Encryptie voor pc en laptop

Ton Valkenburgh

**Cybercrime bestaat al lange tijd. We horen er vaak over in de media.
Maar soms nemen de aanvallen exponentieel toe.
Hoog tijd om maatregelen te treffen. Encryptie is er daar een van.**

Door de oorlog in Oekraïne is er op dit moment meer aandacht voor cybercrime. Daardoor willen we wel eens vergeten dat er ook nog ouderwetse inbrekers actief zijn. In het verleden heb ik daar al eens een artikel aan gewijd. In de tussentijd is er echter wel het een en ander veranderd. Ik heb daarom het oorspronkelijke artikel aangepast aan de huidige stand van zaken. In het belang van de leesbaarheid zit er wel enige herhaling in ten opzichte van het oorspronkelijke artikel. De hier gegeven informatie kan je helpen om de informatie op je pc of laptop te beschermen bij fysieke inbraken, maar geeft geen beveiliging voor inbraken via Internet. Daarvoor zijn andere middelen beschikbaar.

1. Inleiding

De pc of laptop – hierna kan je voor laptop ook pc lezen – bevat steeds meer gegevens van ons. Niet alleen bankgegevens, maar ook kopieën van identiteitspapieren en privéfoto's. Bij het verlies of een diefstal willen we niet dat deze gegevens in handen van anderen vallen. Hoe kunnen we ons het beste daartegen beschermen en waarop moet je dan letten? Ik wil hier een aantal mogelijkheden uitdiepen die je kunnen helpen bij het beslissen wat voor jou de beste oplossing is. Houd altijd in je achterhoofd dat alles is te kraken. De kunst is te zorgen dat het zo lang duurt dat je na het constateren van de inbraak bijvoorbeeld wachtwoorden hebt kunnen veranderen. Anderzijds kun je met lange wachtwoorden (>25 karakters) zorgen dat het kraken zó lang gaat duren dat het niet interessant is om op het resultaat te wachten.

2.Encryptie

Met versleutelen kun je ervoor zorgen dat, als de laptop in handen valt van anderen, ze de gegevens niet kunnen lezen. Dergelijke encryptie kan op de hele schijf, maar ook op alleen de bestanden gebeuren. Is het voldoende om alleen belangrijke bestanden te versleutelen? Wie weet of de programma's die hij gebruikt gegevens of sporen van gegevens achterlaten, en zo ja waar? Hoevelen van ons realiseren zich dat bij de optie 'snel opstarten' de inhoud van het werkgeheugen op schijf wordt opgeslagen? In het wisselbestand en het hibernate-bestand staan ook gegevens. Gewoon wissen bij afsluiten van de laptop is niet voldoende. De gegevens staan dan nog steeds op schijf. Wissen wil namelijk niet zeggen dat het echt van de schijf is verwijderd. Alleen de referenties naar de gegevens zijn verwijderd. Het overschrijven met willekeurige getallen i.p.v. standaard wissen werkt wel bij een harde schijf, maar niet bij een SSD. Moderne laptops hebben steeds vaker een SSD i.p.v. een harde schijf. Omdat we eigenlijk zo weinig weten van onze trouwe kameraad, is het verstandig om de hele schijf, of eigenlijk nog beter alle schijven, te versleutelen. We hebben dan de volgende opties:

- Software-encryptie;
- Self Encrypted Drive (SED).

Programma's voor software-encryptie hebben in het algemeen veel mogelijkheden. Terwijl *Self Encrypted Drives* in het gebruik praktisch transparant zijn voor de gebruiker. Gezien mijn eigen ervaring concentreer ik me op de platformen Windows en Linux op x86-AMD64-systemen. Met Mac heb ik geen ervaring, en wat de mogelijkheden zijn met de nieuwe M1- en M2-machines weet ik niet.

