Hack the Gadget!

Daniel Maslowski
Agenda

- Hacks in the past
- Going beyond root
- Understanding your device
Hacks in the past
The Exploiteers, DEFCON 22 (2014)
group presentation quickly walking through a lot of devices - printers, smart bulbs, cameras, Android TV… taught about USB serial adapters and MMC (yay!).
finished with live action.

DualCore-AllTheThings
https://www.youtube.com/watch?v=h5PRvBpLuJs
Hack All The Things: 20 Devices in 45 Minutes

The Exploiteers, DEF CON 22 (2014)
Hack All The Things: 20 Devices in 45 Minutes

The Exploiteers, DEF CON 22 (2014)

- group presentation quickly walking through a lot of devices
- printers, smart bulbs, cameras, Android TV…
- taught about USB serial adapters and eMMC (yay!)
- finished with live act *Dual Core - All The Things*

https://www.youtube.com/watch?v=h5PRvBpLuJs
Hack everything: re-purposing everyday devices

Matt Evans at Linux.conf.au 2012 (2 years before The Exploiteers)

- Re-use hardware stuff!
- Don’t just consume… re-consume:)
- If you discover something cool, teach others and tell the world
- Collaborate at a local hackerspace

https://axio.ms/ Matt’s website
https://www.youtube.com/watch?v=VY9SBPo1Oy8
Hack everything: re-purposing everyday devices

Matt Evans at Linux.conf.au 2012 (2 years before The Exploiteers)
Hack everything: re-purposing everyday devices

Matt Evans at Linux.conf.au 2012 (2 years before The Exploiteers)

Re-use hardware stuff!

- Don’t just consume… re-consume :-)  
- If you discover something cool, teach others and tell the world
- Collaborate at a local hackerspace

https://axio.ms/ Matt’s website
https://www.youtube.com/watch?v=VY9SBPo1Oy8
Hellaphone: Replacing the Java in Android

John Floren at DEF CON 20, 2012

https://www.youtube.com/watch?v=EpTTU4lcR1Q

https://jflorenc.net/b/2015/8/18/2

*Hellaphone was a project we did at Sandia that stripped out the
Java portions of an Android stack and put Inferno in its place.*

https://github.com/floren/hellaphone
People are still doing it
People are still doing it

Turn an old smartphone into a 24/7 weather monitor – Solving problems by repurposing gadgets

By Julie Strietelmeier / February 5, 2022 / Articles / Do-It-Yourself, Repurpose / 36 Comments

We use affiliate links. If you buy something through the links on this page, we may earn a commission at no cost to you. Learn more.

https://the-gadgeteer.com/2022/02/05/turn-an-old-smartphone-into-a-24-7-weather-monitor-solving-problems-by-repurposing-gadgets/
The industry is doing it

So apparently imax theaters ran off of palm pilots for the quick turn reel unit. And nowadays, rather than having it run off a microcontroller or PC or raspberry pi or iPad, they just run a palm OS emulator? Lmao

Emulate Palm OS to reuse old software

https://twitter.com/torbar/status/1681073517989617664
I am doing it... or am I?
I am doing it... or am I?
I am doing it... or am I?

Gamification
I am doing it... or am I?

Gamification

AliExpress Diamond (not just Platinum :p)
I am doing it... or am I?

Gamification

AliExpress Diamond (not just Platinum :p)
Root on Arrival (tm)
I am doing it... or am I?

Gamification

- AliExpress Diamond (not just Platinum :p)
- Root on Arrival (tm)
- bell rings, package arrives - unwrap, solder, attach - boom, root!
Malware on TV boxes

Doyou own an Android TV Box similar to one of these:

- T95·AllWinnerH616
- T95Max·AllWinnerH618
- X12·Plus·RockChip3328
- X88·Pro·10·RockChip3328

...and have a folder named:

- /data/system/Corejava
- or a file named /data/system/shared_prefs/open_preference.xml

Your device is infected with malware, constantly trying to find a C2 server to upload 'telemetry' and await commands without your knowledge or permission. It's included with the device, straight from the merchant you order it from.

