# Bientôt des terminaux mobiles sous Linux avec une sécurité transparente ?



Cette page web documente mes tentatives pour configurer des smartphones et tablettes sous Linux en minimisant le nombre de composants non standards, de binaires opaques et de backdoors potentielles. Actuellement mon meilleur candidat est une tablette Lenovo Yoga 2 851 sous Fedora 22.

**READ THE HYPERTEXT VERSION HERE:**
**http://www.pabr.org/compsec/auditabletablet/auditabletablet.fr.html**

| Historique des versions | | |
|---|---|---|
| 1.1 | 2015-10-08 | Version française. |
| 1.0 | 2015-09-23 | Première publication. |

# Table des matières

# 1. Motivations

Nos terminaux mobiles sont trop vulnérables pour être connectés directement à Internet :

- D'un côté, les ordinateurs portables ont maintenant les même fonctionnalités que les PC de bureau et les serveurs, y compris la virtualisation ("ring -1"), le System Management Mode ("ring -2") et l'Active Management Technology ("ring -3"). Tous ces mécanismes tendent à isoler le système d'exploitation du hardware. Comment raisonner sur la sécurité lorsqu'on ne sait plus si l'OS tourne vraiment sur le matériel ou seulement sur une machine virtuelle mise en place par le BIOS ? Ou lorsque la plate-forme contient un processeur auxiliaire qui peut espionner et contrôler la plupart des périphériques même lorsque le CPU est en veille ?

- D'un autre côté, les smartphones et tablettes sont encore souvent des systèmes verrouillés tournant sur du hardware peu documenté. Du point de vue de la sécurité, ils deviennent inutilisables après quelques années, lorsque les fabricants cessent de publier les correctifs indispensables.

Dans ce projet j'essaie de configurer un appareil mobile suffisamment transparent pour que sa sécurité puisse être auditée. En pratique cela signifie que je privilégie :

- **Les logiciels open-source .** Il n'est certes pas évident que les logiciels open-source soient intrinsèquement plus sûrs que les logiciels propriétaires, comme on l'a vu récemment (Heartbleed, Shellshock). Cependant, la disponibilité du code source facilite indubitablement les audits.

- **Les versions génériques des logiciels .** Les logiciels populaires sont gérés plus activement que leurs déclinaisons (*forks*) peu utilisées. D'un côté cela peut conduire à un développement un peu chaotique. D'un autre côté, les projets qui survivent dans ces conditions sont forcément soutenus par des pratiques de génie logiciel robustes. Par exemple, la branche principale du noyau Linux sert de cas d'étude à bon nombre de projets de recherche universitaires ; son code source est ainsi contrôlé quotidiennement par toutes sortes d'analyseurs statiques et autres outils automatiques. Les branches dérivées gérées par les fabricants de terminaux et de composants ne bénéficient pas de la même vigilance (voir par exemple CVE-2012-6422 [http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-6422]).

Notons que plusieurs projets affirment déjà installer Linux sur divers appareils Android :

- ***KBOX* [http://kevinboone.net/android_nonroot.html] .** Fournit une interface en ligne de commande UNIX. Il s'agit essentiellement de **busybox** et de quelques autres commandes dans un émulateur de terminal, sous la forme d'une appli Android.

- ***Linux Deploy* [https://github.com/meefik/linuxdeploy], *LinuxonAndroid* [http://sourceforge.net/projects/linuxonandroid/] .** Font tourner des distributions avec un serveur VNC dans un bac à sable **chroot**, sous Android.

- ***Ubuntu Touch* [https://wiki.ubuntu.com/Touch] .** Fait tourner Ubuntu avec une interface tactile, en parallèle avec un OS Android minimaliste dans un container LXC. Reste dépendant du noyau Android et des pilotes propriétaires associés.

