Hacking The Google TV

Presented by: Amir “zenofex” Etemadieh, CJ “cj_000” Heres, Tom “tdweng” Dwenger, and Dan “bliss” Rosenberg

http://gtv hacker.com/pres/dc20.ppt
GTVHacker is a group of 6 hackers with individual skill sets who work together to unlock Google TV devices.

Our primary goal is to bypass hardware and software restrictions to allow for unsigned kernels to be loaded and used.

To date the team has released multiple methods for unlocking Google TV devices.

GTVHacker team won $500 bounty for being the first to root the Google TV.
Team Members

The GTVHacker team officially consists of 6 members:

- AgentHH – First human outfitted with metal legs.
- cj_000 - Destroyer of words
- Gynophage – German rockstar reverse engineer
- [mbm] - known for founding the Open-WRT project and tossing 251 children down a well
- Tdweng – software developer turned super hero.
- Zenofex – ❌❌❌❌❌❌❌❌❌❌

With special guest:
- Bliss - a vulnerability researcher who takes sick pleasure in exploiting anything with a CPU. He once punched an Android in the face.
Google TV is a platform that bridges the gap between your TV and an Android device.

Platform creates an overlay on television stream and also contains an IR transmitter to transmit to media center devices (cable box, TV, sound system).

Device was originally released without the Android Market available but was eventually updated to include it.

Platform receives Over-the-Air updates automatically from OEM manufacturer.

Platform contains forked version of Chrome with all plug-ins and extensions other than Flash disabled.
Android vs. Google TV

Although Google TV runs Android there are differences:

- The device has a Chrome browser out of the box which provides a fairly reliable and safe browsing experience.
- The Gen 1 Google TV platform is currently the only x86 set of Android devices.
- Although the platform does have the Android market, the amount of actual applications available is far below that of the actual market.
- Due to the fact that some Android applications include native code, some applications are not able to run on the x86 chip-set.
- Unlike most Android devices, GTV devices are USB hosts requiring ADB to be used over the network and ADB is restricted to one white-listed IP.
x86 vs ARM?

- Most commonly deployed boxes are x86
- Newest Google TV Devices are ARM based
- Devices by Sony, LG and Vizio (Availability is still limited)
- More on the ARM devices a bit later!

http://gtv hacker.com/pres/dc20.ppt
From the initial release of the platform, the Google TV has been in a constant battle with the content providers.

Content providers believed giving Google access to television programming advertising streams would strengthen Google's position in web advertising, as well as convince users to drop services like cable.

Websites enforced checks by verifying the browser User-Agent as well as the Flash version string.
Platform: x86 Android

- There are no other mainstream Android x86 devices.
- Architecture differences make for a crippled marketplace.
- Code compiled for device can usually be compiled without the need for compiler toolchain.
Current generation of Google TV devices use an Intel CE41xx CPU.
45nm Atom core 1.2 Ghz with System-on-Chip (SoC).
“On-die” security processor to handle DRM.

- Revue – CE4100
- Sony TV / Blu-Ray – CE4150
The bootloader for the CE41xx devices is known as the “Intel CEFDK” (Consumer Electronics Firmware Development kit).

Bootloader is signed and signature is verified by security processor, beginning “chain of trust”.

Intel supplies a stage 1 and stage 2 boot-loader in the SDK.

Logitech uses both stages of CEFDK in its device.

Sony uses Intel's stage 1 and it's own proprietary “NBL” for stage 2.
GTV platform utilizes a “Chain of Trust” boot

1) SoC decrypts and verifies signature of stage 1 CEFDK
2) Stage 1 CEFDK boots, checks signature, and decrypts Stage 2
3) Stage 2 boots and checks signature on Kernel
4) Kernel takes over
5) (Sony) Kernel SHA1 hashchecks init
6) (Sony) Init RSA verifies init.rc / init.(eagle/asura).rc
Kernel Security

- Kernel requires modules to be properly signed before being inserted.
- All partitions except /data & /cache are marked as RO by the kernel.
- ADB shell only allows RW access to folders with “shell” permissions.
- Functions like ptrace are left out of the kernel.
- Access /dev/mem is restricted.
- Kernel is patched from all known public Android vulnerabilities.
Current Devices

