Qiling Framework:
Learn how to build a fuzzer based on a 1day bug

HITB Lockdown 002, Virtual Lab
July, 2020

KaiJern LAU, kj -at- qiling.io
NGUYEN Anh Quynh, aquynh -at- gmail.com
huitao, CHEN null -at- qiling.io
TianZe DING, dliv3 -at- gmail.com
BoWen SUN, w1tcher.bupt -at- gmail.com
Tong YU, spikeinhouse -at- gmail.com

twitter: @qiling_io https://qiling.io

About xwings

JD.COM

Beijing, Stays in the lab 24/7 by hoping making the world a better place

- IoT Research
- Blockchain Research
- Fun Security Research

Qiling Framework

Cross platform and multi architecture advanced binary emulation framework

- https://qiling.io
- Lead Developer
- Founder

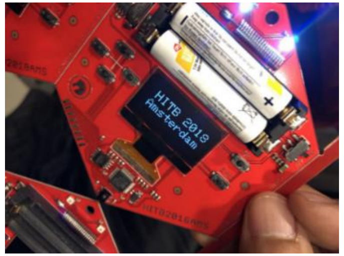
HACKERSBADGE.COM

Badge Maker

Electronic fan boy, making toys from hacker to hacker

- Reversing Binary
- Reversing IoT Devices
- Part Time CtF player

Badge Designer for Hacking Conferences



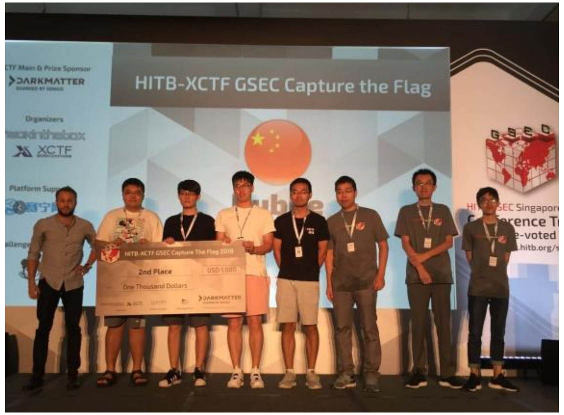
Some Recent Talk (Partial)

- 2016, Qcon, Beijing, Speaker, nRF24L01 Hijacking
- 2016, Kcon, Beijing, Speaker, Capstone Unicorn Keystone
- 2017, Kcon, Beijing, IoT Hacking Trainer
- 2018, Kcon, Beijing, IoT Hacking Training
- 2018, Brucon, Brussel, Speaker, IoT Virtualization
- 2018, H2HC, San Paolo, Speaker, IoT Virtualization
- 2018, HITB, Beijing/Dubai, Speaker, IoT Virtualization
- 2018, beVX, Hong Kong, Speaker, HackCUBE - Hardware Hacking
- 2019, DEFCON USA, Qiling Framework Preview
- 2019, Zeronights, Qiling Framework to Public
- 2020, Nullcon GOA, Building Reversing Tools with Qiling
- 2020, HITB AMS, Building Reversing Tools with Qiling
- 2020, HITB Singapore, Training, How to Hack IoT with Qiling
- 2020, Blackhat USA, Building IoT Fuzzer with Qiing
- 2020, Blackhat Singapore, Building Fuzzer with Qiing

Qiling Framework

- Cross platform and cross architecture binary instrumentation framework
- Emulate and instrument ARM, ARM64, MIPS, X86 and X8664
- Emulate and instrument Linux, MacOS, iphoneOS, Windows and FreeBSD
- High-level Python API access to register, CPU and memory
- 1,100+ Github star, more than 3,000 pypi download, 40+ contributors worldwide
- Contributor from Dell, Intel, Fireeye and etc

About Dliv3/w1tcher/Null/Sp1ke



THE SHEPHERD LAB 京东牧者安全实验室

DUBHE HACKING SECURITY

Rest of the team members are from theshepherdlab , Dubhe CTF team & community

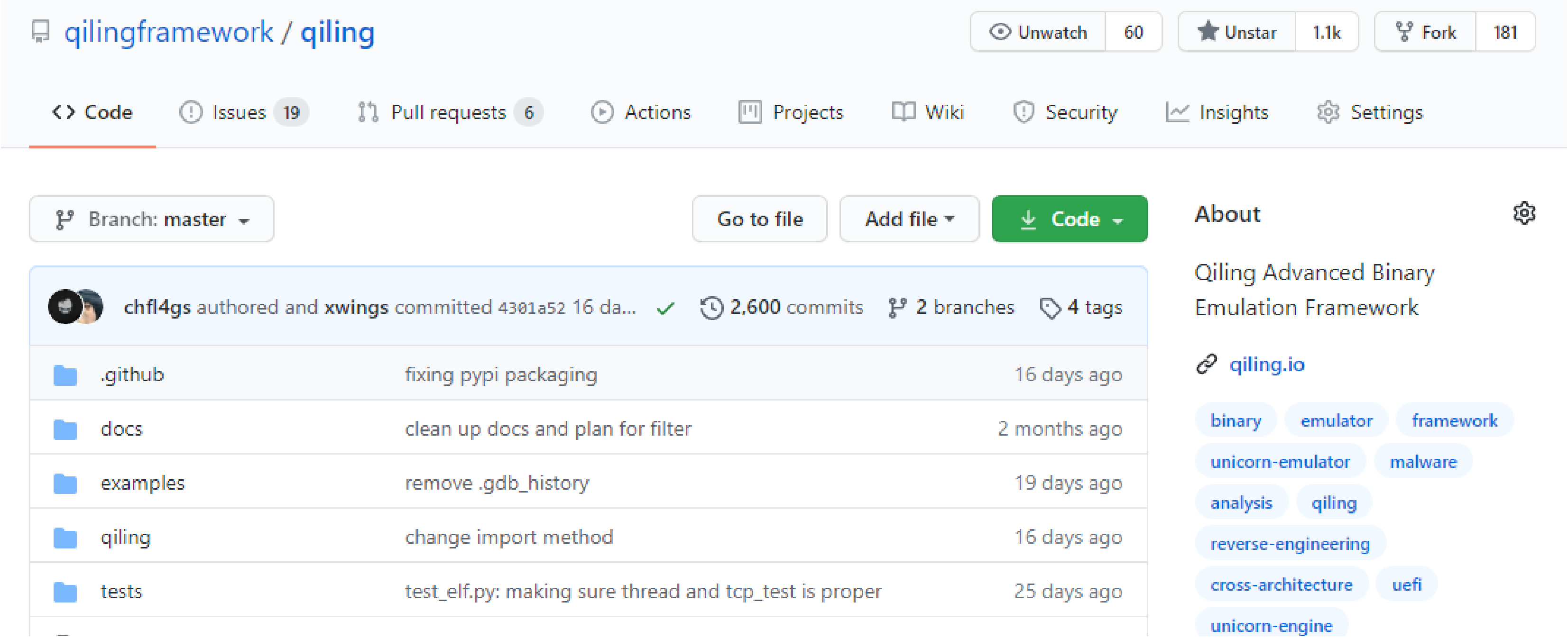
NGUYEN Anh Quynh

- Nanyang Technological University, Singapore
- PhD in Computer Science
- Operating System, Virtual Machine, Binary analysis, etc
- Usenix, ACM, IEEE, LNCS, etc
- Blackhat USA/EU/Asia, DEFCON, Recon, HackInTheBox, Syscan, etc
- Capstone disassembler: http://capstone-engine.org
- Unicorn emulator: http://unicorn-engine.org
- Keystone assembler: http://keystone-engine.org

https://github.com/qilingframework/qiling

- Motivation
- Qiling framework
- Design & implementation
- Build dynamic analysis tools on top of Qiling Framework
- Hands On

Star us



About

Qiling Advanced Binary Emulation Framework

qiling.io

binary emulator framework unicorn-emulator malware analysis qiling reverse-engineering cross-architecture uefi unicorn-engine

chfl4gs authored and xwings committed 4301a52 16 da... ✓ 2,600 commits 2 branches 4 tags

.github	fixing pypi packaging	16 days ago
docs	clean up docs and plan for filter	2 months ago
examples	remove .gdb_history	19 days ago
qiling	change import method	16 days ago
tests	test_elf.py: making sure thread and tcp_test is proper	25 days ago