New motivation: rid of malware
Malware on TV boxes


Do you own an Android TV Box similar to one of these:

- T95 · AllWinner H616
- T95Max · AllWinner H618
- X12-Plus · RockChip 3328
- X88-Pro-10 · RockChip 3328

...and have a folder named:
/data/system/Corejava or a file named
/data/system/shared_prefs/open_preference.xml

Your device is infected with malware, constantly trying to find a C2 server to upload ‘telemetry’ and await commands without your knowledge or permission. It’s included with the device, straight from the merchant you ordered it from.
Malware on TV boxes


Do you own an Android TV Box similar to one of these:
- T95 · AllWinner H616
- T95Max · AllWinner H618
- X12-Plus · RockChip 3328
- X88-Pro-10 · RockChip 3328

…and have a folder named:
/data/system/.Corejava or a file named
/data/system/shared_prefs/open_preference.xml

Your device is infected with malware, constantly trying to find a C2 server to upload ‘telemetry’ and await commands without your knowledge or permission. It’s included with the device, straight from the merchant you ordered it from.

New motivation: rid of malware
T95 TV Box
Going beyond root
Gadget hacking and development boards
Gadget hacking and development boards
Talk to the SoC

WhystopattheOSlevel? Hackintosystem,sure…
Buildandrunyourown,makesuperawesome—it'sfeasible!
Therearemanygadgets,nottoo manySoCs/vendors,really.
Theyareoftenbasedonreferencedesigns.

Opportunity
Useupstreamcodeandadjust
Exchangewithcommunity
https://linux-sunxi.org/ShareVDI_R1
Talk to the SoC

Why stop at the OS level? Hack into the system, sure…
Talk to the SoC

Why stop at the OS level? Hack into the system, sure…
Build and run your own, make it super awesome - it’s feasible!
There are many gadgets, not too many SoCs/vendors, really.
They are often based on reference designs.
Talk to the SoC

Why stop at the OS level? Hack into the system, sure…
Build and run your own, make it super awesome - it’s feasible!
There are many gadgets, not too many SoCs/vendors, really.
They are often based on reference designs.

Opportunity
- use upstream code and adjust
- exchange with community

https://linux-sunxi.org/ShareVDI_R1
Talk to the SoC

Why stop at the OS level? Hack into the system, sure…
Build and run your own, make it super awesome - it’s feasible!
There are many gadgets, not too many SoCs/vendors, really.
They are often based on reference designs.

Opportunity
- use upstream code and adjust
- exchange with community

https://linux-sunxi.org/ShareVDI_R1
Some of those details are lies: the F133 (Allwinner SoC aka D1s) only has 512 Mbit DDR2 DRAM in-package, or 64 MiB. Is 1024x600 really HD?

https://www.amazon.de/Portable-Wireless-Carplay-Stereo-7-Inch/dp/B0C23SNRTC
Car Media Player

Portable Wireless Carplay Android Car Stereo 7 Inch HD Touchscreen Car MP5 Player with Mirrorlink Remote Control FM Radio USB 12 LED Camera

Brand: wepeculer

'113.96

Prices for items sold by Amazon include VAT. Depending on your delivery address, VAT may vary at Checkout. For other items, please see details.

Brand: wepeculer

Connectivity technology: Bluetooth, Auxiliary, Wi-Fi, USB

Controller type: Android

Compatible devices: Smartphone, Speaker

Connector Type: USB Type A, 3.5 mm Klinkke

Audio output mode: Stereo

Control method: Touch

About this item

- Mirror Link: This full touch screen car radio supports Mirror Link for iOS and Android smartphones. You can sync maps, movies etc. on the large 7 inch screen. The full touch HD display with a resolution of 1024 x 600 provides you with a clear and responsive viewing experience. Equipped with a remote control, it offers you a more convenient experience.

