



# Introduction To Google Chromebooks and Chromeboxes

Trever Nightingale, NERSC/LBL, 10/10/2013

# What is a Chromebook?

It is a special device\* you must buy to get all the features I will discuss.

It runs Google's Chrome web browser.

You can NOT install software on it.

\* Chromebook, Chromebox, tablets rumored

# **Bill Richardson, Google Chrome OS firmware engineer:**

"I think of the Chromebook as an internet appliance, or a web browser with a keyboard. It **looks** like a laptop computer, but it really isn't... All the Chromebook really does is browse the web."

**Me:**

"What's a web browser? What is browsing the web?"

- Is using Google Docs offline "browsing the web"? What about Chrome Remote Desktop?
- Is it a browser if you are playing Unreal Tournament in it?  
(See Fluent 2013 "Javascript at 18")
- Have you seen what HTML5 can do?  
V8? ASM.JS? Dart?

*The "web" evolution/revolution continues...*

# What are the limitations of this www "Internet Appliance"?

## Chromebooks today:

- Have an ssh client
- Have a serial port terminal (Beagle Term)
- GUI for Google Drive and local (noncloud) files and folders
- VPN support (OpenVPN, L2TP over IPsec with PSK or cert) → not Lab's
- USB Audio
- USB Video (webcams) [note: will NOT play DVD's / Blue Ray]
- OpenPGP for web mail (Mailvelope)
- "Web Apps" (Text, Keep, more on the way)
- Google Apps, Evernote, Netflix, etc.
- Gaming in browsers
- What else?

# Supported external storage devices

You can open and save files on external standard mass storage USB devices connected to your Chromebook, if they use the following types of filesystems.

- Ext2, Ext3, Ext4
- FAT
- HFS+ (read-only on journaled HFS+)
- ISO9660 (read-only)
- NTFS
- UDF (read-only)

# Supported file types

- Microsoft Office files (read-only)  
.doc, .docx, .ppt, .pptx, .xls, .xlsx
- Media [note: no support for .wmv]  
.3gp, .avi, .mov, .mp4, .m4v, .m4a, .mp3, .mkv, .ogv, .ogm, .ogg, .oga, .webm, .wav
- Images [note: raw DSLR formats have G+ support]  
.bmp, .gif, .jpg, .jpeg, .png, .webp
- Compressed files  
.zip, .rar, .tar, .tar.gz (.tgz), .tar.bz2 (.tbz2)
- Other  
.txt, .pdf

# Supported external peripherals

Here are the types of peripherals you can use with your Chromebook.

- USB keyboards (Windows and Mac)
- USB mice with the following features: left button, right button, scrollwheel
- USB hubs
- Bluetooth mice and keyboards
- Monitors with DisplayPort, DVI, HDMI, or VGA connections
- USB, DisplayPort, and HDMI audio devices
- Headset with a 3.5 mm jack
- Webcam with a USB cable
- MP3 player or mobile phone with a USB cable (can be used for charging)

**Most newer models: External monitor and portrait mode make all day office work possible**



# Yes, this kind of thing works

## USB DAC



# So what is a Chromebook?

- Nothing but the web?

No, a bit more than that. Eg. openssh client.

- And let's not forget: the web and browsers are able to do *a lot* more each passing day.
- But it is true: no skype, no java, can't upload to Google Play... etc.

*Know your use cases!*

**"How is this different than running Chrome on a Mac or PC?"**

**It's the exact same thing. Except, I can't add needed software to it.**

**Very limiting.**

**Why buy a Chromebook? What's the point of these things?"**

**Appliance is the point**

**I think of:**

**Chromebooks as the NetApp's of  
desktops/laptops**

**( ...of personal productivity computers )**

# Appliance means:

- Zero maintenance
- Much more secure
- Fast
- Easy to use, hard to misconfigure / break
- Share beautifully
- Less expensive for comparable hardware\*

\* TCO/Initial cost/Google's monetization model where OS is free,  
lower costs are thus a product of many factors, not just appliance model

## Appliance model has real value

# **Security: persistent malware proof**

Appliance means software pre-defined, initially installed, and regularly updated by Google.

Critically: Google signs all of the onboard executable bits and the Chromebook verifies.

## **Verified boot:**

Chromebooks will detect and repair if OS has been tampered with (note: also detects bit rot).