Internet of Things

What is IoT

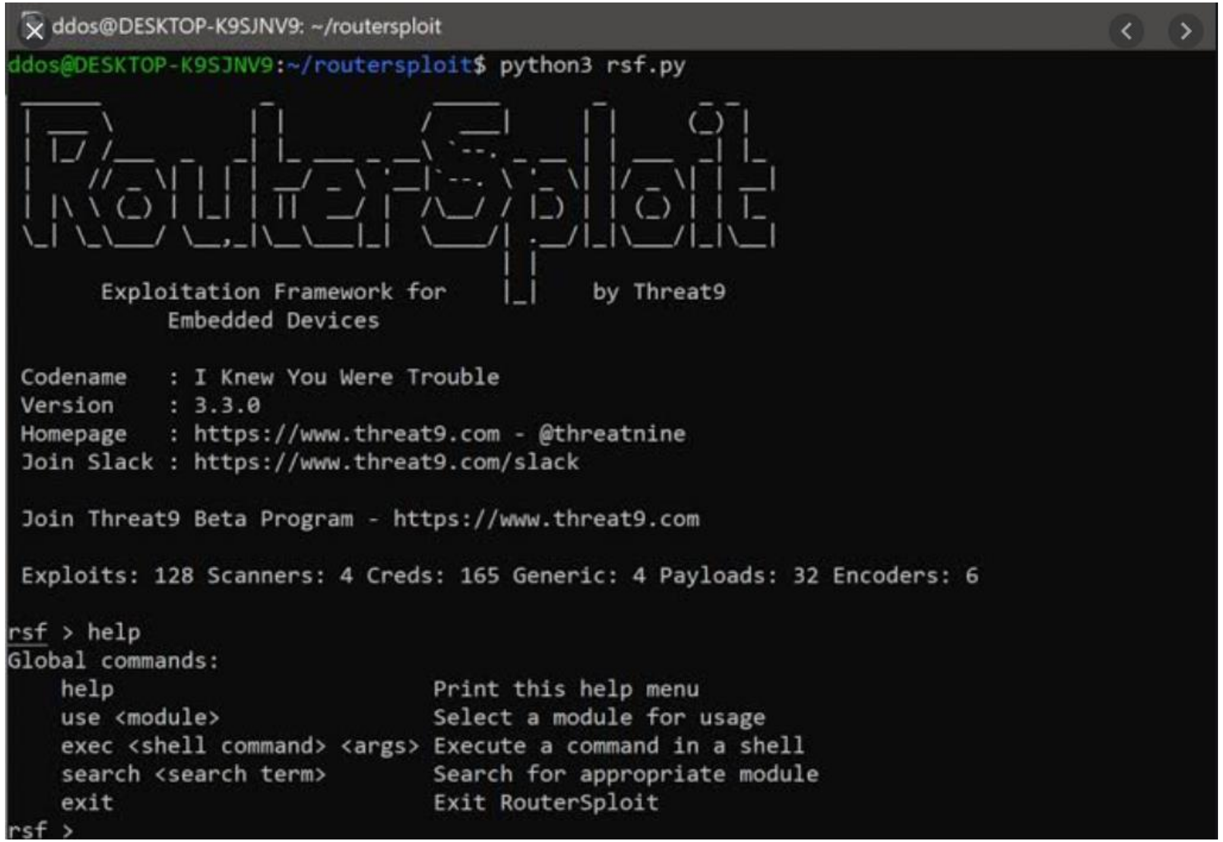
And more on our smartphone app: (Now available on Qi teamaker Vita+)
android
iOS

IoT

- Camera
- Air-con
- TV
- FAN
- Heater
- Fridge
- Watch
- Lock
- Security
- Kitchen
- Phone

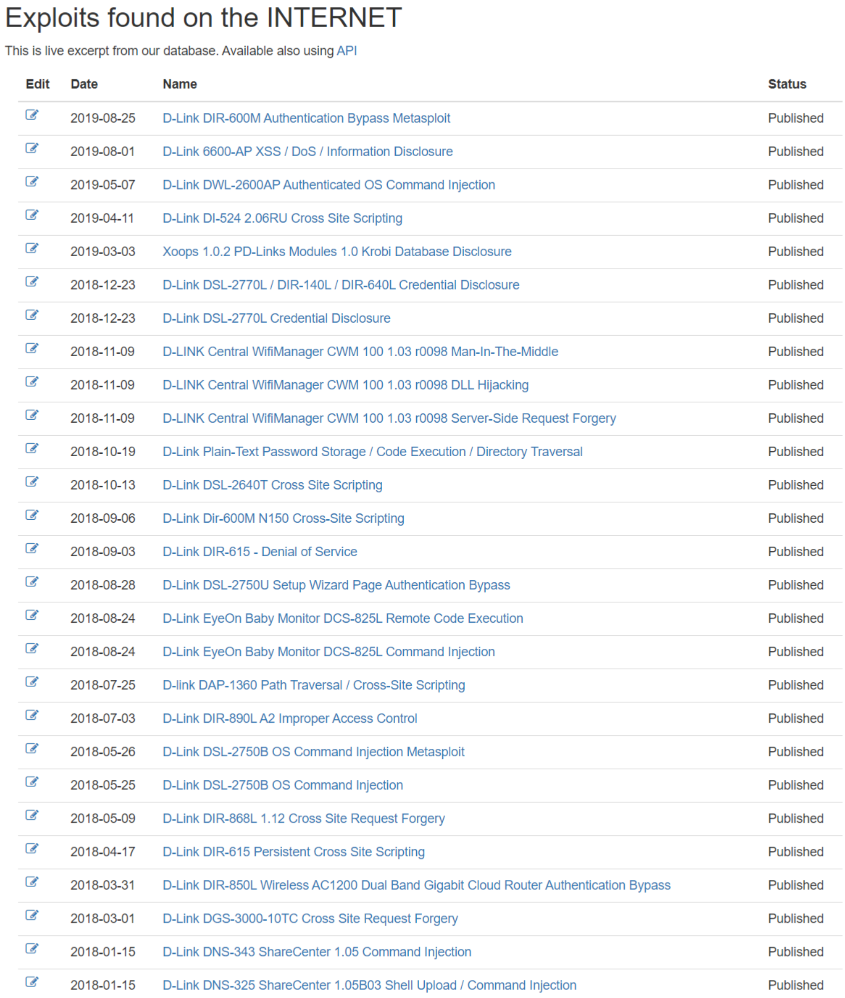
Traditional IoT Hacking

The Web Hacker



ddos@DESKTOP-K9SJNV9: ~/routersploit

```
ddos@DESKTOP-K9SJNV9:~/routersploit$ python3 rsf.py
 ______            _            _____       _       _ _
 | ___ \          | |          /  ___|     | |     (_) |
 | |_/ /___  _   _| |_ ___ _ __\ `--. _ __ | | ___  _| |_
 |    // _ \| | | | __/ _ \ '__|`--. \ '_ \| |/ _ \| | __|
 | |\ \ (_) | |_| | ||  __/ |  /\__/ / |_) | | (_) | | |_
 \_| \_\___/ \__,_|\__\___|_|  \____/| .__/|_|\___/|_|\__|
                                     | |
     Exploitation Framework for      |_|      by Threat9
            Embedded Devices

 Codename   : I Knew You Were Trouble
 Version    : 3.3.0
 Homepage   : https://www.threat9.com - @threatnine
 Join Slack : https://www.threat9.com/slack

 Join Threat9 Beta Program - https://www.threat9.com

 Exploits: 128 Scanners: 4 Creds: 165 Generic: 4 Payloads: 32 Encoders: 6

rsf > help
Global commands:
    help                        Print this help menu
    use <module>                Select a module for usage
    exec <shell command> <args> Execute a command in a shell
    search <search term>        Search for appropriate module
    exit                        Exit RouterSploit
rsf >
```