Some of those details are lies: the F133 (Allwinner SoC aka D1s) only has 512Mbit DDR2 DRAM in-package, or 64 MiB. Is 1024 x 600 really HD?

https://www.amazon.de/‑/en/Portable‑Wireless‑Carplay‑Touchscreen‑Mirrorlink/dp/B0C23SNRTC

Product details

- CPU F133
- 1 Gbit DRAM memory.
- Memory None
- 1024 x 600 screen resolution.
- 7 inch HD screen size
Some of those details are lies: the F133 (Allwinner SoC aka D1s) only has 512 Mbit DDR2 DRAM in-package, or 64MiB. Is 1024 x 600 really HD? …

https://www.amazon.de/-/en/Portable-Wireless-Carplay-Touchscreen-Mirrorlink/dp/B0C23SNRTC
DEMO: Talk to the SoC
Interludium: Leg Assembly
Interludium: Leg Assembly

Arm Assembly
& Reverse Engineering
Arm Assembly Internals

http://leg-assembly.com
https://azeria-labs.com/writing-arm-assembly-part-1/
Hello MRMCD!

```
ldr r0, =0x01c28000
mov r1, #0x4D
str r1, [r0]
mov r1, #0x52
str r1, [r0]
mov r1, #0x4D
str r1, [r0]
mov r1, #0x43
str r1, [r0]
mov r1, #0x44
str r1, [r0]

_loop: 
  b _loop
```
Hello MRMCD!

_start:
  ldr r0, =0x01c28000
  mov r1, #0x4D
  str r1, [r0]
  mov r1, #0x52
  str r1, [r0]
  mov r1, #0x4D
  str r1, [r0]
  mov r1, #0x43
  str r1, [r0]
  mov r1, #0x44
  str r1, [r0]
_loop:
  b _loop
DEMO: A little MMIO
Kernel hacking

- Bring up `find indicator stoseehowfaryouget inearlyasm, directMMIOonserialforsinglecharoutput

  ▶ becarefulwithregisters-theyhavespecialmeaninginearlyasm

  ▶ doinga `bl` willmessupthereturnaddress!

  ▶ `debug.S` reallyhandy, canprint2,4,8-digithexvaluesandASCII

- Share logs!

  earlycon, figureitout

  https://falstaff.agner.ch/2015/10/17/linux-earlyprintkearlycon-support-on-arm/

  ▶ for8250/16550: earlycon=uart,mmio32,$UARTBASE_ADDR

  loglevel=8, initcall_debug, kernelconfigoptions

- https://gist.github.com/apritzel/c128b29c601d180d32d68ee4c9ed8f47

- https://gist.github.com/orangecms/723a49c37f16c5d9dde2a9023669bf88
Kernel hacking

Bringup

- find **indicators** to see how far you get
- in early asm, direct MMIO on serial for single char output
- `arch/$ARCH/kernel/head.S`
  - be careful with registers - they have special meaning in early asm
  - doing a `b1` will mess up the return address!
  - `debug.S` **really handy**, can print 2,4,8-digit hex values and ASCII

---

Share logs! earlycon, figure it out

https://falstaff.agner.ch/2015/10/17/linux-earlyprintkearlycon-support-on-arm/

---

for 8250/16550:

```
earlycon=uart,mmio32,$UARTBASE_ADDR
loglevel=8
initcall_debug
```

---

https://gist.github.com/apritzel/c128b29c601d180d32d68ee4c9ed8f47
https://gist.github.com/orangecms/723a49c37f16c5d9dde2a9023669bf88
Kernel hacking

Bringup

- find indicators to see how far you get
- in early asm, direct MMIO on serial for single char output
- arch/\$ARCH/kernel/head.S
  - be careful with registers - they have special meaning in early asm
  - doing a `bl` will mess up the return address!
  - debug.S *really handy*, can print 2,4,8-digit hex values and ASCII

Share logs!