# 2. État d'avancement

| | **Yoga Tablet 2 851** | **Yoga Tablet 2 830** | **Nexus 7 2013** |
|---|---|---|---|
| Release date | 2014-10 | 2014-10 | 2013-07 |
| Serial console | Maybe | Maybe | OK |
| Boot custom kernels | OK | | OK |
| Boot mainline kernel | OK | | OK |
| Framebuffer graphics | OK | | |

| | Yoga Tablet 2 851 | Yoga Tablet 2 830 | Nexus 7 2013 |
|---|---|---|---|
| Accelerated graphics | Unstable | | |
| Brightness control | | | |
| LED control | | | |
| Orientation sensor | | | |
| Touchscreen pointer | OK | | |
| Multitouch | | | |
| WLAN | OK | | |
| Bluetooth | OK | | |
| Battery charging | OK | | |
| Battery monitoring | OK | | |
| Suspend/resume | | | |
| Shutdown/reboot | OK | | |
| Buttons | OK | | |
| Sound | | | |
| Front camera | | | |
| Rear camera | | | |
| MMC | | NA | |
| USB OTG | OK | | |

# 3. Lenovo Yoga Tablet 2 830/851

## 3.1. Aperçu

Points forts :

- Le SoC et le GPU sont fournis par Intel. Le chip WLAN/Bluetooth est fourni par Broadcom.
- La plate-forme (Bay Trail Atom Z3745) n'implémente par AMT.
- Il y a une version Android (830) et une version Windows (851). Le code source du noyau Linux de la version Android est disponible. La version Windows a un BIOS UEFI déverrouillable.
- Certains indices suggèrent que la prise micro/casque donne accès à une console série. D'après le code source, on l'activerait en ajoutant `switch_to_uart=1` à la ligne de commande. Voir `osc_4_blade2_L_result/linux/kernel/drivers/misc/uart_audiojack_sw.c`
- Il y a un port microSD.
- La batterie est de grande capacité, par rapport aux autres tablettes. C'est un avantage pour le cas où Linux ne saurait pas exploiter toutes les fonctions d'économie d'énergie.
- Le partitionnement est simple :

```
mmcblk0p1      100 MiB   EFI
mmcblk0p2      128 MiB   Microsoft reserved (empty)
mmcblk0p3    23351 MiB   Microsoft basic data
mmcblk0p4     6239 MiB   Windows recovery environment
mmcblk0boot1     4 MiB   Empty
```

```
mmcblk0boot0     4 MiB   Empty
mmcblk0rpmb      4 MiB   Replay Protected Memory Block (content unknown)
```

- En plus du stockage eMMC, il y a une flash SPI de 8 MiB SPI que l'on pourrait remplacer ou reprogrammer si nécessaire.

Faiblesses :

- Le bootloader de la version Android (830) est verrouillé. **fastboot boot** renvoie *"boot command stubbed on this platform"*.
- La publication du code source respecte la GPL *a minima*. Aucune information de versionnement n'est incluse, pas même les dates de modification des fichiers.
- La version Windows (851) est plus chère que la version Android : 249 EUR au lieu de 149 EUR. La différence correspond à 16 GiB de flash supplémentaires, un bouton Windows sur le côté droit, une LED d'activité à côté de l'écran, une finition noire, et une licence Windows.
- En tant que machine x86 avec un BIOS de PC, le modèle 851 est une plate-forme plus complexe qu'une simple tablette ARM. Il est difficile de savoir ce qui se passe sous le capot, notamment au sujet de la virtualisation et du SMM.
- Le fabricant est connu pour installer des logiciels dangereux sur ses appareils. En principe cela n'affecte que Windows.

Travaux similaires :

- *Fedlet: a Fedora Remix for Bay Trail tablets.* [https://www.happyassassin.net/fedlet-a-fedora-remix-for-bay-trail-tablets/] La version 20150810 boote GNOME avec `nomodeset`. Adresse MAC WiFi codée en dur, pas de Bluetooth, problèmes de `clocksource`.
- *Ubuntu (or other Linux) on the Asus Transformer Book T100.* [http://www.jfwhome.com/2014/03/07/perfect-ubuntu-or-other-linux-on-the-asus-transformer-book-t100/]

# 3.2. Fedora 22 sur le Lenovo Yoga Tablet 2 851

Mon 851F était livré avec le BIOS en version 02WT18WW (2015-07-24). Il est possible que des version antérieures n'autorisent pas la désactivation du *Secure Boot*.