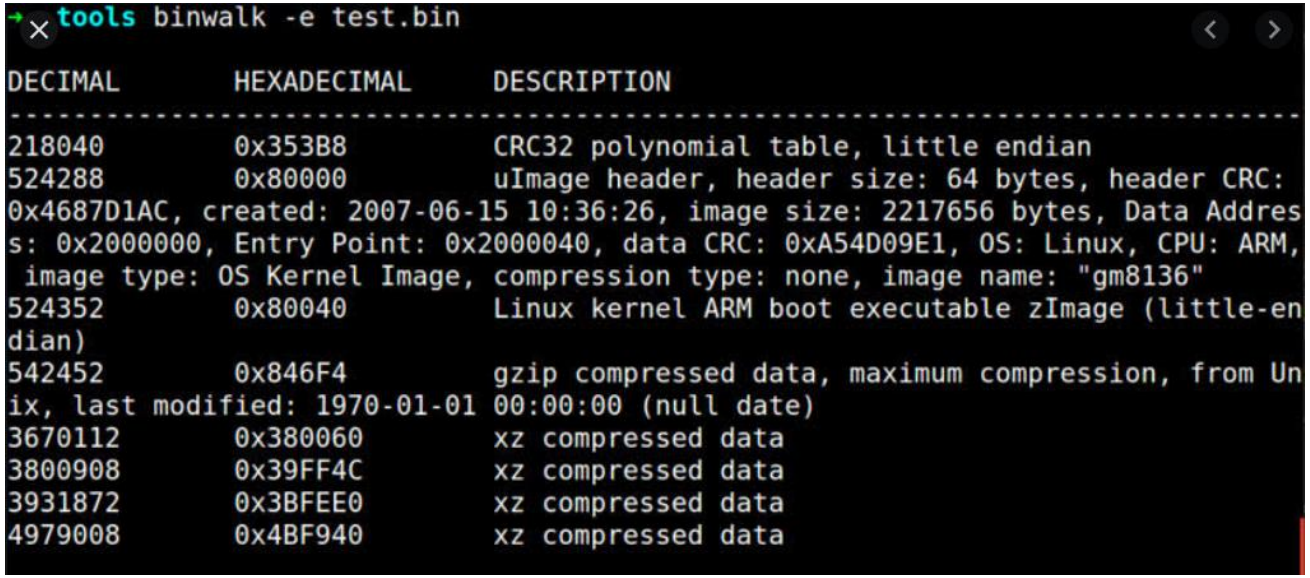
Exploits found on the INTERNET

This is live excerpt from our database. Available also using API

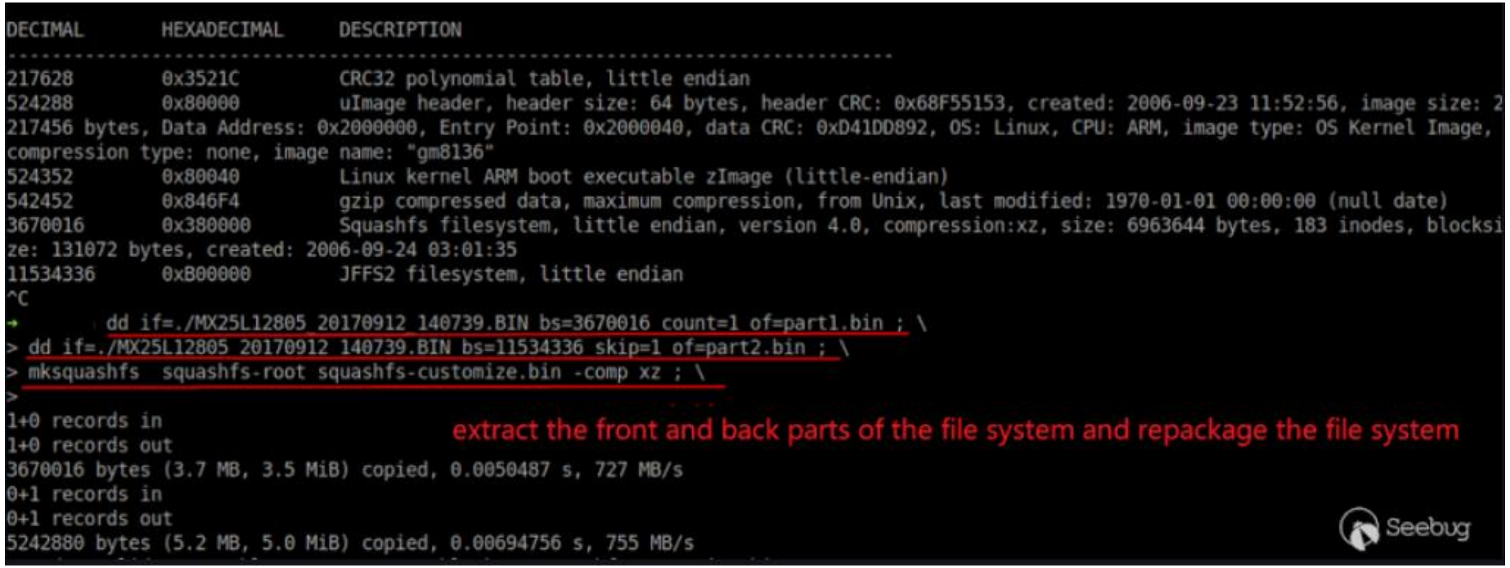
Edit	Date	Name	Status
✎	2019-08-25	D-Link DIR-600M Authentication Bypass Metasploit	Published
✎	2019-08-01	D-Link 6600-AP XSS / DoS / Information Disclosure	Published
✎	2019-05-07	D-Link DWL-2600AP Authenticated OS Command Injection	Published
✎	2019-04-11	D-Link DI-524 2.06RU Cross Site Scripting	Published
✎	2019-03-03	Xoops 1.0.2 PD-Links Modules 1.0 Krobi Database Disclosure	Published
✎	2018-12-23	D-Link DSL-2770L / DIR-140L / DIR-640L Credential Disclosure	Published
✎	2018-12-23	D-Link DSL-2770L Credential Disclosure	Published
✎	2018-11-09	D-LINK Central WifiManager CWM 100 1.03 r0098 Man-In-The-Middle	Published
✎	2018-11-09	D-LINK Central WifiManager CWM 100 1.03 r0098 DLL Hijacking	Published
✎	2018-11-09	D-LINK Central WifiManager CWM 100 1.03 r0098 Server-Side Request Forgery	Published
✎	2018-10-19	D-Link Plain-Text Password Storage / Code Execution / Directory Traversal	Published
✎	2018-10-13	D-Link DSL-2640T Cross Site Scripting	Published
✎	2018-09-06	D-Link Dir-600M N150 Cross-Site Scripting	Published
✎	2018-09-03	D-Link DIR-615 - Denial of Service	Published
✎	2018-08-28	D-Link DSL-2750U Setup Wizard Page Authentication Bypass	Published
✎	2018-08-24	D-Link EyeOn Baby Monitor DCS-825L Remote Code Execution	Published
✎	2018-08-24	D-Link EyeOn Baby Monitor DCS-825L Command Injection	Published
✎	2018-07-25	D-link DAP-1360 Path Traversal / Cross-Site Scripting	Published
✎	2018-07-03	D-Link DIR-890L A2 Improper Access Control	Published
✎	2018-05-26	D-Link DSL-2750B OS Command Injection Metasploit	Published
✎	2018-05-25	D-Link DSL-2750B OS Command Injection	Published
✎	2018-05-09	D-Link DIR-868L 1.12 Cross Site Request Forgery	Published
✎	2018-04-17	D-Link DIR-615 Persistent Cross Site Scripting	Published
✎	2018-03-31	D-Link DIR-850L Wireless AC1200 Dual Band Gigabit Cloud Router Authentication Bypass	Published
✎	2018-03-01	D-Link DGS-3000-10TC Cross Site Request Forgery	Published
✎	2018-01-15	D-Link DNS-343 ShareCenter 1.05 Command Injection	Published
✎	2018-01-15	D-Link DNS-325 ShareCenter 1.05B03 Shell Upload / Command Injection	Published