Google TV™

http://gtvhaeker.com/pres/dc20.ppt
Logitech Revue

- Released October 2010
- Full sized keyboard with built in touchpad
- Originally priced at $249 later reduced to $199 and finally $99
- Discontinued but still favoured by a majority of GTV users
Logitech Revue Motherboard

- UART1 - Console
- J3 - PICKIT2
- SW1 – Unused switch
- J20 – I2C
- J69 – USB
- SATA1 – SATA Header
- J24 – Unknown
- J13 – Power for SATA
- XDP1 – Intel XDP Debug Header

Image via ifixit.com
Revue: Recovery

Recovery mode is an “Android 2e recovery” which is standard on many Android devices.

- Reboot
- Apply Update from USB (update.zip)
- Wipe data/factory reset
- Wipe cache partition

All update files provided are RSA verified before the box attempts installation.
UART On The Revue

- First root on the Google TV Platform.
- Required a virgin Revue.
- Still works on newly purchased Revues.
- Soldering to four pads on the Revue and booting into recovery mode.
- Method allowed for Read/Write access to File System.
UART On The Revue

- Created a manual update process that mirrored Google's but did not perform any of the signature checks.

- Continued to release modified updates which included an ADB running as root as well as our first attempt at a content provider bypass.

http://gtv hacker.com/pres/dc20.ppt
First “Content Provider Bypass”

Bypassing Hulu/CBS/NBC/ABC's browser/flash checks was relatively easy and could be done quickly with a hex editor and RW /system access. All that was required was a simple change from:

00969F5269 6E 3A 00 47 54 56 20 31 30 2C 31 30 37 2C 31 30 37 2C
00969F6331 39 31 00 50 6C 75 67 49 6E 00 35 2E 31 00 25 32

To:

00969F5269 6E 3A 00 41 54 56 20 31 30 2C 31 30 38 2C 31 30 38 2C
00969F6331 35 31 00 50 6C 75 67 49 6E 00 35 2E 31 00 25 32

Changing one letter in the flash version string as well as changing the browser user agent (which can be done directly from the box in Chrome's settings) will allow a user to watch normally restricted content.
Honeycomb Suprises: Message from Logitech?

- Logitech removed the recovery menu and replaced it with a message to the GTVHacker team.
- Removed functionality to install manual updates therefore removing a user's ability to recover other than via the automatic process of erasing /cache and /data.
- The message was encoded in a ROT13 cypher.
- Each of the current GTVHacker team members' names were listed as no longer functioning recovery menu items.

http://gtv hacker.com/pres/dc20.ppt
Honeycomb Suprises: Message from Logitech?

“A @gtvhackers congratulations if your reading this please post a note on your forum to let me know ;)”

http://gtvhacker.com/pres/dc20.ppt
Flash Sabotage: Revue

Getting a secret message from Logitech was awesome. Having them remove the recovery menu functionality was not.

So we needed a way to play with the update functionality of the box...

The OTA updater writes to /cache/recovery/command, which uses the following syntax:

--update_package=CACHE:/somefile.zip

Now if only we had a way to write to cache...
Flash Sabotage: Revue

- /cache and /chrome are EXT3 partitions stored on NAND flash.

- Luckily, that flash is connected to the Revue via a USB Controller.

- It's a flash drive!

- We can tap the data lines and stick our own flash drive in line.

http://gtv hacker.com/pres/dc20.ppt
Revue root kernel exploit

To be added
Revue Module Signing Exploit

Revue RSA kernel module signing bypass

To be added
Sony Devices (Gen1)

Blu-Ray Player (NSZ-GT1)

Television 24” - 46” (NSX-#GT1)

GTV hardware is nearly identical, other than the obvious differences.
NSZ-GT1: Motherboard

- Bulkier than the Revue.
- Built like a Sony.
- Populated debug pads!
- Contains a faster processor – CE4150 @ ~1.7GHz.
SATA Sabotage: Sony