- earlycon, figure it out https://falstaff.agner.ch/2015/10/17/linux-earlyprintkearlycon-support-on-arm/
  - for 8250/16550: earlycon=uart,mmio32,\$UARTBASE_ADDR
  - loglevel=8, initcall_debug, kernel config options
Kernel hacking

**Bringup**

- find **indicators** to see how far you get in early asm, direct MMIO on serial for single char output
- `arch/$ARCH/kernel/head.S`
  - be careful with registers - they have special meaning in early asm
  - doing a br will mess up the return address!
  - debug.S *really handy*, can print 2,4,8-digit hex values and ASCII

**Share logs!**

- earlycon, figure it out https://falstaff.agner.ch/2015/10/17/linux-earlyprintkearlycon-support-on-arm/
  - for 8250/16550: earlycon=uart,mmio32,$UARTBASE_ADDR
- loglevel=8, initcall_debug, kernel config options https://gist.github.com/apritzel/c128b29c601d180d32d68ee4c9ed8f47
- https://gist.github.com/orangecms/723a49c37f16c5d9dde2a9023669bf88
Projects focusing on products

OpenWrt, pfSense/OPNsense routers, network gear, WiFi excellent OpenWrt wiki

OpenIPC (network) cameras

lot of tooling, tutorials, etc

OpenBMC, u-bmc board management controllers

remote OOB management

Start a new one - pick u-root and cpu

https://github.com/u-root/cpu

https://github.com/orangecms/arm-cpu

https://github.com/u-root/sidecore
Projects focusing on products

OpenWrt, pfSense/OPNsense
- routers, network gear, WiFi
- excellent OpenWrt wiki
Projects focusing on products

OpenWrt, pfSense/OPNsense
- routers, network gear, WiFi
- excellent OpenWrt wiki

OpenIPC
- (network) cameras
- lots of tooling, tutorials, etc

https://github.com/u-root/cpu
https://github.com/orangecms/arm-cpu
https://github.com/u-root/sidecore
Projects focusing on products

**OpenWrt, pfSense/OPNsense**
- routers, network gear, WiFi
- excellent OpenWrt wiki

**OpenIPC**
- (network) cameras
- lots of tooling, tutorials, etc

**OpenBMC, u-bmc**
- board management controllers
- remote OOB management
Projects focusing on products

**OpenWrt, pfSense/OPNsense**
- routers, network gear, WiFi
- excellent OpenWrt wiki

**OpenIPC**
- (network) cameras
- lots of tooling, tutorials, etc

**OpenBMC, u-bmc**
- board management controllers
- remote OOB management

Start a new one - pick u-root and cpu

- https://github.com/u-root/cpu
- https://github.com/orangcms/arm-cpu
- https://github.com/u-root/sidecore
A little userland

build-arm32.sh

#!/bin/sh

set -e

export GOARCH=arm
CPIO="/tmp/u-root-$GOARCH.cpio"

# build a root fs using the embedded template
go run . -uroot-source . -o "$CPIO" embedded

# https://github.com/u-root/u-root/#compression
xz --check=crc32 -9 --lzma2=dict=1MiB --stdout "$CPIO" |
da d conv=sync bs=512 of="$CPIO.xz"
Understanding your device
Firmware vs OS

**U-Boot**
- configs in `configs/` - they determine the ARCH themselves
- device trees in `arch/$ARCH/dts/`
- boards in `board/$VENDOR/` - emphasis on SoC, but not consistently

**Linux**
- configs in `arch/$ARCH/configs/` - `$ARCH` must be provided by user
- device trees in `arch/$ARCH/boot/dts/` [`VENDOR/`]
- board is described by firmware *and* own DTB, merged at runtime
Hardware Description: Device Tree

Standardization in progress; current version: 0.4

ADT must have a memory node provided by firmware, usually.