exploitalert.com

Firmware Hacking



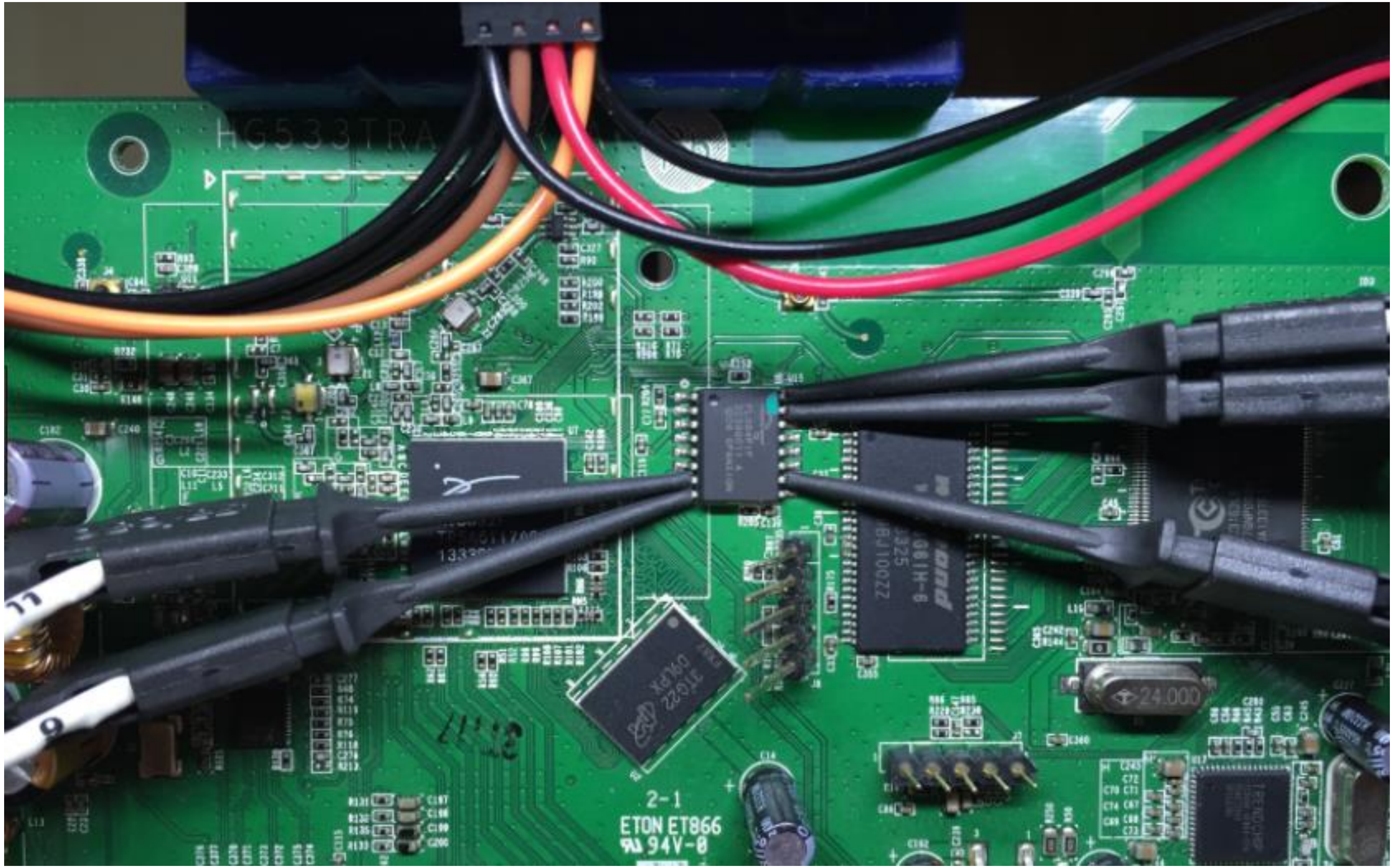
```
→ tools binwalk -e test.bin

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
218040        0x353B8         CRC32 polynomial table, little endian
524288        0x80000         uImage header, header size: 64 bytes, header CRC:
0x4687D1AC, created: 2007-06-15 10:36:26, image size: 2217656 bytes, Data Addres
s: 0x2000000, Entry Point: 0x2000040, data CRC: 0xA54D09E1, OS: Linux, CPU: ARM,
 image type: OS Kernel Image, compression type: none, image name: "gm8136"
524352        0x80040         Linux kernel ARM boot executable zImage (little-en
dian)
542452        0x846F4         gzip compressed data, maximum compression, from Un
ix, last modified: 1970-01-01 00:00:00 (null date)
3670112       0x380060        xz compressed data
3800908       0x39FF4C        xz compressed data
3931872       0x3BFEE0        xz compressed data
4979008       0x4BF940        xz compressed data
```



```
DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
217628        0x3521C         CRC32 polynomial table, little endian
524288        0x80000         uImage header, header size: 64 bytes, header CRC: 0x68F55153, created: 2006-09-23 11:52:56, image size: 2
217456 bytes, Data Address: 0x2000000, Entry Point: 0x2000040, data CRC: 0xD41DD892, OS: Linux, CPU: ARM, image type: OS Kernel Image,
compression type: none, image name: "gm8136"
524352        0x80040         Linux kernel ARM boot executable zImage (little-endian)
542452        0x846F4         gzip compressed data, maximum compression, from Unix, last modified: 1970-01-01 00:00:00 (null date)
3670016       0x380000        Squashfs filesystem, little endian, version 4.0, compression:xz, size: 6963644 bytes, 183 inodes, blocksi
ze: 131072 bytes, created: 2006-09-24 03:01:35
11534336      0xB00000        JFFS2 filesystem, little endian
^C
→        dd if=./MX25L12805_20170912_140739.BIN bs=3670016 count=1 of=part1.bin ; \
> dd if=./MX25L12805_20170912_140739.BIN bs=11534336 skip=1 of=part2.bin ; \
> mksquashfs  squashfs-root squashfs-customize.bin -comp xz ; \
>
1+0 records in
1+0 records out
3670016 bytes (3.7 MB, 3.5 MiB) copied, 0.0050487 s, 727 MB/s
0+1 records in
0+1 records out
5242880 bytes (5.2 MB, 5.0 MiB) copied, 0.00694756 s, 755 MB/s
```

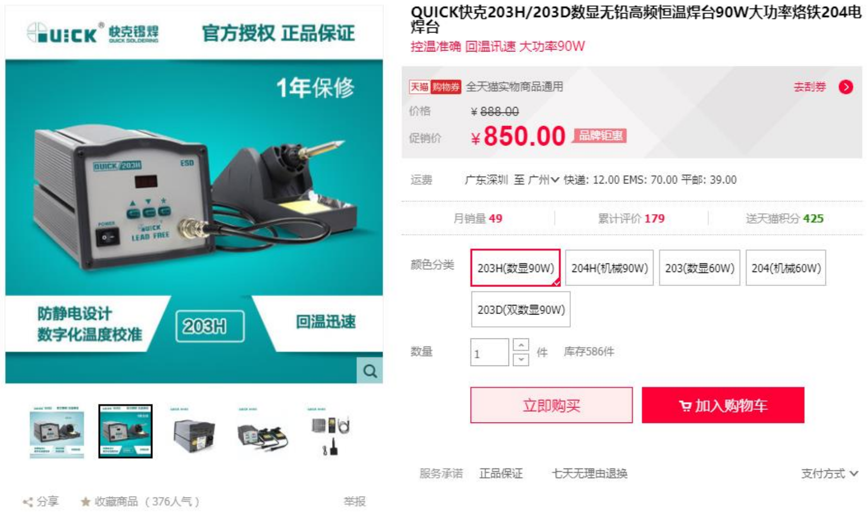
extract the front and back parts of the file system and repackage the file system

Hardware Hacking



Additional Note: What To Buy

Hot Air Gun



QUICK快克203H/203D数显无铅高频恒温焊台90W大功率烙铁204电焊台

控温准确 回温讯速 大功率90W

天猫 购物券 全天猫实物商品通用 去刮券

价格 ¥ ~~888.00~~

促销价 ¥850.00 品牌钜惠

运费 广东深圳 至 广州 快递: 12.00 EMS: 70.00 平邮: 39.00

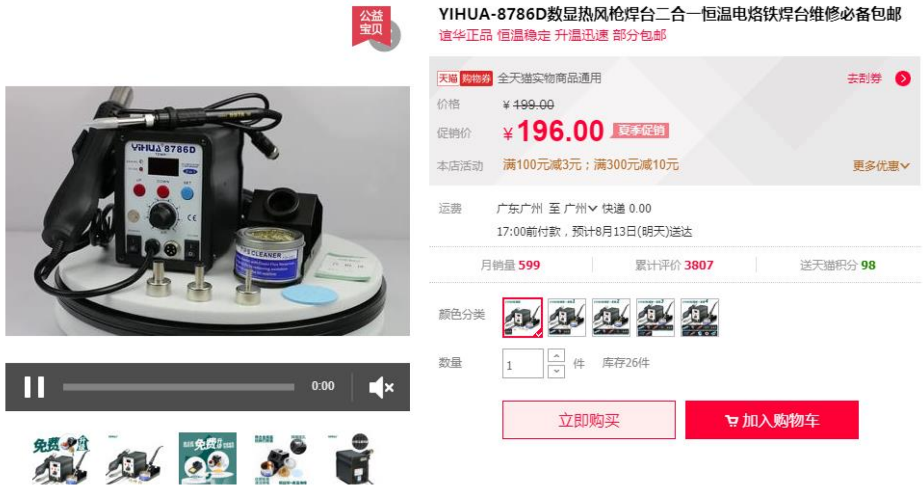
月销量 49 | 累计评价 179 | 送天猫积分 425

颜色分类 203H(数显90W) 204H(机械90W) 203(数显60W) 204(机械60W) 203D(双数显90W)