- Internal SSD via SATA
- GBDriver RS2 AES encrypts all data on NAND flash.
- ATA Password
- Sony stored all data on SSD, except bootloader and kernel.
- Risky procedure. Small points.
- Able to “redirect” SATA bus to our own device, which we had RW access to externally.
- Used this to downgrade to old SW versions, to look for flaws.
Sony GTV: Recovery

- Far more interesting than that of the Revue.
- Like the Revue, has a similar “Update from USB” feature.
- Nearly entire backend is done through a series of scripts. Not standard Android, so no debug log is left behind. Though not impossible thanks to the UART.
- Sony updates are RC4 encrypted.

http://gtvhacker.com/pres/dc20.ppt
Can you spot the problem here?

```
ls /tmp/mnt/diskb1/package_list_* .zip | head -1 | grep "package_list_"

/bin/sony/check_version.sh $1
```
Sony Google TV
Command Execution Through Recovery

http://gtvhacker.com/pres/dc20.ppt
The exploit was simple, a package with a command:

```
package_list_;cd tmp; cd mnt; cd diskb1; sh t.sh; .zip

/package-updater.sh -l 0 -p /tmp/mnt/diskb1/package_list_;cd /tmp;cd /mnt;cd /diskb1;sh t.sh;.zip
```

The command above involved a t.sh bash script (to meet filename size limitations) which spawned a shell over UART and telnetd.

From there we proceeded to dump the recovery file system.
Unfortunately this exploit was patched in the 7/2011 update.

“It's not exactly what we'd call a easy jailbreak, seeing as how it requires a soldering iron, a NAND format procedure, and a Logitech Revue that's never even been powered on, but it looks like it is possible to root a Google TV box after all.” - engadget.com

That was said about 4 large pads for the Revue.

Needless to say, this was not a viable option for the public.
Sony Google TV
UART

- Active UART line (output only)
- After initial hack - achieved root console in Linux.
- Memory dump shows existence of "NBL" - an extra step after Intel's initial bootloader.
- Mashing escape over UART at start-up brings us to a "Password:"
- Password found after reversing NBL areas of memory:
  
  console_ON

http://gtv hacker.com/pres/dc20.ppt
Sony Google TV
UART / Bootloader

- NBL options included loading files into memory, and executing from internal flash or network via TFTP.

- Insecure booting features were disabled on production units.

- NBL Utilized signature and hash checks similar to the normal start-up mode.

Remember that exploitable recovery version?
Successful patch applied!
SEC FM: SEC ready
SEC FM: firmware module sent to SEC for authentication and load
Successful firmware download!
SEC FM: firmware version valid: 2.1.0.6.

Verify stage2 PASS
Intel(R) Consumer Electronics Firmware Development Kit (Intel(R) CEFDK)
Copyright (C) 1999-2010 Intel Corporation. All rights reserved.
Build Time (06/11/10 16:00:37),
Loading BIOS MicroCode at 0x800000
SATA 0: BTSSD01 - 8G
SATA 1: SONYDDP-410 - 0G

Password:
NBL BTV-EC 5.7.1 (base: 5.0-BTV_20100707) <built 11:14:57, 07/30/10 JST>
Machine: EAGLE (i386/sdeadville/btv)
RAM: 0x00100000-0x02000000 available
Hda: BTSSD01

(C/H/S = 30728/16/32, Total 15728640 sectors)
NBL> boot -f net=http://vmlinux_recovery.trf -c "root=/dev/ram0 console=ttyS0,115200 mem=exactmap memmap=1M$0 memmap=199M$1M" -initrd http://vmlinux_recovery.trf

i8254x: Link is up, 100Mbp Full Duplex
dev.net.c.net_getparams: enable bootp because IP=0
bootp: 'myip' is 192.168.1.121
'serverip' is 192.168.1.148
bootp: mask: 255.255.255.0
net_open: client addr: 192.168.1.121
net_open: subnet mask: 255.255.255.0
net_open: net gateway: 192.168.1.2
net_open: root addr: 192.168.1.148
net_open: server addr: 192.168.1.148
net_open: server path: /net_open: file name: vmlinux_recovery.trf
TRF file is loaded: start = 0x00100000, length = 0x00396086
i8254x: Link is up, 100Mbp Full Duplex
Booting via TFTP allowed us to set kernel args.

```
boot -f net:tftp:/vmlinux_recovery.trf -c "root=/dev/ram0 console=ttys0,115200"
   -initrd net:tftp:/initrd.trf
```

Booting via TFTP however kept the internal SSD ATA locked.