# Verified Boot Starts In The Hardware:

- This is why you must buy a chromebook
- Custom firmware (coreboot and uboot)
- Google's 8192-bit public key burned at factory into read-only firmware
- PKI chain of trust verifies OS

Note: Linux verity fs developed by Google and Netflix.  
Open Source.



# Verified Boot

- Part of the BIOS flash is read-only
- The read-only BIOS runs first
- The read-only BIOS verifies the read-write BIOS, then executes it
- The read-write BIOS verifies the kernel, then executes it
- The kernel verifies the rootfs as each block comes off the drive.
- If anything fails, it reboots into Recovery mode (read-only BIOS again).

# Extremely Fast Boot (~ always on):

Appliance means hardware pre-defined.

- Firmware always knows its hardware (no hardware probing)
- No multistage boot loader(s)
- Fastest path to loading and executing the Linux kernel

My Chromebox boots faster than my flat screen monitor can sync a signal.

# Zero configuration BIOS:

Appliance means pre-defined hardware, so firmware doesn't need to have knobs.

You never deal with the BIOS unless you enter recovery mode (probably never).

Recovery mode just says: "Press the space key." Dad and Mom can do this, IF needed.

Even faster and simpler than today's Macs.

# Zero maintenance:

Appliance means all software maintenance taken care of automatically and no user or administrator can change or break this.

On disc layout pre-defined for:

- self healing-- failover software partitions
- separates minimal user data from all else

Firmware and OS software take care of updates automatically. Non disruptive- user reboots.

**In case you missed that:**

No more weekends upgrading your computer.

**In case you missed that:**

You never do anything. Ever. It just works.

# **Much more secure user data:**

Appliance means automatic, timely software updates in case of identified vulnerabilities.

User data partitions always encrypted.

Users/Owners can't break any of this.

Design encourages no backup needs and keeping data in the cloud.

# Lost Chromebook (I did this):

Appliance means no problem (if locked...).

Units are actually *disposable*. (Funny videos...)

Replace, pick up where you left off.

## More on security:

- Executable bits are on read only partitions
- Linux OS is hardened in various ways (see online presentations)
- Chrome browser itself is considered a better design re security
- Ongoing security work being done

Chromebooks were designed from the ground up to provide much greater security. Most secure off the shelf computer you can buy? AFAICT



# **Security take away:**

Do your online banking on a chromebook

# Sharing Chromebooks

**Simply put:**

I can easily and quickly use your chromebook or loan you mine, and neither of us has any worry or hassles.

# Sharing Chromebooks:

- Access or changing any other user's data is impossible. Worst case: owner deletes
- There's no administrator. The first person to use a chromebook can limit who can use the device, or just leave it usable to others.
- No one can misconfigure / change it.
- Maintenance happens automatically as usual no matter who is using it.

Note: Enterprise enrollment provides various configuration and account options.

# Sharing Chromebooks continued:

- User's just login with Google credentials. Or Guest Mode. No account management.

## Sharing summary:

Share away! Absolutely zero worries.

Great for kids, kiosks and my parents...

# **Chromebooks as Geek Appliance:**

## **Dev Mode**

# Geek Appliance / Dev Mode:

By flicking a switch, you can put Chromebooks into a mode where:

- They warn you they are in dev mode
- You can boot from USB or the internal drive
- They will boot properly formatted linux kernels or wrapped executables that aren't actually signed by Google, though it is possible to also re-enable verification to detect OS tampering
- You get access to a local shell

# Geek Appliance Benefits:

You can use the underlying minimal Linux userland and/or add your own, while keeping Chrome and its OS updates and verified boot.

`crouton, dev_install`

See my write up on the official Chrome OS Wiki

<https://sites.google.com/site/chromeoswikisite/home/what-s-new-in-dev-and-beta/shell-access-with-verified-boot>

**The Geek Appliance Mode means a chromebook is an intriguing, maintained Linux laptop you can buy off the shelf.**



# **Not advised, but some suffering geeks just use the hardware:**

You can dual boot into Chubuntu, and later re-wipe to get your Chrome OS appliance back.

Why this hardware? Hardware support:  
3.9 and above Linux kernels fully support Chromebook hardware.

Pixel has built in SeaBIOS option that allows to boot just about anything (haven't tried myself).