数量 1 件 库存586件

立即购买 | 加入购物车

服务承诺 正品保证 七天无理由退换 支付方式



YIHUA-8786D数显热风枪焊台二合一恒温电烙铁焊台维修必备包邮

谊华正品 恒温稳定 升温迅速 部分包邮

天猫 购物券 全天猫实物商品通用 去刮券

价格 ¥ ~~199.00~~

促销价 ¥196.00 夏季促销

本店活动 满100元减3元；满300元减10元 更多优惠

运费 广东广州 至 广州 快递 0.00

17:00前付款，预计8月13日(明天)送达

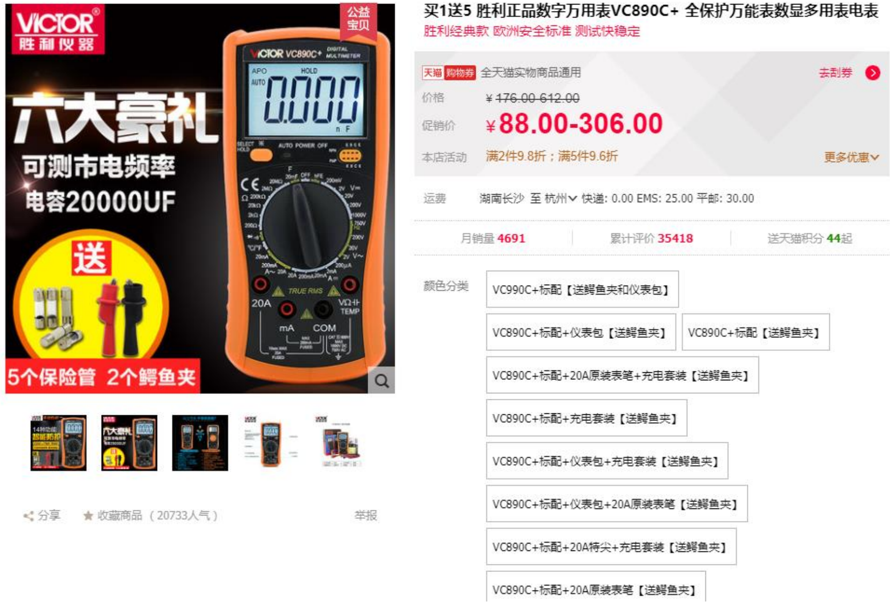
月销量 599 | 累计评价 3807 | 送天猫积分 98

颜色分类

数量 1 件 库存26件

立即购买 | 加入购物车

Multi Meter



分享 收藏商品（20733人气） 举报

买1送5 胜利正品数字万用表VC890C+ 全保护万能表数显多用表电表

胜利经典款 欧洲安全标准 测试快稳定

天猫 购物券 全天猫实物商品通用 去刮券

价格 ¥ ~~176.00-612.00~~

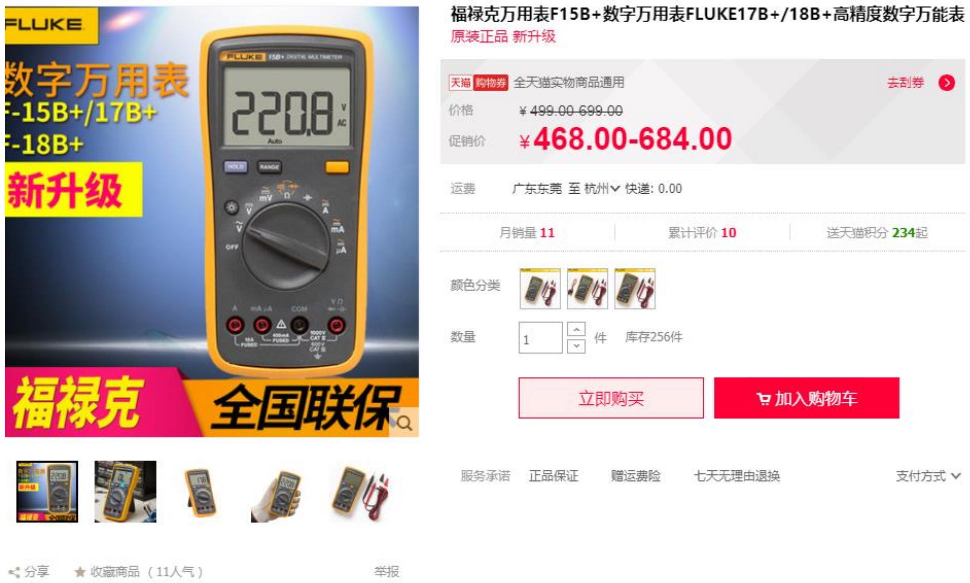
促销价 ¥88.00-306.00

本店活动 满2件9.8折；满5件9.6折 更多优惠

运费 湖南长沙 至 杭州 快递: 0.00 EMS: 25.00 平邮: 30.00

月销量 4691 | 累计评价 35418 | 送天猫积分 44起

颜色分类
- VC990C+标配【送鳄鱼夹和仪表包】
- VC890C+标配+仪表包【送鳄鱼夹】
- VC890C+标配【送鳄鱼夹】
- VC890C+标配+20A原装表笔+充电套装【送鳄鱼夹】
- VC890C+标配+充电套装【送鳄鱼夹】
- VC890C+标配+仪表包+充电套装【送鳄鱼夹】
- VC890C+标配+仪表包+20A原装表笔【送鳄鱼夹】
- VC890C+标配+20A特尖+充电套装【送鳄鱼夹】
- VC890C+标配+20A原装表笔【送鳄鱼夹】