Armtimer frequency must also be in DT, as I learned.

I simply put them in the kernel’s DT, so I can do firmware without DT augmentation.

https://lore.kernel.org/linux-arm-kernel/25965de3-cc82-7fe6-6b3d-5754c329ac07@suse.de/
Hardware Description: Device Tree

devicetree.org

Standardization in progress; current version: 0.4
Hardware Description: Device Tree

devicetree.org

Standardization in progress; current version: 0.4

A DT must have a memory node - provided by firmware, usually.

Hardware Description: Device Tree

A DT must have a memory node - provided by firmware, usually.


Arm timer frequency must also be in DT, as I learned.

I simply put them in the kernel’s DT, so I can do firmware without DT augmentation.

https://lore.kernel.org/linux-arm-kernel/25965de3-cc82-7fe6-6b3d-5754c329ac07@suse.de/
Getting stuck

cat /sys/kernel/debug/devices_deferred

1c50000.ethernet

platform: wait for supplier

In this case, I missed describing the power supply. It was a wrong guess anyway. More later.
Getting stuck

```bash
#!/ cat /sys/kernel/debug/devices_deferred
1c50000.ethernet platform: wait for supplier
/soc/i2c@1c2ac00/pmic@34/regulators/dc1sw
```
Getting stuck

```
#!/usr/bin/env bash

cat /sys/kernel/debug/devices_deferred

1c50000.ethernet platform: wait for supplier
/soc/i2c@1c2ac00/pmic@34/regulators/dc1sw
```

In this case, I missed describing the power supply.
Getting stuck

```
#/ cat /sys/kernel/debug/devices_deferred
1c50000.ethernet       platform: wait for supplier
            /soc/i2c@1c2ac00/pmic@34/regulators/dc1sw
```

In this case, I missed describing the power supply.

It was a wrong guess anyway. More later.
Device Tree is nice, but…
Device Tree is nice, but…

The DT could be checked at build time!
Device Tree is nice, but…

The DT *could be checked at build time!*

Unless… the firmware is expected to provide (part of) it.

How about fallbacks?
Device Tree is nice, but…

The DT *could be checked at build time!*

Unless… the firmware is expected to provide (part of) it.

How about fallbacks?

**Solving Devicetree Issues, part 3.0**

Frank Rowand at ELCE 2016

https://www.youtube.com/watch?v=BDS6Hydtsx8

Device Tree is nice, but…

The DT *could be checked at build time!*

Unless… the firmware is expected to provide (part of) it.

How about fallbacks?

**Solving Devicetree Issues, part 3.0**
Frank Rowand at ELCE 2016
https://www.youtube.com/watch?v=BDS6Hydtsx8
Some great ideas which never landed upstream. Anyone?
Living the lie

DeviceTree is a tree - but your hardware is not!

Clocks, interrupts, GPIO pins, power supplies are all across.

Some references in DT are just loose strings, e.g., phy-supply.

https://elinux.org/Device_Tree_Mysteries#Phandle

Let's create a device tree visualizer! :‑)
Device Tree is a tree - but your hardware is **not**!
Device Tree is a tree - but your hardware is not!

Clocks, interrupts, GPIO pins, power supplies are all across.
Device Tree is a tree - but your hardware is not!

Clocks, interrupts, GPIO pins, power supplies are all across.

Some references in DT are just loose strings, e.g., phy-supply.
Device Tree is a tree - but your hardware is **not**!

Clocks, interrupts, GPIO pins, power supplies are all across.