3.Software-encryptie

Software-encryptie is flexibel en biedt de mogelijkheid om de schijf, partitie, containers of bestanden te versleutelen. Er is wel enige impact op de performance van de laptop, maar moderne systemen zijn zo krachtig dat dit verwaarloosbaar is. Zeker als er gebruik wordt gemaakt van de CPU-eigenschap *AES New Instructions*. Een nadeel is dat ze vaak afhankelijk zijn van het platform waarop het systeem draait. Dat kan vervelend zijn als je meer dan één type platform gebruikt. Ook moet er rekening worden gehouden met de technologie van de schijf. SSD's en hybride schijven en USB-sticks werken anders dan een harde schijf. Zoals eerder gesteld, kan, na het wissen van data bij een Solid State Drive (SSD) deze data altijd nog staan in andere cellen. Ze komen zelfs in het deel van de SSD dat is gereserveerd voor *overprovisioning*. Het is daarom verstandig bij het gebruik van software-encryptie op dergelijke media (flash drives) uit te gaan van ongebruikt of schoon (veilig gewist) materiaal.

3.1.Windows

Ik wil hier niet alle programma's voor Windows behandelen, dat zou het artikel veel te lang maken. Ik beperk me dus tot de bekendste, respectievelijk interessantste:

- Veracrypt;
- Encrypted File System (EFS);

BitLocker.3.1.1.VeraCrypt

VeraCrypt (link 1.) is een open source-programma dat is voortgekomen uit TrueCrypt. De ontwikkelaars van TrueCrypt zijn ermee gestopt. In VeraCrypt is een aantal problemen van TrueCrypt opgelost en zijn er nieuwe mogelijkheden bij gekomen. VeraCrypt werkt op de platformen Windows, Linux, Mac OS-X en Raspbian.

Het ondersteunt het versleutelen van schijven, partities, containers en USB-schijven/sticks. Het versleutelen kan 'on-the-fly' plaatsvinden, zonder dat gegevens verloren gaan.

VeraCrypt staat toe om diverse encryptiemethodes op elkaar te stapelen. Dat maakt het National Security Agency (NSA)-bestendig (voorlopig?).
Voor Windows is het ook mogelijk de systeemschijf te versleutelen. Voor de andere platformen geldt dit niet. Het kent verder een zogenaamde verborgen partitie. Het is dan niet zichtbaar dat er een versleuteld systeem is. Van USB-schijven en sticks kun je een *traveler versie* maken.

Verder kun je de oude TrueCrypt-versleuteling gebruiken of deze, zonder verlies van gegevens, omzetten naar VeraCrypt-versleuteling.

Nieuwe versies van VeraCrypt kunnen worden geïnstalleerd zonder dat je eerst het systeem moet decrypten.

3.1.2.EFS
EFS is aanwezig in de zakelijke versies van Windows. EFS kan schijven, partities en bestanden versleutelen. Vanaf Windows 2000 zijn de functies stap voor stap uitgebreid. Omdat dit voor de zakelijke markt is, ga ik er verder niet op in.

3.1.3.BitLocker
BitLocker (link 2.) is de Microsoft-oplossing voor encryptie. Het is aanwezig in de zakelijke versies van Windows. Om het te kunnen gebruiken moet aan de volgende voorwaarden worden voldaan: de laptop moet minstens Windows 10 en een Trusted Platform Module versie 1.2 hebben of er moet een USB-stick worden gebruikt om op te starten. Omdat thuisgebruikers meestal Windows Home hebben, vind ik dit geen interessante optie.

3.2.Linux
Voor Linux zijn de volgende programma's interessant:
- Dm-crypt met Linux Unified Key Setup (LUKS);
- Encrypted File System (EncFS);
- VeraCrypt;
- ZuluCrypt/Mount.

Ik zal ze niet allemaal even uitgebreid behandelen.

3.2.1.LUKS
LUKS met dm-crypt kan schijven, partities, logical volumes, containers, bestanden en USB-schijven/sticks versleutelen. Het is voor het Linux-platform, hoewel er een niet meer onderhouden Windows-versie LibreCrypt bestaat. Niet onderhouden betekent niet meer veilig. LUKS kan ook de systeemschijf (als een Logical Volume) versleutelen. Bij de installatie van Ubuntu en daarvan afgeleide varianten is het versleutelen van de systeemschijf opgenomen in de installatieprocedure: een vinkje zetten en het wachtwoord invoeren. Dat is alles. Dit kan dus ook een niet-ervaren gebruiker doen. Ook is er de mogelijkheid om alleen het home-gebied te versleutelen. Dat geeft een slechtere performance en is daarom niet aan te raden.