分享 收藏商品（11人气） 举报

福禄克万用表F15B+数字万用表FLUKE17B+/18B+高精度数字万能表

原装正品 新升级

天猫 购物券 全天猫实物商品通用 去刮券

价格 ¥ ~~499.00-699.00~~

促销价 ¥468.00-684.00

运费 广东东莞 至 杭州 快递: 0.00

月销量 11 | 累计评价 10 | 送天猫积分 234起

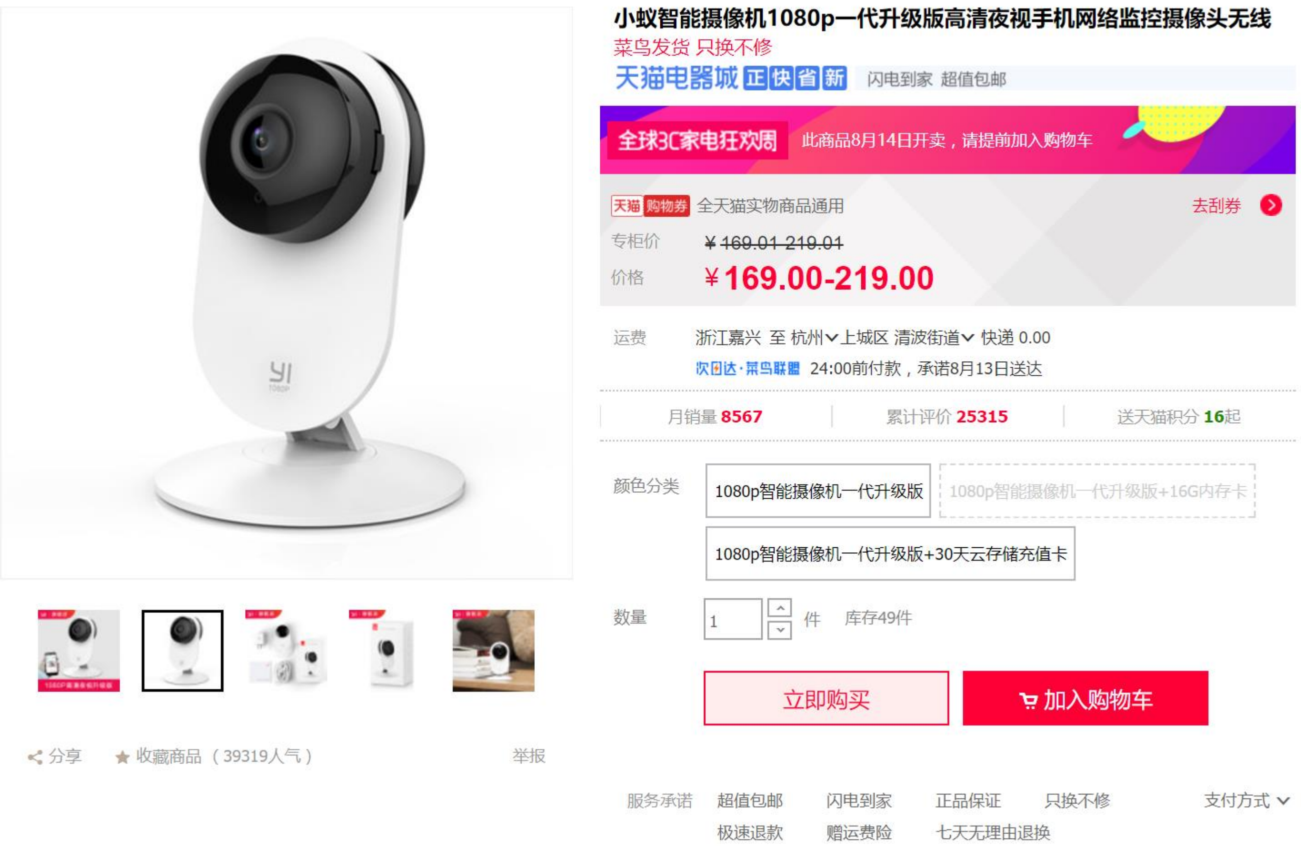
颜色分类

数量 1 件 库存256件

服务承诺 正品保证 赠运费险 七天无理由退换 支付方式

Case Study

Buying a China Only Cam



小蚁智能摄像机1080p一代升级版高清夜视手机网络监控摄像头无线

菜鸟发货 只换不修

天猫电器城 正 快 省 新 闪电到家 超值包邮

全球3C家电狂欢周 此商品8月14日开卖，请提前加入购物车

天猫 购物券 全天猫实物商品通用

专柜价 ~~¥169.01-219.01~~

价格 ¥169.00-219.00

运费 浙江嘉兴 至 杭州 上城区 清波街道 快递 0.00

次日达·菜鸟联盟 24:00前付款，承诺8月13日送达

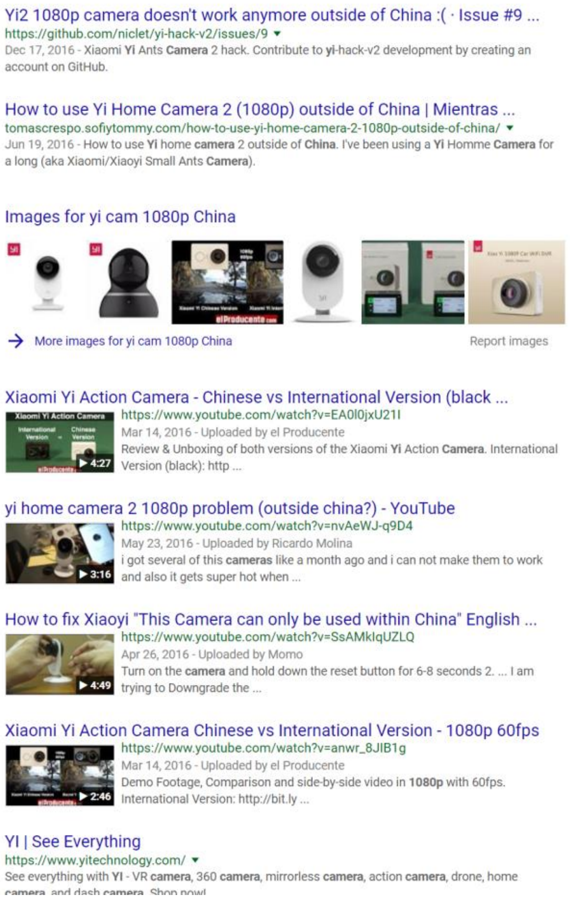
月销量 8567 | 累计评价 25315 | 送天猫积分 16起

颜色分类 1080p智能摄像机一代升级版 | 1080p智能摄像机一代升级版+16G内存卡 | 1080p智能摄像机一代升级版+30天云存储充值卡

数量 1 件 库存49件

服务承诺 超值包邮 闪电到家 正品保证 只换不修 极速退款 赠运费险 七天无理由退换

Talking Cam’s Warming



Yi2 1080p camera doesn't work anymore outside of China :(· Issue #9 ...
https://github.com/niclet/yi-hack-v2/issues/9 ▾
Dec 17, 2016 - Xiaomi **Yi** Ants **Camera** 2 hack. Contribute to **yi**-hack-v2 development by creating an account on GitHub.

How to use Yi Home Camera 2 (1080p) outside of China | Mientras ...
tomascrespo.sofiytommy.com/how-to-use-yi-home-camera-2-1080p-outside-of-china/ ▾
Jun 19, 2016 - How to use **Yi** home **camera** 2 outside of **China**. I've been using a **Yi** Homme **Camera** for a long (aka Xiaomi/Xiaoyi Small Ants **Camera**).

Images for yi cam 1080p China

→ More images for yi cam 1080p China

Report images

Xiaomi Yi Action Camera - Chinese vs International Version (black ...
https://www.youtube.com/watch?v=EA0l0jxU21I
Mar 14, 2016 - Uploaded by el Producente
Review & Unboxing of both versions of the Xiaomi **Yi** Action **Camera**. International Version (black): http ...

yi home camera 2 1080p problem (outside china?) - YouTube
https://www.youtube.com/watch?v=nvAeWJ-q9D4
May 23, 2016 - Uploaded by Ricardo Molina
i got several of this **cameras** like a month ago and i can not make them to work and also it gets super hot when ...

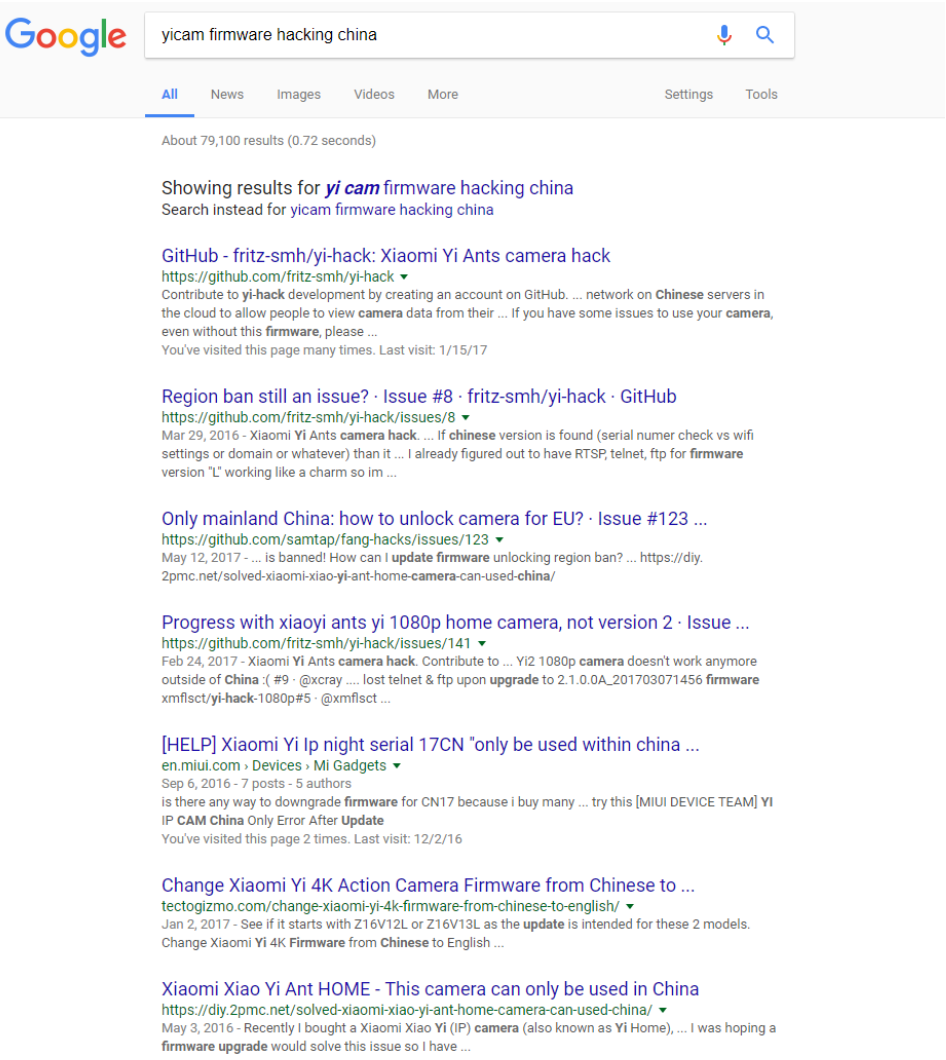
How to fix Xiaoyi "This Camera can only be used within China" English ...
https://www.youtube.com/watch?v=SsAMkIqUZLQ
Apr 26, 2016 - Uploaded by Momo
Turn on the **camera** and hold down the reset button for 6-8 seconds 2. ... I am trying to Downgrade the ...

Xiaomi Yi Action Camera Chinese vs International Version - 1080p 60fps
https://www.youtube.com/watch?v=anwr_8JIB1g
Mar 14, 2016 - Uploaded by el Producente
Demo Footage, Comparison and side-by-side video in **1080p** with 60fps. International Version: http://bit.ly ...

YI | See Everything
https://www.yitechnology.com/ ▾
See everything with **YI** - VR **camera**, 360 **camera**, mirrorless **camera**, action **camera**, drone, home **camera**, and dash **camera**. Shop now!

“Not Allow To Use Outside China”

Answer from Google and Baidu



About 79,100 results (0.72 seconds)

Showing results for ***yi cam*** firmware hacking china
Search instead for yicam firmware hacking china

GitHub - fritz-smh/yi-hack: Xiaomi Yi Ants camera hack
https://github.com/fritz-smh/yi-hack ▾
Contribute to **yi-hack** development by creating an account on GitHub. ... network on **Chinese** servers in the cloud to allow people to view **camera** data from their ... If you have some issues to use your **camera**, even without this **firmware**, please ...
You've visited this page many times. Last visit: 1/15/17

Region ban still an issue? · Issue #8 · fritz-smh/yi-hack · GitHub
https://github.com/fritz-smh/yi-hack/issues/8 ▾
Mar 29, 2016 - Xiaomi **Yi** Ants **camera hack**. ... If **chinese** version is found (serial numer check vs wifi settings or domain or whatever) than it ... I already figured out to have RTSP, telnet, ftp for **firmware** version "L" working like a charm so im ...