The good news was that when that recovery booted to a locked ATA, the box dropped us into a console!
Exploitable Recovery:

- System boot binaries stored on flash at /dev/Glob_Spectraa2
- ATA was locked, flash was not! Drivers just needed to be loaded.
- Replaced new recovery on flash with the old, exploitable version.
- Now we had an exploitable recovery!
- Wait for the rumored 3.2 release in late September to release exploit
- Google and Sony were slow – it was December.
Come the 3.2 release in December, we did not want to let on about the bootloader password being found. So, two weeks of intense bug finding was started.

We found a few bugs, but not what we needed for privileged code execution.

However, we got to really, really know the update process...
Sony Google TV
Downgrade via USB (nodev)

- Recovery mounts USB to /tmp/mnt/diskb1
- Looks for package_list_*.zip
- Passes this to package_updater.sh
- package_updater.sh then copies the file to /cache
- package_updater then unzips build.prop, and displays to the user

If the update is accepted, it's copied again to /cache

I'm sure they checked to see if there was a destination file...
Sony Google TV
Downgrade via USB (nodev)

The Sony recovery mounted ext2/3 partitions with no mount parameters meaning a block device on the USB could allow us to write to a device node as root.

- **USB1**
  - contains update file which fools the updater to think a properly formatted update is on the drive.

- **USB2**
  - contains update file which is a file system node point to /dev/Glob_Spectraa2

  Recovery moves the FS node to /cache/package_list_*_.zip and then errors leaving the file in place.

- **USB1**
  - is used again now that the FS node is in place, we restart update and perform part 2 of attack.

- **USB3**
  - contains update file which contains files to overwrite .trfs on /dev/Glob_Spectraa2

  The recovery version is now downgraded to the LCE exploitable version.

Recovery confirms that an update is inserted and asks the user if they'd like to continue.

Recovery confirms that an update is inserted and asks the user if they'd like to continue.

http://gtv hacker.com/pres/dc20.ppt
Assuming the downgrade went correctly, use LCE recovery exploit.

Exploit:
- Re-partitions internal SSD
- Copies /boot to a new partition.
- Edits initial /boot to include kexec files.
- Hijacks initial boot process to call kexec.
"kexec (kernel execution) is a mechanism of the Linux kernel that allows "live" booting of a new kernel "over" the currently running kernel. " ~ Wikipedia

- Kexec is normally built into the kernel, so we opted to build it as a kernel module.

- Kexec allows us to boot the system, have it kick over after in less than 1 second, and load our unsigned kernel.

But what about that init hash, and those RSA signatures?
Sony Google TV
Unsigned Kernels

- Chain of Trust needed to be broken
- kexec had to be called before the platform's security firmware was loaded.
- Where do we attack?
  - /bin/e2fsck
- / is mounted from sda1, on the SSD, that we can now write to
/bin/e2fsck was replaced with a script which:

- Mounted /system
- insmod our kexec modules
- kexec to load our new kernel

Our new kernel, apart from featuring no hash on init, had a few other tweaks:

- no initd hash
- no signed init.rc
- no signed init.(eagle/asura).rc
- modified init.rc
- modified init.(eagle/asura).rc
- modified default.prop
  - ro.secure=0
  - ro.debuggable=1
But wait – there's more!

Our update script pulled Chrome's Flash player and mutated the Flash plug-in string randomly per each install.

Since each box has a unique ID, content providers will have a harder time blocking streaming content for Google TV users.
Google TV
“Future” / ARM Devices

In the last few months we've seen a release of the second generation of Google TV devices, all of which are ARM:

Sony NSZ-GS7 – Network Streamer
Sony NSZ-GP9 – Blu-Ray Player *unreleased*

Vizio VAP430 (CoStar) – Network Streamer *unreleased*
Vizio VBR430 – Blu-Ray Player *unreleased*
Vizio R3D*0VS (42/47/55/65) – Google TV *unreleased*

LG 47/55G2 (LMG620) – Google TV
The Sony ARM devices feature a Marvell 88DE3100 SoC, which has a 1.2GHZ Dual Core Processor.

