Estudo de Criptografia para HDDs e SSDs

Autores: Alessandro Wagner Palmeira e Caio Renato Bedulli do Carmo
Supervisor: Prof. Dr. Marco Dimas Gubitoso
Instituto de Matemática e Estatística - Universidade de São Paulo

Introdução

Segurança de informação é um tema recorrente em nossa sociedade há séculos. Sendo assim, o presente trabalho visa o estudo de dois métodos que objetivam *Full Disk Encryption* (FDE): *cryptsetup* em conjunto com *Linux Unified Key Setup* (LUKS) e o utilitário SEDutil (interface para OPAL) e suas interações com o Sistema Operacional e a máquina.
Disponibilizamos diferenças no desempenho quando testados esses métodos com os dispositivos HDD e SSD de uso próprio, vulnerabilidades as quais estão expostas como ataque DMA, *Cold Boot* e *Evil Maid* e como reduzir tais exposições.
No que tange a segurança, nossa revisão literária mostrou que ambos os métodos são vulneráveis se não houver outra medida de prevenção usada em conjunto como BitVisor, Qubes OS, senha na BIOS para inicialização, suporte na CPU de IOMMU e TPM.

SEDutil

O projeto SEDutil tem como objetivo uma implementação livre de FDE usando SED em dispositivos compatíveis com a especificação OPAL 2.0 ou superior descrita pelo órgão *Trusted Computing Group* (TGC), órgão esse que se preocupa em criar padrões abertos e livres para segurança de dados. A OPAL propõe que fabricantes de armazenamento como *Samsung*, *Intel* e OCZ implementem uma API em seus dispositivos para que, quando chamados de forma correta, o dispositivo tenha uma *Master Boot Record* (MBR) falsa para que no momento de *boot* da máquina, a BIOS consiga carregá-la e então com uma imagem mínima de S.O., como *Linux Pre Boot Authorization* (linuxPBA), é trazida à memória e assim pedido ao usuário que insira a senha do dispositivo. Quando a senha correta é inserida, o dispositivo é então liberado, mostra-se realmente suas partições internas e MBR presente.

Detalhes do SEDutil

- Dispositivos que suportam OPAL tem como responsabilidade criptografar e decriptografar todos os dados armazenados internamente regidos por AES128 ou AES256, a escolha do fabricante e com algum tipo de *hash* para confronto das senhas, tornando toda operação de segurança transparente para usuários e S.O.s.
- Desse modo, como a segurança é feita no próprio dispositivo, a CPU fica totalmente livre do encargo de tal operação e enquanto os processos esperam pelos dados, a CPU pode continuar outras tarefas escalonadas até que receba uma interrupção de E/S.
- O binário SEDutil tem como responsabilidade se comunicar de forma adequada com a API disponível para que corretamente configure as travas e a partição escondida de MBR.

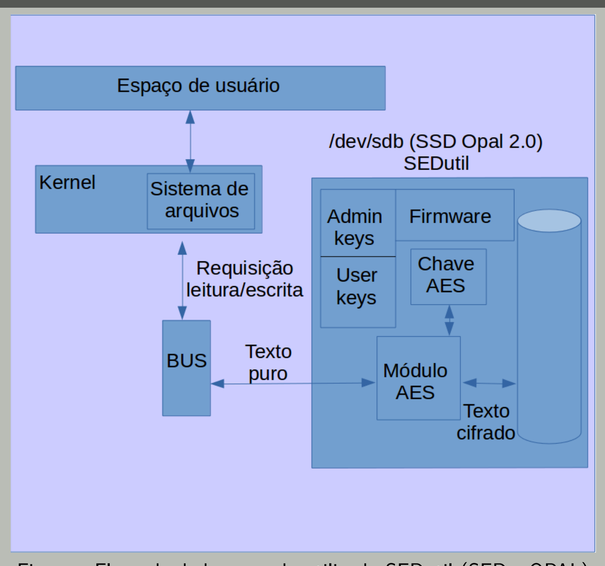


Figura : Fluxo de dados quando utilizado SEDutil (SED - OPAL)

LUKS

O sistema de criptografia LUKS é usado em conjunto com o pacote *cryptsetup* ou *dm_crypt* para criptografia de dados em dispositivos de armazenamentos secundário. Entretanto, toda operação nesse método é feita pela CPU e os dados passado para o BUS que liga o dispositivo à CPU ou RAM (através de DMA) saem ou entram criptografados. Diferentemente de SED, podemos escolher entre criptografar somente uma partição específica, todo o dispositivo ou até mesmo criar um "cofre" (um arquivo em uma partição, inicializá-lo com o *cryptsetup* + LUKS e então formatá-lo com algum sistema de arquivos como *extN*, *FAT*, *NTFS* etc. e depois utilizar o sistemas de *loop*).