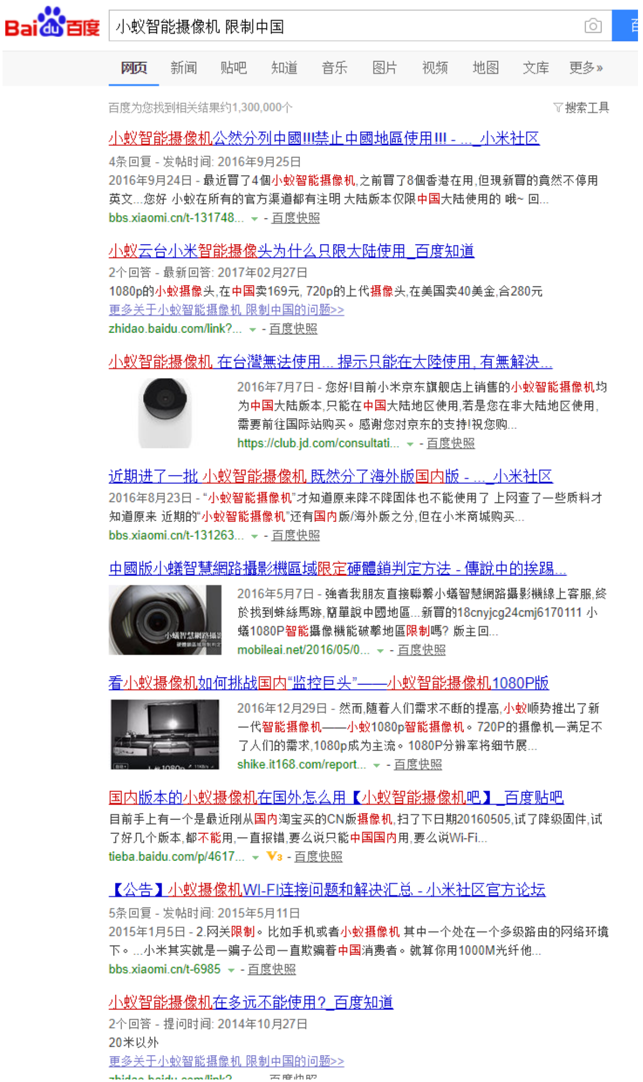
Only mainland China: how to unlock camera for EU? · Issue #123 ...
https://github.com/samtap/fang-hacks/issues/123 ▾
May 12, 2017 - ... is banned! How can I **update firmware** unlocking region ban? ... https://diy.2pmc.net/solved-xiaomi-xiao-**yi**-ant-home-**camera**-can-used-**china**/

Progress with xiaoyi ants yi 1080p home camera, not version 2 · Issue ...
https://github.com/fritz-smh/yi-hack/issues/141 ▾
Feb 24, 2017 - Xiaomi **Yi** Ants **camera hack**. Contribute to ... Yi2 1080p **camera** doesn't work anymore outside of **China** :(#9 · @xcray .... lost telnet & ftp upon **upgrade** to 2.1.0.0A_201703071456 **firmware** xmflsct/**yi-hack**-1080p#5 · @xmflsct ...

[HELP] Xiaomi Yi Ip night serial 17CN "only be used within china ...
en.miui.com › Devices › Mi Gadgets ▾
Sep 6, 2016 - 7 posts - 5 authors
is there any way to downgrade **firmware** for CN17 because i buy many ... try this [MIUI DEVICE TEAM] **YI** IP **CAM China** Only Error After **Update**
You've visited this page 2 times. Last visit: 12/2/16

Change Xiaomi Yi 4K Action Camera Firmware from Chinese to ...
tectogizmo.com/change-xiaomi-yi-4k-firmware-from-chinese-to-english/ ▾
Jan 2, 2017 - See if it starts with Z16V12L or Z16V13L as the **update** is intended for these 2 models. Change Xiaomi **Yi** 4K **Firmware** from **Chinese** to English ...

Xiaomi Xiao Yi Ant HOME - This camera can only be used in China
https://diy.2pmc.net/solved-xiaomi-xiao-yi-ant-home-camera-can-used-china/ ▾
May 3, 2016 - Recently I bought a Xiaomi Xiao **Yi** (IP) **camera** (also known as **Yi** Home), ... I was hoping a **firmware upgrade** would solve this issue so I have ...



百度为您找到相关结果约1,300,000个

小蚁智能摄像机公然分列中國!!!禁止中國地區使用!!! - ..._小米社区
4条回复 - 发帖时间: 2016年9月25日
2016年9月24日 - 最近買了4個小蚁智能摄像机,之前買了8個香港在用,但現新買的竟然不停用英文...您好 小蚁在所有的官方渠道都有注明 大陆版本仅限中国大陆使用的 哦~ 回...
bbs.xiaomi.cn/t-131748... ▾ - 百度快照

小蚁云台小米智能摄像头为什么只限大陆使用_百度知道
2个回答 - 最新回答: 2017年02月27日
1080p的小蚁摄像头,在中国卖169元, 720p的上代摄像头,在美国卖40美金,合280元
更多关于小蚁智能摄像机 限制中国的问题>>
zhidao.baidu.com/link?... ▾ - 百度快照

小蚁智能摄像机 在台灣無法使用... 提示只能在大陸使用, 有無解決...
2016年7月7日 - 您好!目前小米京东旗舰店上销售的小蚁智能摄像机均为中国大陆版本,只能在中国大陆地区使用,若是您在非大陆地区使用,需要前往国际站购买。感谢您对京东的支持!祝您购...
https://club.jd.com/consultati... ▾ - 百度快照

近期进了一批 小蚁智能摄像机 既然分了海外版国内版 - ..._小米社区
2016年8月23日 - “小蚁智能摄像机”才知道原来降不降固体也不能使用了 上网查了一些质料才知道原来 近期的“小蚁智能摄像机”还有国内版/海外版之分,但在小米商城购买...
bbs.xiaomi.cn/t-131263... ▾ - 百度快照

中國版小蟻智慧網路攝影機區域限定硬體鎖判定方法 - 傳說中的挨踢...
2016年5月7日 - 強者我朋友直接聯繫小蟻智慧網路攝影機線上客服,終於找到蛛絲馬跡,簡單說中國地區...新買的18cnyjcg24cmj6170111 小蟻1080P智能攝像機能破擊地區限制嗎? 版主回...
mobileai.net/2016/05/0... ▾ - 百度快照

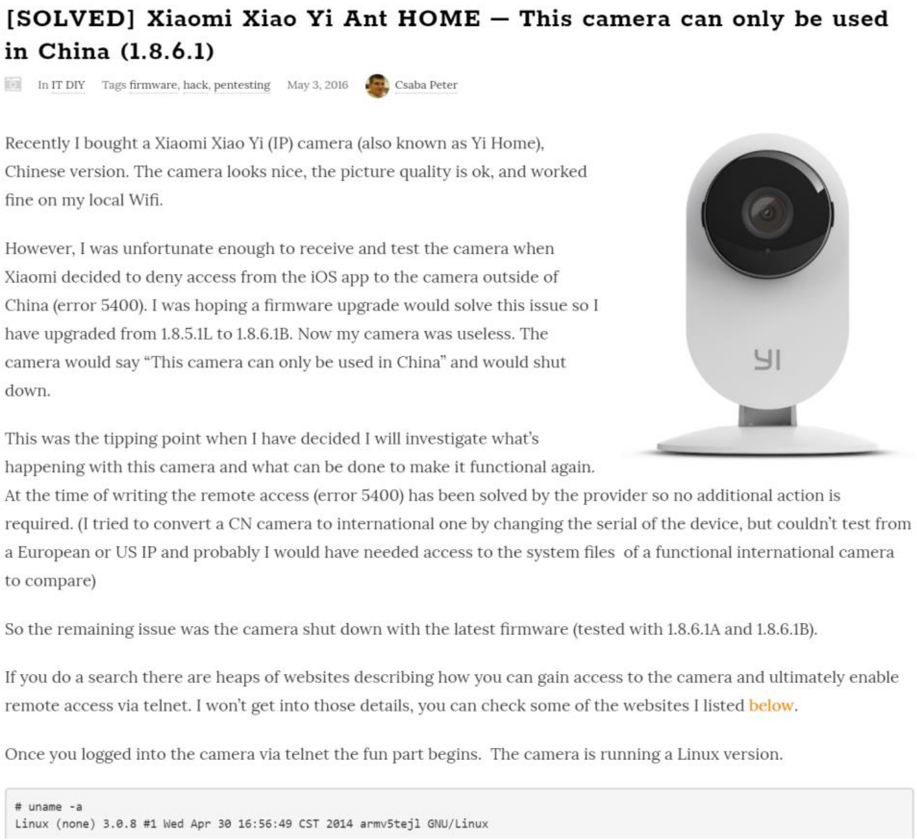
看小蚁摄像机如何挑战国内“监控巨头”——小蚁智能摄像机1080P版
2016年12月29日 - 然而,随着人们需求不断的提高,小蚁顺势推出了新一代智能摄像机——小蚁1080p智能摄像机。720P的摄像机一满足不了人们的需求,1080p成为主流。1080P分辨率将细节展...
shike.it168.com/report... ▾ - 百度快照

