Hardware - Séquence de démarrage

Difficulté : Débutant

Notions : Systèmes, matériel, séquence de démarrage.

I. Introduction

Entre le moment où l'on presse le bouton '***start***' et le moment où l'on arrive sur le bureau, il se passe beaucoup de choses.

Comprendre la séquence de démarrage permet de diagnostiquer les pannes, optimiser les performances, sécuriser le système et mieux comprendre l’architecture matérielle/logicielle. C’est une compétence clé pour tout technicien ou administrateur système.

Mais également pour bien comprendre les aspects matériels de la cybersécurité.

II. Phase matérielle

Voici à quoi ressemble la séquence côté matériel :

2.1 Power On

Au moment de l'appui, le bouton d’alimentation ferme un circuit électrique qui envoie un signal à l’alimentation (PSU). Celle-ci fournit une tension stable aux composants (CPU, RAM, chipset, périphériques). Le système passe de l’état S5 (Soft Off) à S0 (Fully On).

La PSU vérifie que les tensions sont stables. Il envoie ensuite le signal Power Good à la carte mère, indiquant que le CPU peut commencer l’exécution. Sans ce signal, aucun démarrage n’est possible.

2.2 Init firmware (BIOS / UEFI)

Le CPU démarre à une adresse fixe en ROM où se trouve le BIOS/UEFI.

Il charge son microcode interne, puis exécute le firmware.

> **Info** : Son microcode peut être mis à jour lors des mises à jour BIOS / UEFI

Le firmware initialise alors les composants essentiels : RAM, contrôleurs, bus, clavier, GPU, etc.

2.3 Le POST

Le POST, pour '***Power On Self Test***' (auto diagnostic de démarrage), vérifie l'intégrité minimale du matériel : RAM, CPU, GPU, clavier, stockage. En cas d'erreur, le système émet des bips ou affiche un code. Si tout est correct, le firmware passe à la phase suivante.

Par exemple :

- *RAM défectueuse non reconnue ou mal alimentée = bips répétés.*
- *RAM défectueuse = boucle de reboot*
- *Bootloader corrompu = "Operating System not found"*
- *Secure Boot = "Signature invalid"*
- *TPM = "BitLocker recovery key required"*

2.4 Initialisation périphériques

Le firmware détecte les périphériques bootables (SSD, HDD, USB, réseau). Il applique les paramètres UEFI/BIOS (ordre de boot, Secure Boot, mode AHCI/RAID, etc.). Il détecte et initialise également les périphériques spécifiques (cartes RAID, fond de panier, etc...)

2.5 *optionnel - Vérification d'intégrité via TPM (PCR)*

C'est à ce moment que TPM (**T**rusted **P**latform **M**odule) est chargé.

- Le TPM stocke des **mesures cryptographiques** (PCR) sur l'intégrité du système : firmware, configuration UEFI, bootloader, fichiers critiques.
- Lors du démarrage, l'UEFI **interroge le TPM** pour vérifier que l'environnement n'a pas été modifié.
- Si les mesures ne correspondent pas (rootkit, bootkit, modification du firmware), le TPM **refuse de libérer les clés de déchiffrement**.

Point sur le bootkit

- **Quest-ce qu'un bootkit ?**

Un **bootkit** est un malware qui s'installe dans la chaîne de démarrage, avant le système d'exploitation. Il modifie le firmware, le bootloader ou les fichiers de démarrage pour prendre le contrôle très tôt. Il permet à l'attaquant de charger du code malveillant avant que l'OS ne soit protégé. C'est l'une des formes les plus avancées de persistance.

- **Pourquoi est-il si difficilement détectable ?**

Il s'exécute **avant** l'antivirus, l'OS et les outils de sécurité. Il peut masquer ses fichiers, falsifier les mesures d'intégrité et contourner les logs. Les scanners classiques ne voient pas les modifications du firmware ou du bootloader. Il peut survivre à une réinstallation du système si le firmware est compromis.

- **Comment TPM et secure boot protègent des rootkits ?**

Secure Boot vérifie que le firmware, le bootloader et les binaires de démarrage sont **signés** et non modifiés. Le **TPM** enregistre des mesures cryptographiques (PCR) de chaque étape du boot. Si un bootkit modifie un composant, les signatures ou les PCR ne correspondent plus. Le système bloque le démarrage ou demande la clé de récupération (BitLocker), empêchant l'exécution du bootkit.

Le firmware compare les mesures stockées dans le TPM avec celles calculées au démarrage.
Si elles diffèrent, le système peut bloquer le démarrage ou refuser l'accès aux clés BitLocker.

III. Phase de Bootloader

3.1 Choix du périphérique de démarrage

En fonction des priorités définies dans le BIOS, le firmware va choisir le mode de démarrage et le périphérique concerné.

Le firmware lit le premier secteur du disque (MBR) ou la partition EFI (GPT). Il y trouve le bootloader (Windows Boot Manager, GRUB, etc.).

Le bootloader est un programme minimal chargé en mémoire. Il prépare le lancement du système d’exploitation : sélection du noyau, options de démarrage, gestion du multi-boot.