Detalhes do LUKS

- Chaves de até 8MB para abrir a chave *Advanced Encrypting Standard* (AES).
- Transparência de implementação.
- Pode ser utilizado com ou sem suporte nativo às novas instruções de AES.
- Dados percorrem o barramento de forma criptografada.
- Porém demanda maior utilização por parte da CPU, principalmente quando não há suporte a AES e facilmente pode se ter os dados perdidos caso não há uma cópia de restauração de seu cabeçalho.

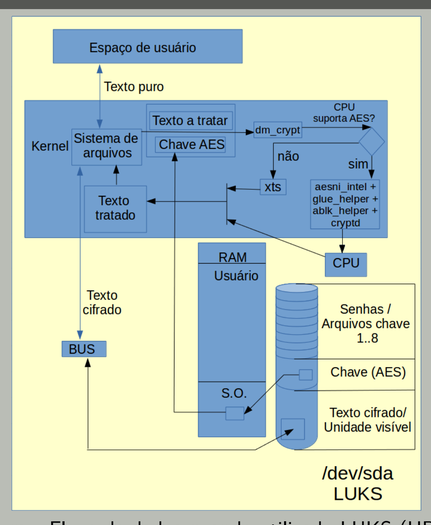
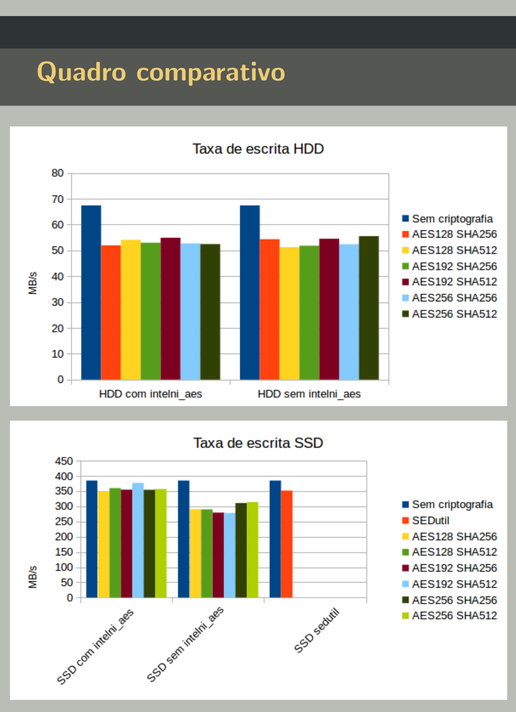
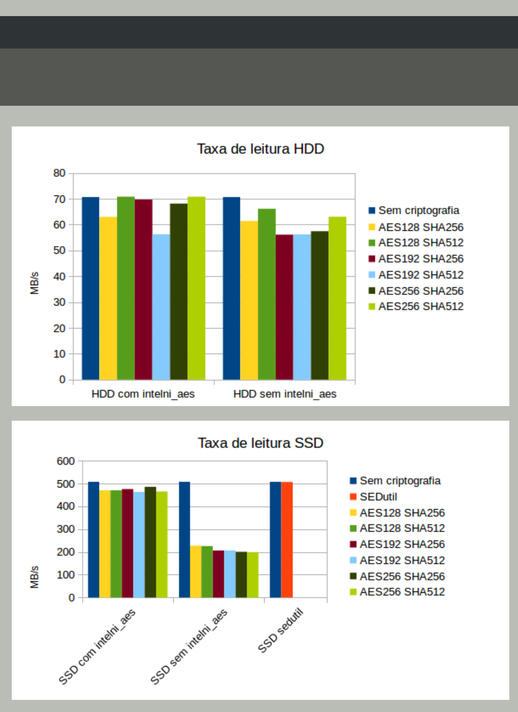


Figura : Fluxo de dados quando utilizado LUKS (HDD)

Quadro comparativo





Ataques

Ambas as abordagens de FDE são vulneráveis aos seguintes ataques quando não utilizadas em conjunto com algum tipo de prevenção:

- DMA. Nesse tipo de ataque, podemos ler e modificar regiões da memória RAM inserindo algum dispositivo malicioso em um entrada que suporta DMA.
- *Cold Boot* (LUKS). Com esse método, podemos ler informações retidas na RAM mesmo após o desligamento da máquina.
- *Hot Swap* (OPAL). Como as chaves criptográficas são armazenadas no dispositivo, podemos inserir um segundo par de cabos, energia e SATA, enquanto o dispositivo está destravado e ligá-lo em outra máquina. Já nessa outra máquina, tendo controle do S.O., podemos ler todo o dispositivo.
- *Evil Maid*. De posse da máquina mas sem a o conhecimento da pessoa dona dela, temos dois caminhos: Caso esteja usando LUKS, a MBR está não criptografada, possibilitando assim a reescrita da MBR para que, quando digitada a senha, esta seja enviada por rede ou armazenada na própria MBR. Caso esteja usando OPAL, a MBR está protegida contra escrita. Porém é possível a substituição do dispositivo verdadeiro por um similar contendo uma MBR com um binário que se comporte de forma análoga a anterior.