3.2.2.EncFS
Het open source programma EncFS is het eenvoudigste encryptiesysteem op Linux. Het vereist geen root-autorisatie. Het wordt door o.a. het back-upprogramma 'Back in Time' gebruikt. Een belangrijk nadeel is, dat als het bestand tweemaal versleuteld is op een verschillend tijdstip, met de combinatie van beide bestanden de versleuteling is te breken. In de laatste versie is wel een aantal problemen opgelost, maar bij het installeren van EncFS onder Linux wordt er nog steeds een waarschuwing gegeven betreffende de huidige zwakheid van EncFS.

Van EncFS is zowel een Windows- als een Mac OS-X-versie beschikbaar.

3.2.3.VeraCrypt
Van VeraCrypt wil ik alleen nog benadrukken dat het onder Linux geen systeemdisk kan versleutelen. Voor de rest verwijs ik naar wat onder Windows is beschreven.

3.3.4.ZuluCrypt/Mount
ZuluCrypt/Mount (link 3.) is een klein wonder. Het ondersteunt versleuteling volgens zowel dm-crypt met LUKS als VeraCrypt. De mount-optie maakt het eenvoudig om met één klik versleutelde schijven, partities of containers te openen.

3.4.Solid State Disk
Bij de Solid State Disk is het i.v.m. softwareversleuteling van belang om te kijken hoe *overprovisioning* en *garbage collection* werken.
Het schrijven op een SSD kan alleen naar een blok dat nullen bevat. De controller van de SSD ziet echter pas welke blokken zijn vrijgekomen als er een poging wordt gedaan om ernaar te schrijven. De controller schrijft dan naar een schoon blok in het *overprovisioning*-deel. Dit wordt dus nu deel van de systeempartitie. Als er geen schone blokken meer beschikbaar zijn, moet een blok eerst worden gewist alvorens ernaar geschreven kan worden. Dit vertraagt de schrijfactie.

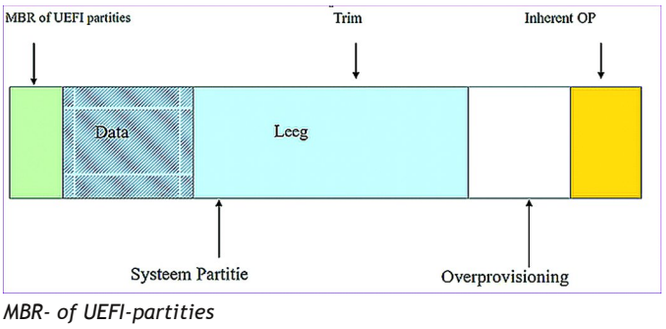
De fabrikant heeft daarom een deel met lege blokken voor *overprovisioning* gereserveerd. De gebruiker kan hier niet bij. Deze inherent *overprovisioning* kan voldoende zijn bij weinig schrijfacties op de SSD. De gebruiker kan de *overprovisioning* ruimte groter maken door een deel van de schijf niet te alloceren voor een partitie.

Garbage collection maakt vrijgekomen blokken beschikbaar door ze te wissen en dit moet zo veel mogelijk gebeuren als er geen schrijfacties plaatsvinden.

Het bedrijfssysteem weet welke blokken op de partitie vrij zijn gekomen. Door middel van het trimcommando wordt de SSD-controller geïnformeerd over welke blokken vrij zijn. Deze worden dus opgenomen in het *garbage collection*-proces. Het deel van de partitie waar geen data staat, noemen we het dynamische *overprovisioning* deel.

Je kunt de benodigde statische *overprovisioning* verkleinen door het trimcommando vaker uit te voeren en dus beter gebruik te maken van de dynamische *overprovisioning*.

Windows doet de trim op wekelijkse basis. Bij Linux is dit instelbaar naar bijvoorbeeld dagelijks.