Si '***secure boot***' est activé, Celui-ci vérifie la signature du bootloader. Il empêchera alors les bootkits non signés de démarrer.

> **Conseil** : La plupart des BIOS / UEFI ont une fonction de '***boot override***' ou '***One Time Boot***' permettant de sélectionner un périphérique de démarrage spécifique exceptionnellement. Le système pourra alors démarrer sur ce périphérique en ignorant temporairement sa séquence de démarrage habituelle.

> **Info** : Il est également possible de choisir de démarrer sur le réseau. Entamant ainsi une séquence de démarrage spécifique qui ne sera pas couverte ici.

3.2 *optionnel - Déverrouillage du disque via BitLocker*

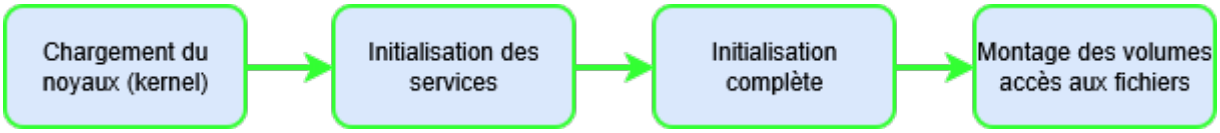
BitLocker intervient juste après le firmware, au moment du chargement du bootloader, et avant le chargement du noyau. Il chiffre le disque système (partition Windows). Pour déchiffrer le disque, il a besoin d’une clé de déverrouillage.

Cette clé est stockée dans le TPM ou protégée par un autre mécanisme (PIN, clé USB, mot de passe).

- Le bootloader tente d’accéder à la partition système.
- BitLocker demande la clé de déchiffrement.
- Le TPM libère la clé **uniquement** si l’intégrité du système est validée.
- Une fois le disque déchiffré, le bootloader peut charger le noyau.

IV. Phase logicielle

Voici à quoi ressemble la séquence côté logicielle :



4.1 Chargement du noyau

Le bootloader charge le noyau du système (ntoskrnl.exe pour Windows, vmlinuz pour Linux). Le noyau initialise les pilotes essentiels : gestion mémoire, processus, système de fichiers, drivers basiques.

4.2 Initialisation des services

Le noyau lance les services critiques :

- gestion des processus
- gestion des entrées/sorties
- pile réseau
- pilotes matériels
- sécurité (LSASS, SAM pour Windows)

4.3 Initialisation Complète

Le système procède ensuite à l'ensemble des composants non essentiels au démarrage.

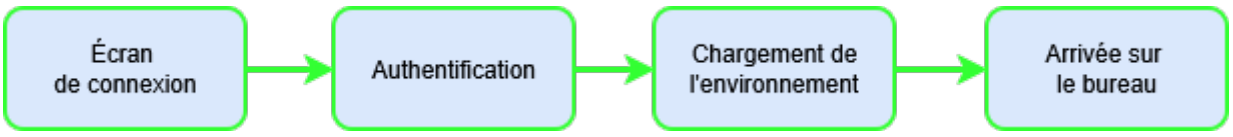
- L’initialisation des *drivers non essentiels*
- Le lancement du *service manager* (systemd, init, Windows SCM)
- La gestion des *runlevels* (Linux)
- Le chargement des modules du noyau (drivers dynamiques)

4.4 Montage des volumes / Accès au système

Le système monte les partitions nécessaires (C:, /root, /home…). Il charge les pilotes de stockage avancés (NVMe, SATA, RAID).

V. Phase Utilisateur

Voici à quoi ressemble la séquence côté Utilisateur :



5.1 Écran de connexion

Le système démarre l’environnement graphique (GDM, SDDM, Windows LogonUI). Les services d’authentification sont prêts (Kerberos, SAM, PAM). L'écran de connexion s'affiche.

> **Info** : En cas de connexion automatique, en l'absence de mot de passe par exemple, cette phase et la phase d'authentification se font quand même, mais sont juste automatisées et transparente pour l'utilisateur.

5.2 Authentification

Le système vérifie les identifiants, charge le profil utilisateur, applique les stratégies (GPO, scripts de login).

5.3 Chargement de l'environnement

Le système charge :

- l’explorateur (Explorer.exe, GNOME Shell, KDE Plasma)
- les paramètres utilisateur
- les applications au démarrage
- les services en arrière-plan

5.4 Arrivée sur le bureau

Le système est pleinement opérationnel. L’utilisateur peut interagir avec l’interface graphique, lancer des applications, accéder aux fichiers et aux ressources réseau.

> **Info** : Sur un système sans GUI, donc un système CLI, il n'y a pas d'initialisation d'environnement graphique. Mais cette phase à tout de même lieu à travers, le prompt

> d'authentification, l'authentification et le chargement de l'environnement utilisateur.

VI. Conclusion

La séquence de démarrage est un enchaînement précis de phases matérielles et logicielles. La maîtriser permet de diagnostiquer les pannes, optimiser les performances et comprendre le fonctionnement interne d’un système moderne. C’est une base essentielle pour tout professionnel de l’informatique.

Revision #4
Created 2026-08-12 09:45:13 UTC by Olivier
Updated 2026-08-12 11:12:54 UTC by Olivier