Voici où j'en suis :

## 3.2.1. Préparation

- Copier le contenu de `Fedora-Live-Workstation-x86_64-22-3.iso` à la racine d'un disque USB FAT. On ne peut pas utiliser l'image ISO9660 (en lecture seule) car des modifications sont nécessaires.
- Copier `bootia32.efi` [https://github.com/jfwells/linux-asus-t100ta/raw/master/boot] vers `EFI/BOOT/`. Cette étape est nécessaire parce que le BIOS UEFI 32-bit ne peut pas booter le chargeur EFI x86_64 de Fedora. Il ne peut pas non plus booter le chargeur ISOLINUX de Fedora i386.
- Copier `EFI/BOOT/grub.cfg` vers `boot/grub/grub.cfg`. Sans cela `bootia32` ne le trouve pas et présente seulement une invite de commande GRUB.
- Éditer `boot/grub/grub.cfg` :

```
linux /isolinux/vmlinuz0 root=live:LABEL=FEDLIVE ro rd.live.image nomodeset
initrd /isolinux/initrd0.img
```

- Renommer la partition du disque USB pour que GRUB la reconnaisse :

```
# fatlabel /dev/sdX1 FEDLIVE
```

### 3.2.2. Installation

- Brancher le disque USB et un clavier USB à la tablette via un hub USB et un cordon USB OTG. Certains recommandent d'utiliser un hub alimenté.
- Éteindre, maintenir VolumeUP, allumer. On arrive dans le "Novo Menu".
- Choisir "BIOS Setup" et désactiver le *Secure Boot*.
- Rebooter à nouveau jusqu'au "Novo Menu".
- Choisir "Boot Menu", sélectionner "EFI USB Device".
- Laisser faire l'installateur Fedora. Monter `/dev/mmcblk0p1` sur `/boot/efi` et `/dev/mmcblk0p3` sur `/`. Préserver `/dev/mmcblk0p4` (outil de restauration Windows) à toutes fins utiles.
- Modifier GRUB une fois de plus :

```
# cp /run/initramfs/live/EFI/BOOT/bootia32.efi /boot/efi/EFI/Boot
# mkdir -p /boot/efi/boot/grub
# cp /boot/efi/EFI/fedora/grub.cfg /boot/efi/boot/grub/
```

Éditer `/boot/efi/boot/grub/grub.cfg` :

```
linux /boot/vmlinuz-4.0.4-301.fc22.x86_64 root=UUID=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxx
initrd /boot/initramfs-4.0.4-301.fc22.x86_64.img
```

Sans `clocksource=tsc`, le noyau bascule sur `refined-jiffies` et le système est anormalement lent. HPET est désactivé sur Bay Trail [http://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/commit/arch/x86/kernel/early-quirks.c?id=62187910b0fc7a75cfec9c30fda58ce2f39d689b].

- Rebooter sur le disque interne.

### 3.2.3. Améliorations et palliatifs

- Dans Gnome settings -> Power, désactiver la mise en veille automatique. Elle n'est pas encore fiable.
- Récupérer dans EFI les données de configuration du firmware WLAN, dont l'adresse MAC individuelle :

```
# cp /sys/firmware/efi/efivars/nvram-XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX /lib/firmwar
```

Il y a deux fichiers `nvram` dans ce répertoire. Le bon contient environ 3 Ko de paramètres en format texte.

- Créer `/etc/modprobe.d/brcmfmac.conf`. Ceci évite "*brcmf_escan_timeout: timer expired*" provoqué par la mise en veille de sdio.

```
install brcmfmac echo on > /sys/bus/platform/drivers/sdhci-acpi/INT33BB:00/power/contr
```

- Le chip WiFi semble configuré pour se mettre en veille très rapidement. Les connexions sortantes fonctionnent, mais les paquets entrants non attendus s'accumulent quelque part jusqu'à ce que le pilote `brcmfmac` se réveille. Il en résulte qu'on ne peut pas se connecter par **ssh**. J'ai trouvé trois palliatifs :

- Faire en sorte qu'il y ait suffisamment de trafic sortant pour occuper le pilote :

  ```
  yogatablet# ping wlanap &
  ```

- Ou réveiller la tablette avec des paquets brodcast. Pour une raison indéterminée ils réveillent toujours le pilote sans délai.

  ```
  wlanap# arping -I ath0 -b yogatablet
  ```

- Ou inonder la tablette avec sufisamment de paquets entrants pour empêcher le chip de passer en veille. La période maximale est d'environ 200 ms ; avec un paquet tous les 90 ms on tolèrera donc des pertes occasionnelles. Noter qu'une des autres techniques ci-dessus reste nécessaire pour embrayer le processus.

  ```
  wlanap# ping -i 0.090 yogatablet
  ```

- `/sys/bus/acpi/devices/80860F0A:00/BCM2E84:00/path = \_SB_.URT1.BTH0` révèle que la partie Bluetooth du BCM43241 est connectée au premier port d'une interface série dont l'ID ACPI est 80860F0A.