MBR- of UEFI-partities

Voor VeraCrypt en LUKS wordt afgeraden om het bij SSD's te gebruiken. De achtergrond hiervoor is dat gegevens niet worden overschreven, zelfs als de opdracht is om de oude gegevens te overschrijven. Ook wordt afgeraden om trim te gebruiken omdat dan meer van de structuur van de data zichtbaar wordt voor het kraken van de encryptie. Helaas wordt de SSD dan al snel trager. Bij het gebruik op SSD's of USB-sticks wordt aangeraden in ieder geval uit te gaan van een schoon medium. Bij de SSD moet er dan ook geen overprovisioning worden gebruikt. Het is dan wel aan te raden de TRIM op dagelijks te zetten. Voor meer informatie: lees de documentatie van VeraCrypt en LUKS.

Er is echter een betere encryptie oplossing voor SSD's: de Self Encrypted Drive (link 4.).

4.Self Encrypted Drive (SED)

Self Encrypte Drives zijn verkrijgbaar als hard disks en *Solid State Drives*. Met de SSD's is de bekendheid toegenomen, maar niet iedereen weet hoe de functie is te gebruiken. Er zijn drie uitvoeringen. Vaak allemaal beschikbaar in één SSD:
- SATA-encryptie;
- Trusted Computing Group Opal Security Subsystem Class;
- Encrypted Drive (eDrive).

De gegevens op een *Self Encrypted Drive* zijn altijd versleuteld. De controller regelt dit. De bovengenoemde opties zijn de methodes om de toegang tot de versleutelde gegevens te vergrendelen en te ontgrendelen. Standaard is de toegang open en merkt de gebruiker niet dat zijn gegevens zijn versleuteld. Een *secure erase* verandert de versleute-

ling. De SSD is dan ook weer ontgrendeld, maar toegang tot de oude gegevens is geblokkeerd.

De vergrendeling kan ongedaan worden gemaakt zonder dat de gegevens op de SSD verloren gaan. Dat maakt het ook relatief eenvoudig om van vergrendelingsmethode te wisselen.

4.1.SATA-encryptie
SATA-encryptie wordt ook wel class 0 genoemd. De vergrendeling wordt via het BIOS/UEFI-harddisk-wachtwoord geregeld. Niet alle BIOS/UEFI-uitvoeringen ondersteunen dit. Een nadeel is dat het wachtwoord alleen uit kleine alfanumerieke tekens mag bestaan. Dit geeft dus een beperkte tekenset. Mijn ervaring met mijn ASUS-laptops is, dat je de wachtwoorden niet te snel moet intikken. Het wachtwoord wordt dan soms niet herkend, omdat er karakters verloren zijn gegaan. Deze vorm wordt niet meer als veilig gezien. Sommige fabrikanten van SED's bieden het daarom ook niet meer aan.

4.2.Opal
De trusted Computer Group heeft de Opal-specificatie vastgelegd om tot een fabrikant-onafhankelijke uitvoering te komen. Er is extra software nodig om de autorisatie uit te voeren.

Deze software wordt in de SSD opgeslagen, maar is niet standaard aanwezig. De software kan bij commerciële bedrijven worden gekocht, maar er is ook een open source-versie verkrijgbaar. Omdat de software niet standaard in de drive zit, wordt deze methode minder gauw gebruikt. Bij sommige fabrikanten bleek deze methode niet veilig. Nieuwe *firmware* moest dit oplossen. Gebruik daarom altijd de laatste firmware-versie.

4.3.eDrive
De eDrive is een door Microsoft uitgebreide Opal-implementatie volgens IEEE 1667. Het vereist UEFI 2.3.1, BitLocker en TPM 1.2 of een USB-stick.

4.4.Drive Trust Alliance
De Drive Trust Alliance (link 5.) heeft als doel voor bekendheid en implementaties van de *Self Encrypted Drive* te zorgen. Hun focus ligt op de *Opal*-methode. Er is een *open source*-uitvoering van de benodigde software voor de SSD.

We gaan met de versie *open source* aan de slag. Om een SED te versleutelen is de beste methode een *boot rescue USB-stick* aan te maken. Deze USB-stick kan later altijd worden gebruikt in het geval van problemen. We halen een image voor de USB-stick op van de website (link 6.). Er is een 32-bit BIOS- en een 64-bit UEFI-versie. Bij een UEFI-machine moet *secure boot* worden uitgeschakeld. De versleuteling verandert de gegevens op de SSD niet, maar ik raad aan voor alle zekerheid een back-up van de SSD te maken.