Some references in DT are just loose strings, e.g., `phy-supply`.

https://elinux.org/Device_Tree_Mysteries#Phandle
Living the lie

Device Tree is a tree - but your hardware is not!
Clocks, interrupts, GPIO pins, power supplies are all across.
Some references in DT are just loose strings, e.g., phy-supply.
https://elinux.org/Device_Tree_Mysteries#Phandle
Let’s create a device tree visualizer! :-)
Tracing Components
Tracing Components

SoC platformsmay use PMICs to supply power to components.
SoC platforms may use PMICs to supply power to components.
SoC platforms may use PMICs to supply power to components.
DEMO: Firmwareless full stack
Small computers everywhere
Small computers everywhere

MCUs getting closer to application processors
Small computers everywhere

- MCUs getting closer to application processors
- FreeRTOS, Zephyr, Hubris, embOS, EPOS, LiteOS, Melis…

You can get one for free: Wettersonde
https://github.com/arnobert/rs41_rust
Small computers everywhere

- MCUs getting closer to application processors
- FreeRTOS, Zephyr, Hubris, embOS, EPOS, LiteOS, Melis…

Many microcontrollers are usually more open.

You can get one for free: Wettersonde
https://github.com/arnobert/rs41_rust
Small computers everywhere

- MCUs getting closer to application processors
- FreeRTOS, Zephyr, Hubris, embOS, EPOS, LiteOS, Melis…

Many microcontrollers are usually more open.

You can get one for free: Wettersonde

https://github.com/arnobert/rs41_rust
Hardware keeps changing (really?)
Hardware keeps changing (really?)

AMP being established; https://www.openampproject.org/
- Bouffalo Lab BL808 (MCU + app core)
- JH7110 (monitor + 4 app cores)

RPi is similar: starting on GPU, releasing Arm cores thereafter

AMP widen the attack surface (!)
- Same thing: baseband, Bluetooth etc in phones!
- Desktop/SBC audio cores DMA into shared DRAM is nothing new
- open audio firmware attempts do exist: https://www.sofproject.org/
- Same with components running in different privilege levels:
Hardware keeps changing (really?)

AMP being established; https://www.openampproject.org/
  - Bouffalo Lab BL808 (MCU + app core)
  - JH7110 (monitor + 4 app cores)

RPi is similar: starting on GPU, releasing Arm cores thereafter
Hardware keeps changing (really?)

AMP being established; https://www.openampproject.org/
- Bouffalo Lab BL808 (MCU + app core)
- JH7110 (monitor + 4 app cores)

RPi is similar: starting on GPU, releasing Arm cores thereafter

AMP widens the attack surface (!)
- same thing: baseband, Bluetooth etc in phones!
Hardware keeps changing (really?)

- AMP being established; https://www.openampproject.org/
  - Bouffalo Lab BL808 (MCU + app core)
  - JH7110 (monitor + 4 app cores)

- RPi is similar: starting on GPU, releasing Arm cores thereafter

- AMP widens the attack surface (!)
  - same thing: baseband, Bluetooth etc in phones!

- desktop/SBC audio cores DMAing to shared DRAM is nothing new
  - open audio firmware attempts do exist: https://www.sofproject.org/
Hardware keeps changing (really?)

- AMP being established; https://www.openampproject.org/
  - Bouffalo Lab BL808 (MCU + app core)
  - JH7110 (monitor + 4 app cores)

- RPi is similar: starting on GPU, releasing Arm cores thereafter

- AMP widens the attack surface (!)
  - same thing: baseband, Bluetooth etc in phones!

- desktop/SBC audio cores DMAing to shared DRAM is nothing new
  - open audio firmware attempts do exist: https://www.sofproject.org/

- same with components running in different privilege levels:
Related

Repurposing Gadgets

Drivers from Outer Space (CLT 2022)

Platform System Interface - Design und Evaluation holistischer Computerarchitektur (rC3 2022)

Die wirre Welt der kleinen Computer (Tübix 2023)
Thank you! :)

Follow Me

Daniel Maslowski

https://github.com/orangecms
https://twitter.com/orangecms
https://mastodon.social/@cyrevolt
https://youtube.com/@cyrevolt
https://twitch.tv/cyrevolt


License: CC BY 4.0 https://creativecommons.org/licenses/by/4.0/