  On configure donc `CONFIG_SERIAL_8250_DW=m` pour compiler `8250_dw.ko` (non fourni avec le noyau Fedora).

  Démarrer **hciattach** par exemple avec ce `/etc/systemd/system/hciattach.service` :

  ```
  [Unit]
  Description=Broadcom BCM43241 in Yoga Tablet 2
  After=syslog.target
  Before=bluetooth.service

  [Service]
  Type=simple
  ExecStartPre=/usr/bin/bash -c '/usr/bin/echo on > /sys/class/tty/ttyS1/power/control'
  # bcm43xx_init is required after power up, but will fail after warm reboot.
  # Don't know where to find the bdaddr. Use the WLAN MAC instead.
  ExecStart=/usr/bin/bash -c 'MAC=$(sed -e "s/macaddr=\(..:..:..:..:..:..\)/\1/;t;d" /sy

  [Install]
  WantedBy=multi-user.target
  ```

  ```
  yogatablet# systemctl enable hciattach.service
  yogatablet# systemctl start hciattach.service
  ```

- À ce stage **Xorg** s'appuie sur le framebuffer EFI, qui ne prend pas en charge la rotation par **xrandr**. Pour pivoter l'écran en mode horizontal, créer `/etc/X11/xorg.conf.d/90-RotateCW.conf` :

  ```
  Section "Device"
      Identifier  "Card0"
      Driver      "fbdev"
      Option      "Rotate" "CW"
  EndSection

  Section "InputClass"
      Identifier  "RotateCW"
      MatchIsTouchscreen "True"
      Option      "TransformationMatrix" "0 1 0 -1 0 1 0 0 1"
  ```

```
EndSection
```

Malheureusement **gnome** se plaint alors de l'absence de support RANDR et crashe. Je n'avais de toute façon pas l'intention de l'utiliser. Les gestionnaires de fenêtrage classiques fonctionnent sans problème en mode horizontal.

### 3.2.4. Problèmes identifiés

- On peut activer le GPU en remplaçant `nomodeset` par `i915.modeset=1` dans la ligne de commande du noyau. La plupart du temps il se bloque pendant for 60 secondes dans `intel_crtc_wait_for_pending_flips()` au démarrage. Ensuite **xrandr -rotate** fonctionne, et **glxgears** tourne à 60 im/s en plein écran. Malheureusement lorsque le GPU est activé la machine plante généralement après quelques minutes.
- La mise en veille ne fonctionne pas.

# 4. Fedora sur le ASUS / Google Nexus 7 2013 (flo)

## 4.1. Aperçu

Points forts :

- Les appareils Nexus sont recommandés pour les développeurs, donc plutôt ouverts.
- Le bootloader est déverrouillable. Les outils d'[AOSP] permettent de compiler et de booter facilement des noyaux.
- Le bootloader est re-verrouillable. Ceci apporte un peu de sécurité contre les attaques physiques.
- Le code source est géré dans le cadre d'AOSP.
- La prise micro/casque donne accès à une console série.
- Il y a vraisemblablement un port JTAG sur le circuit imprimé, caché sous un code-barres.
- L'appareil peut être alimenté inductivement pendant que le port USB est utilisé en mode OTG pour le développement.

Faiblesses :

- Le CPU, le GPU et le chip WLAN sont fournis par Qualcomm.
- AOSP intègre de nombreux composants propriétaires ; pas seulement des firmwares, mais aussi des binaires ARM. Il semble que les pilotes graphiques de bas niveau tournent en userspace plutôt que dans le noyau, vraisemblablement pour les faire échapper à la GPL.
- Le Nexus 7 (2013) n'est toujours pas supporté par la branche générique de Linux plus de deux ans après sa sortie.