# Note:

- The custom firmware can be modified by hobbyists (non-trivial)
- Google's firmware engineers have proposed making it easier to burn in one's own key... self signed images/distros- huge potential?
- Hacker work is ongoing
- Coreos: based on Chrome OS

# Speaking of Geek Mode:

You can always see the underlying Linux, no Dev Mode necessary:

`chrome://system`

dmesg

Expand... Collapse...

```
[ 0.000000] Booting Linux on physical CPU 0
[ 0.000000] Initializing cgroup subsys cpu
[ 0.000000] Linux version 3.4.0 (chrome-bot@build47-m2) (gcc version 4.7.x-google 20130114 (prerelease) (4.7.2_cos_gg_c8f69e0) ) #1 SMP Thu Se
[ 0.000000] CPU: ARMV7 Processor [410fc0f4] revision 4 (ARMV7), cr=10c5387d
[ 0.000000] CPU: PIPT / VIPT nonaliasing data cache, PIPT instruction cache
[ 0.000000] Machine: SAMSUNG EXYNOS5 (Flattened Device Tree), model: Google Snow
[ 0.000000] Ramoops: 41f00000 - 41ffffff
[ 0.000000] bitfix_reserve: Detected firmware that needs bitfix
[ 0.000000] Memory policy: ECC disabled, Data cache writealloc
[ 0.000000] CPU EXYNOS5250 (id 0x43520010)
[ 0.000000] exynos5_init_clocks: initializing clocks
[ 0.000000] S3C24XX Clocks, Copyright 2004 Simtec Electronics
[ 0.000000] exynos5_setup_clocks: registering clocks
[ 0.000000] exynos5_setup_clocks: xtal is 24000000
[ 0.000000] EXYNOS5: PLL settings, A=1700000000, B=800000000, C=333000000
[ 0.000000] M=1600000000, E=960000000 V=300000000 G=533000000
[ 0.000000] EXYNOS5: ARMCLK=1700000000, CDREX=800000000, ACLK400G3D=533000000
[ 0.000000] ACLK333=333000000, ACLK266=266666666, ACLK200=200000000
[ 0.000000] ACLK166=166500000, ACLK66=66666666
[ 0.000000] sclk_fimd: source is mout_mpll_user (6), rate is 266666666
[ 0.000000] aclk_266_gsc1: source is aclk_266 (1), rate is 266666666
[ 0.000000] sclk_g3d: source is mout_mpll_user (0), rate is 800000000
[ 0.000000] sclk_sata: source is mout_mpll_user (0), rate is 666666666
[ 0.000000] sclk_gsc1_wrap: source is ext_xtal (0), rate is 24000000
[ 0.000000] sclk_gsc1_wrap: source is ext_xtal (0), rate is 24000000
[ 0.000000] sclk_cam0: source is ext_xtal (0), rate is 24000000
[ 0.000000] sclk_cam1: source is ext_xtal (0), rate is 24000000
[ 0.000000] sclk_usbdrd30: source is mout_mpll (0), rate is 666666666
[ 0.000000] On node 0 totalpages: 524032
[ 0.000000] Normal zone: 3568 pages used for memmap
[ 0.000000] Normal zone: 0 pages reserved
[ 0.000000] Normal zone: 452880 pages, LIFO batch:31
[ 0.000000] HighMem zone: 528 pages used for memmap
[ 0.000000] HighMem zone: 67056 pages, LIFO batch:15
[ 0.000000] PERCPU: Embedded 9 pages/cpu @81852000 s12480 r8192 d16192 u36864
[ 0.000000] pcpu-alloc: s12480 r8192 d16192 u36864 alloc=9*4096
[ 0.000000] pcpu-alloc: [0] 0 [0] 1
[ 0.000000] Built 1 zonelists in Zone order, mobility grouping on. Total pages: 519936
[ 0.000000] Kernel command line: cros_secure console= console= loglevel=7 init=/sbin/init cros_secure oops=panic panic=-1 root=/dev/dm-0 root=
[ 0.000000] device-mapper: init: will configure 1 devices
[ 0.000000] PID hash table entries: 4096 (order: 2, 16384 bytes)
[ 0.000000] Dentry cache hash table entries: 262144 (order: 8, 1048576 bytes)
[ 0.000000] Inode-cache hash table entries: 131072 (order: 7, 524288 bytes)
```

This presentation made on a chromebook

Never saved anything (files metaphor gone)

It is a new computing paradigm?

# Chrome Core Principles:

- Speed
- Security
- Stability
- Simplicity

**THE END**