(a) Cabo usado em *Hot Swap*.

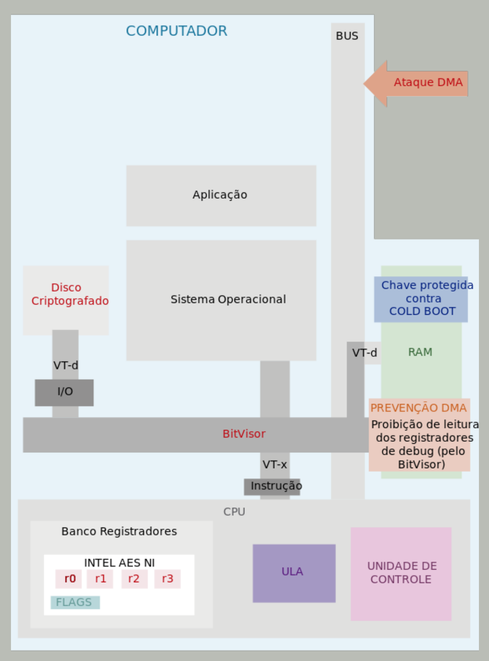
(b) Memória refrigerada para leitura de *bits* após o desligamento da máquina.

Figura : Ilustrações de ataques a FEDs

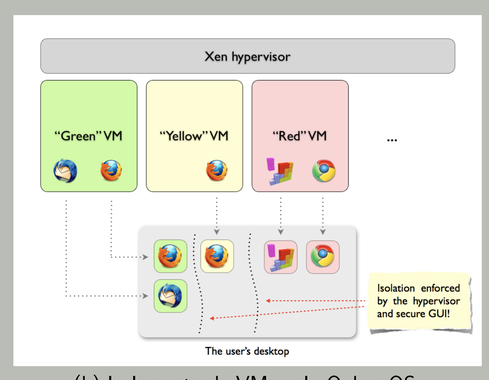
Prevenções

Para minimizar a possibilidade de ataques, encontramos os seguintes métodos:

- IOMMU: suporte por parte da CPU que, entre outros objetivos, protege acesso indevido a RAM quando ocorrem operações de DMA.
- TreVisor: *hypervisor* mínimo que, com suporte do IOMMU, previne ataque DMA; por ter as chaves criptográficas nos registradores de depuração, protege contra *Cold Boot*. Se utilizado em conjunto com TPM, dá-se proteção contra *Evil Maid*.
- Qube-OS: Sistema Operacional que, através do XEN, garante isolamento dos programas sendo executados. Cada *Qube* tem sua própria chave de acesso ao disco, que está criptografado. Utiliza-se do IOMMU e oferece utilização de TPM, garantindo assim proteção contra *Evil Maid* e ataque *DMA*. Porém não suporta prevenção a *Cold Boot* no momento.



(a) Arquitetura do TreVisor.



(b) Isolamento de VMs pelo Qubes-OS.

Figura : Ilustrações de algumas medidas de prevenção contra ataques a FED.

Conclusões

Na análise de desempenho, constatou-se que nos dispositivos testados, houve uma redução entre 18,2% e 22,6% com suporte a AES e entre 17,28% e 23,73% sem suporte a AES na escrita no HDD e de 8% no SSD em relação à não utilização de criptografia (OPAL/SEDutil). Considerou-se esses valores aceitáveis, dado que ambas as taxas de leitura se mantiveram iguais e os dados foram mantidos em sigilo.
A literatura nos mostra que tanto o *cryptsetup* quanto o OPAL são vulneráveis a algum tipo de ataque porém há meios de prevenção contra esses ataques, incluindo a utilização correta do IOMMU, uso do TreVisor ou Qubes-OS, *boot* a partir de um *pendrive* para autenticar a máquina para o usuário e uso de TPM.

Referências

1. https://www.qubes-os.org/
2. Yitbarek et. al (2017) Cold Boot Attacks are Still Hot: Security Analysis of Memory Scramblers in Modern Processors
3. Müller et at (2012) Trevisor:Os-independent software-based full disk encryption secure against main memory attacks
4. Boteanu e Fowler(2015) Bypassing self encrypting drives (sed) in enterprise environments.
5. https://gitlab.com/cryptsetup/cryptsetup
6. https://blog.invisiblethings.org/2011/09/07/anti-evil-maid.html

Figura : Visite nossa página.