersteuning voor Secure Boot geleidelijk verbeterd en kunnen meer systemen met ingeschakelde Secure Boot worden gestart.

Fedora ondersteunt UEFI sinds versie 14 (november 2010) en Secure Boot vanaf versie 18 (januari 2013) op dezelfde manier als openSUSE, zij het dat bij Fedora ook de kernelmodules met een Fedora-certificaat moeten zijn ondertekend. De reden daarvoor zou zijn dat Microsoft dit eist voor het verkrijgen van een Secure Bootondertekening door Microsoft. Anders zou de veiligheid van UEFI worden gecompromitteerd.

Op de websites van de drie hierboven genoemde distributies vindt u de nodige informatie over UEFI, Secure Boot en hoe de distributie daarmee omgaat. Er zijn meer Linux-distributies die Secure Boot ondersteunen, zoals Arch, Gentoo en Slackware. Vooral Arch geeft erg veel informatie over UEFI op zijn wiki-website: wiki.archlinux.org/index.php/Unified_Extensible_Firmware_Interface, alles in het Engels.

Tot slot

Algemeen kan worden gesteld dat de ondersteuning van Secure Boot enkele jaren geleden experimenteel begon en in de loop der tijd beter werd. Niettemin is het nog steeds geen uitgemaakte zaak dat het lukt om een Linux-systeem met ingeschakelde Secure Boot te installeren en te draaien.

Succes!

Noten

1 Deze module wordt normaliter automatisch geladen. In de Linux distributie Fedora is deze module zelfs vast in de kernel opgenomen. Hij maakt sinds versie 3.8 deel uit van de Linux kernel.

2 Ik heb hem geprobeerd met een paar Linux distributies: Ubuntu, openSUSE en Fedora. Dat werkte goed, zij het dat Ubuntu verzuimde zijn bootloader in de standaard locatie (EFI/BOOT) te plaatsen, waardoor het systeem niet startte. Dat was gemakkelijk op te lossen door hem vanuit de EFI shell handmatig te starten en hem eenmaal in Ubuntu op de juiste plek te zetten.

3 In echte hardware implementaties kan het anders gaan. Zo komt het voor dat na het indrukken van de F11 toets bij het opstarten het Boot Manager menu verschijnt.

4 Onlangs had ik te maken met een HP computer met UEFI en Windows 8, waarin een PK van HP zelf was opgenomen.

Links

URL1: http://www.linuxfoundation.org/publications/making-uefi-secure-boot-work-with-open-platforms
URL2: http://www.linuxfoundation.org/news-media/blogs/browse/2012/10/linux-foundation-uefi-secure-boot-system-open-source