Ik beschrijf de procedure voor Linux, maar geef voldoende informatie voor Windows-gebruikers. Haal het gecomprimeerde Rescue-bestand op.

Decomprimeer het Rescue-bestand (bij Windows gebruik je bijvoorbeeld 7-Zip (link 7.) met:
```
gunzip RESCUE32.img.gz
 --of--
gunzip RESCUE64.img.gz
```

Nu gaan we het image-bestand op de USB-stick zetten. Steek de USB-stick in een USB-slot.

Bij Linux gebruiken we het programma *Schijven*. Bij *Drive Options* kies je *Restore Disk Image*. Het gewenste image zet je op deze manier op de USB-stick.

Bij Windows kun je Win32DiskImager (link 8.) gebruiken om het image-bestand naar de USB-stick te schrijven.

We starten de laptop nu op vanaf deze USB-stick. Je krijgt dan een inlog-prompt te zien: vul in *root*. Er is geen wachtwoord.

De volgende instructies voer je uit na het inloggen op de Rescue-stick.

We gaan eerst de SSD opzoeken met het volgende commando:
```
sedutil-cli --scan
```

Je krijgt dan bijvoorbeeld het volgende bericht:
```
#sedutil-cli --scan
Scanning for Opal compliant disks
/dev/nvme0  2  Samsung SSD 960 EVO 250GB  2B7QCXE7
/dev/sda    2  Crucial_CT250MX200SSD1  MU04
/dev/sdb   12  Samsung SSD 850 EVO 500GB  EMT01B6Q
/dev/sdc    2  ST500LT025-1DH142  0001SDM7
/dev/sdd   12  Samsung SSD 850 EVO 250GB  EMT01B6Q
No more disks present ending scan
```

De 2 of 12 in de tweede kolom geeft aan dat de SSD *TCG Opal 2* ondersteunt.

Je kunt de status van een specifieke SSD, bijvoorbeeld de NVMe-SSD, opvragen met het volgende commando:
```
sedutil-cli --query /dev/nvme0
```

Je kunt nu een SSD die *Opal 2* ondersteunt gaan prepareren. N.B. Het kan zijn dat je bij de volgende commando's een melding krijgt dat je niet bent geautoriseerd. Dit komt voor bij o.a. Samsung-SSD's. Dit los je op door de SSD te resetten met *PSID revert*. De PSID is te vinden op de SSD. Deze procedure wist de gegevens op de SSD. Eventueel moet je dus eerst een systeem-back-up maken van de SSD. Dat kun je doen onder Linux met *Schijven* of bij Windows met *Macrium Reflect* (link 9). Om de SSD te resetten geven we, bijvoorbeeld voor de NVMe-SSD, het volgende commando:
```
sedutil-cli --yesIreallywanttoERASEALLmydata
                                    usingthePSID
<PSID> /dev/nvme0
```

waarin <PSID> het PSID is.

Hierna worden de volgende commando's geaccepteerd.

We gaan nu de drive klaarmaken met een aantal commando's (voorlopig gebruiken we als wachtwoord *debug*). Ik blijf de NVMe-SSD als voorbeeld gebruiken.:
```
sedutil-cli  --initialsetup debug /dev/nvme0
sedutil-cli  --enablelockingrage 0 debug
                                        /dev/nvme0
sedutil-cli  --setlockingrage 0 lk debug
                                        /dev/nvme0
sedutil-cli  --setmbrdone off debug /dev/nvme0
```

Om van de SSD een boot-drive te maken moeten we het PBA-image laden. Dat kan vrij lang duren, dus word niet ongeduldig.
Daarbij gebruiken we het *debug*-wachtwoord. Voor een UEFI-systeem:
```
gunzip /usr/sedutil/UEFI64-tarball.img.gz
sedutil-cli  --loadpbaimage debug /usr/sedutil/
                          UEFI64-*.img  /dev/nvme0
```

Nu gaan we de SSD vergrendelen met de volgende commando's: (<password> is het wachtwoord dat je wilt gaan gebruiken)
```
sedutil-cli --setsidpassword debug <password>
                                        /dev/nvme0
sedutil-cli --setadmin1pwd debug <password>
                                        /dev/nvme0
sedutil-cli --setmbrdone on <password>
                                        /dev/nvme0
```