Travaux similaires :

- *Running mainline kernels on a 2013 N7(flo)* [https://plus.google.com/111527804358069266888/posts/S4ajVewveSR]
- *Kernel development – vanilla all the way* [http://delta.zauner.one/168-nexus-7-kernel/]

## 4.2. Procédure

Voici où j'en suis :

- Fabriquer un cordon pour la console série. Voir [CONSOLEJACK]. Le bootloader détecte le cordon et configure la prise micro/casque en mode série.
- Compiler Linux 4.2.0 avec gcc 4.6. Ne PAS utiliser les toolchains 4.8 et 4.9 d'AOSP.

```
$ export ARCH="arm"
$ export CROSS_COMPILE="arm-eabi-"
$ make flo_defconfig
```

- Compiler `fixup.bin` et `qcom-apq8064-nexus7.dtb` comme indiqué ici [https://plus.google.com/111524780435806926688/posts/S4ajVewveSR] ::

```
arm-eabi-gcc -c fixup.S -o fixup.o
arm-eabi-objcopy -O binary fixup.o fixup.bin
cat fixup.bin linux-4.2.0/arch/arm/boot/zImage linux-4.2.0/arch/arm/boot/dts/qcom-apq8
```

- Éteindre, maintenir VolumeDown, allumer. On arrive en mode FASTBOOT.
- Booter jusqu'à un shell à l'aide d'une image recovery quelconque contenant **busybox** :

```
fastboot boot -n 2048 --base 0x80200000 --ramdisk_offset 0x02000000  \
   -c "console=ttyHSL0,115200,n8 user_debug=31 msm_rtb.filter=0x3F ehci-hcd.park=3 vma
   fixup-zImage-dtb recovery/initrd.img
```

À ce stade j'essaie de faire fonctionner le pilote `msm` du noyau Linux générique (`drivers/gpu/drm/msm/`). Ceci implique d'écrire un fichier DTS et de configurer :

```
CONFIG_DRM_MSM=y
CONFIG_CMA=y
CONFIG_DMA_CMA=y
```

# 5. Bilan

Les smartphones et tablettes ont longtemps été des plate-formes propriétaires avec des écosystèmes logiciels très contrôlés. J'ai commencé ce projet en considérant qu'Android avait mis fin à ce status quo il y a sept ans, pensant qu'il serait facile d'installer des distributions Linux standard au moins sur les appareils Nexus. Bizarrement, il s'est avéré que la façon la voie la plus simple pour installer Linux sur un appareil mobile consiste à choisis une tablette qui est en fait un PC x86. Peut-être tous les développeurs arm-linux ont-ils été embauchés par les fabricants de terminaux ? Ceci expliquerait que le développement de la branche générique soit ralenti.

Sur x86, les mécanismes *plug-and-play* de UEFI, ACPI et PCI facilitent les choses. Sur ARM, le *plug-and-play* n'a été introduit que vers 2009. À terme l'infrastructure Open Firmware/device tree permettra à un noyau unique de de booter sur plusieurs plate-formes ARM. Le Nexus 7, comme beaucoup d'appareils Android, utilise encore des structures `platform_data` codées en dur. La plupart des appareils Nexus recents ont une structure `dtb` accollée au noyau ; voir par exemple `tegra124-flounder.dts` [https://android.googlesource.com/kernel/tegra/+/android-tegra-flounder-3.10-lollipop-release/arch/arm/boot/dts/tegra124-flounder.dts] et `tegra124-flounder-generic.dtsi` [https://android.googlesource.com/kernel/tegra/+/android-tegra-flounder-3.10-lollipop-release/arch/arm/boot/dts/tegra124-flounder-generic.dtsi] (Nexus 9). Cela ne me dérange pas d'écrire des fichiers DTS moi-même ; malheureusement, comme le(s) principal(aux) fournisseur(s) de SoCs ARM mobiles ne publie pas la documentation technique de ses produits, cela implique un travail fastidieux pour identifier les adresses d'IO, les numéros d'interruptions et les affectations de broches GPIO.

Une autre difficulté rencontrée dans ce projet, évidemment, est qu'il n'y a pas tant d'appareils Android avec un bootloader déverrouillable. Il est surprenant que les fabricants s'adonnent encore impunément

à de telles pratiques, alors qu'il est maintenant admis que les PC ne doivent pas être liés artificiellement à un système d'exploitation particulier.

# Bibliographie

[CONSOLEJACK] *Un adaptateur USB amélioré pour la console série sur jack audio des appareils Google Nexus* . http://www.pabr.org/consolejack/consolejack.fr.html .

[AOSP] *Android Open-Source Project*. https://source.android.com/.