The Blu-Ray variant should be close to identical specs wise, but with a Blu-ray drive, and a BD playback app.

Sony has yet to branch off into TV integration, as they may have jumped the gun the first time around.
The Vizio ARM devices, like the Sony's feature a Marvell 88DE3100 SoC, which has a 1.2GHz Dual-Core processor.

Again, the Blu-Ray variant should be close to identical specs wise, but with a Blu-ray drive, and a BD playback app.

Multiple devices, a streamer, BD player, and integrated TV.

Hey, you – guy on stage. Is the streamer out yet?
Google TV
LG - ARM Devices

- LG Google TV's are a bit more mysterious.
- 47” & 55” (G2 / LMG620)
- Mostly, there have been few purchases, and at $1200 each, a bit out of our price ranges!
- Dual Core ??? MHZ processor
- Anyone care to donate one?
<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>12/2010</td>
<td>Logitech UART found (and live)</td>
</tr>
<tr>
<td>1/2011</td>
<td>Root package released (content bypass)</td>
</tr>
<tr>
<td>7/2011</td>
<td>Sony (Blu-ray) unit acquired</td>
</tr>
<tr>
<td></td>
<td>Sony unit rooted (SATA modification)</td>
</tr>
<tr>
<td></td>
<td>Sony recovery command execution found</td>
</tr>
<tr>
<td></td>
<td>Software root method found</td>
</tr>
<tr>
<td></td>
<td>Sony update encryption keys found, reversed, decrypted</td>
</tr>
<tr>
<td>8/2011</td>
<td>Revue 3.1 &quot;Honeycomb&quot; leaked</td>
</tr>
<tr>
<td>9/2011</td>
<td>Sony 3.1 Released</td>
</tr>
<tr>
<td></td>
<td>Sony TV acquired</td>
</tr>
<tr>
<td></td>
<td>Sony TV rooted</td>
</tr>
<tr>
<td>Date</td>
<td>Event (Continued)</td>
</tr>
<tr>
<td>----------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>10/2011</td>
<td>Sony bootloader shell found/downgrade achieved</td>
</tr>
<tr>
<td>11/2012</td>
<td>kexec ported as module to x86, unsigned kernels for Sony (saving for 3.2 rls)</td>
</tr>
<tr>
<td>12/2012</td>
<td>3.2 for Sony released</td>
</tr>
<tr>
<td>1/2012</td>
<td>Sony nodev recovery downgrade released</td>
</tr>
<tr>
<td>1/2012</td>
<td>Sony exploit package released (unsigned kernels)</td>
</tr>
<tr>
<td>3/2012</td>
<td>Revue signed module exploit achieved (needed root privileges)</td>
</tr>
<tr>
<td>4/2012</td>
<td>Logitech Revue kernel exploit (awaiting 3.2 release)</td>
</tr>
<tr>
<td>5/2012</td>
<td>Revue 3.2 Released</td>
</tr>
<tr>
<td>6/30/2012</td>
<td>NSZ-GP7 Acquired</td>
</tr>
<tr>
<td>6/30/2012</td>
<td>NSZ-GP7 Root Exploit</td>
</tr>
</tbody>
</table>
Sony NSZ-GP7

- Newest Sony device
- Released this month
- Tear down posted at GTVHacker.com
- CN2000 looks familiar!

http://gtvhelper.com/pres/dc20.ppt
Notice that last bit on the timeline? Yeah.

We gained root access on 6/30, and proceeded to explore.

Our goal is to get unsigned kernels running before a release, which may or may not be done already (you, with the microphone?)
Demo
Questions?

http://gtv hacker.com/pres/dc20.ppt
Thank you!

More information can be found at:

http://www.GTVHacker.com/

http://forum.GTVHacker.com/

http://blog.GTVHacker.com/