Om nu de vergrendeling te effectueren schakelen we de laptop uit. Daarna starten we de laptop weer op. De SSD vraagt dan om het door ons geïnstalleerde wachtwoord. Na invoer van dit wachtwoord wordt de SSD ontgrendeld. Dan herstart het systeem en start het besturingssysteem op vanaf de SSD. De SSD zal weer worden vergrendeld zodra de spanning van de SSD af is geweest.
De vergrendeling kan worden gedeactiveerd met de volgende commando's:

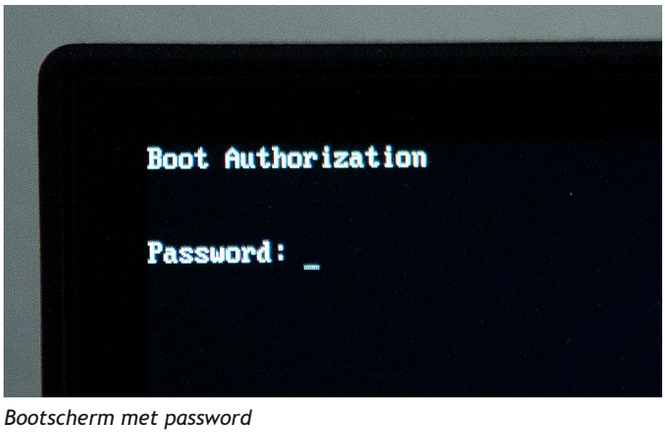
```
sedutil-cli --disablelockingrange 0 <password>
                                          /dev/nvme0
sedutil-cli --setMBREnable off <password>
                                          /dev/nvme0
```

Wil je de vergrendeling van de SSD weer activeren, geef dan de volgende commando's:

```
sedutil-cli --enableLockingRange 0 <password>
                                          /dev/nvme0
sedutil-cli --setMBREnable on <password>
                                          /dev/nvme0
```

Als je de laptop nu opstart, krijg je het loginscherm.
Na het invoeren van het wachtwoord herstart de laptop en start het besturingssysteem.

Als er een aantal SED's in je machine zit, geef ze dan allemaal hetzelfde wachtwoord. Bij het opstarten hoef je dan maar eenmaal een wachtwoord op te geven. Alle SED's worden dan ontgrendeld.



Bootscherm met password

5.Conclusie

Welke encryptiemethode je het beste kun kiezen, is afhankelijk van type schijf en het gebruikte platform. Laten we uitgaan van de schijf.

5.1.Hard Disk Drive

De goedkoopste optie is om hier software-encryptie te gebruiken. Dat is voor Windows: *VeraCrypt* en voor Linux *LUKS*.

5.2.SSD

Bij de aanschaf van een SSD moet je eigenlijk kiezen voor een *Self Encrypted Drive*. Let er wel op dat *TCG Opal 2* wordt ondersteund.
Ik heb hierboven de versie voor systemen met een processor X86-AMD64 gegeven. Bij het gebruik van een AMD Ryzen-CPU is de Sedutil-versie 1.20 of hoger- indien verschenen - noodzakelijk. Er is ook een implementatie voor Arm-processors in ontwikkeling (link 10.).

5.3.USB-sticks

De USB-sticks worden zowel met hardware encryptie als met software encryptie geleverd. Bij software encryptie kun je beter zelf de encryptie regelen met bijvoorbeeld *VeraCrypt*. Eventueel uitgevoerd als een *traveler-disk*. Besef wel dat er hardwareversleutelingen zijn die heel makkelijk zijn te omzeilen.

Interessante links

1. https://www.veracrypt.fr/en/Home.html
2. https://learn.microsoft.com/en-us/windows/security/information-protection/bitlocker/bitlocker-overview
3. https://mhogomchungu.github.io/zuluCrypt/
4. https://www.trentonsystems.com/blog/self-encrypting-drives
5. https://www.drivetrust.com/
6. https://sedutil.com/
7. https://www.7-zip.org/
8. https://sourceforge.net/projects/win32diskimager/
9. https://www.macrium.com/reflectfree
10. https://voidlinux.pkgs.org/current/voidlinux-main-aarch64/sedutil-1.20.0_1.aarch64.xbps.html