国内版本的小蚁摄像机在国外怎么用【小蚁智能摄像机吧】_百度贴吧
目前手上有一个是最近刚从国内淘宝买的CN版摄像机,扫了下日期20160505,试了降级固件,试了好几个版本,都不能用,一直报错,要么说只能中国国内用,要么说Wi-Fi...
tieba.baidu.com/p/4617... ▾ V3 - 百度快照

【公告】小蚁摄像机WI-FI连接问题和解决汇总 - 小米社区官方论坛
5条回复 - 发帖时间: 2015年5月11日
2015年1月5日 - 2.网关限制。比如手机或者小蚁摄像机 其中一个处在一个多级路由的网络环境下。...小米其实就是一骗子公司一直欺骗着中国消费者。就算你用1000M光纤他...
bbs.xiaomi.cn/t-6985 ▾ - 百度快照

小蚁智能摄像机在多远不能使用?_百度知道
2个回答 - 提问时间: 2014年10月27日
20米以外
更多关于小蚁智能摄像机 限制中国的问题>>
zhidao.baidu.com/link?... - 百度快照

Hacking Started



[SOLVED] Xiaomi Xiao Yi Ant HOME — This camera can only be used in China (1.8.6.1)

In IT DIY Tags firmware, hack, pentesting May 3, 2016 Csaba Peter

Recently I bought a Xiaomi Xiao Yi (IP) camera (also known as Yi Home), Chinese version. The camera looks nice, the picture quality is ok, and worked fine on my local Wifi.

However, I was unfortunate enough to receive and test the camera when Xiaomi decided to deny access from the iOS app to the camera outside of China (error 5400). I was hoping a firmware upgrade would solve this issue so I have upgraded from 1.8.5.1L to 1.8.6.1B. Now my camera was useless. The camera would say "This camera can only be used in China" and would shut down.

This was the tipping point when I have decided I will investigate what's happening with this camera and what can be done to make it functional again. At the time of writing the remote access (error 5400) has been solved by the provider so no additional action is required. (I tried to convert a CN camera to international one by changing the serial of the device, but couldn't test from a European or US IP and probably I would have needed access to the system files of a functional international camera to compare)

So the remaining issue was the camera shut down with the latest firmware (tested with 1.8.6.1A and 1.8.6.1B).

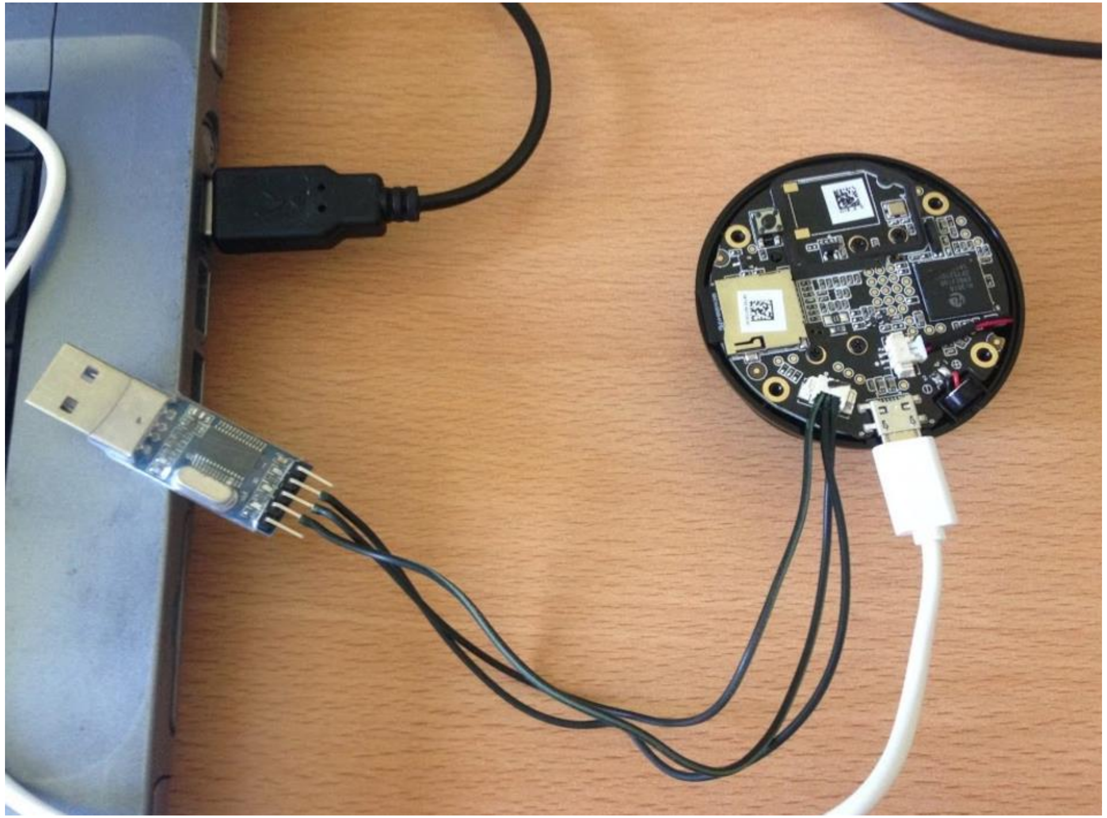
If you do a search there are heaps of websites describing how you can gain access to the camera and ultimately enable remote access via telnet. I won't get into those details, you can check some of the websites I listed below.

Once you logged into the camera via telnet the fun part begins. The camera is running a Linux version.

```
# uname -a
Linux (none) 3.0.8 #1 Wed Apr 30 16:56:49 CST 2014 armv5tejl GNU/Linux
```

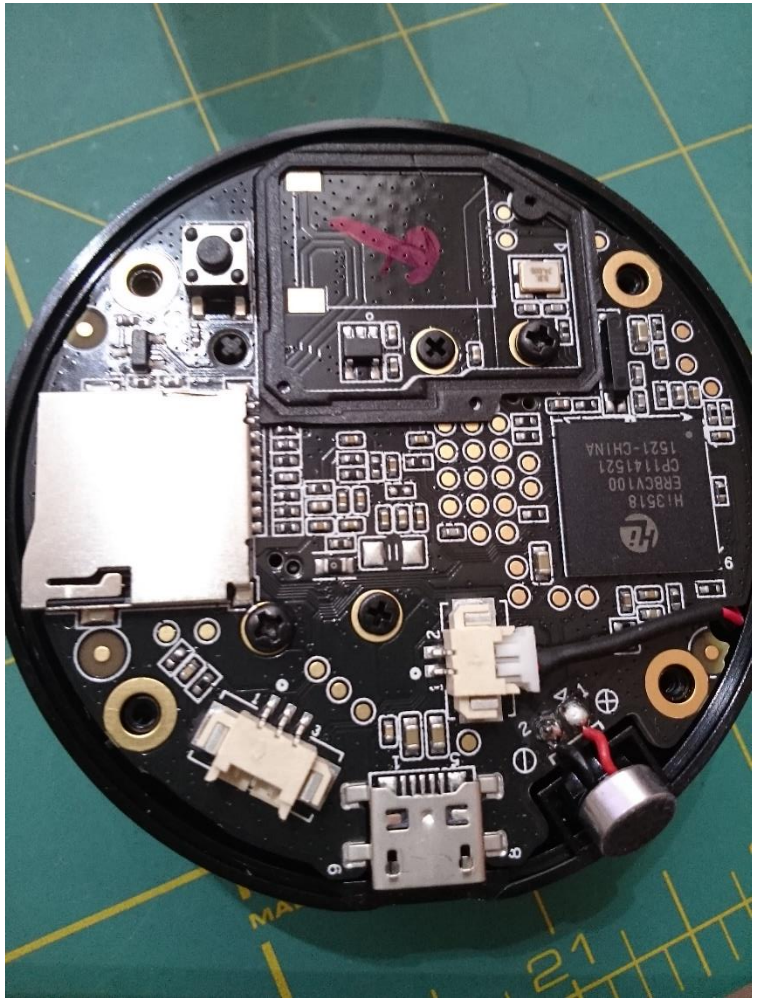
- 17CN 1.8.6.1R_201611191201
- Not downgrade able
- Not down gradable could be a bug

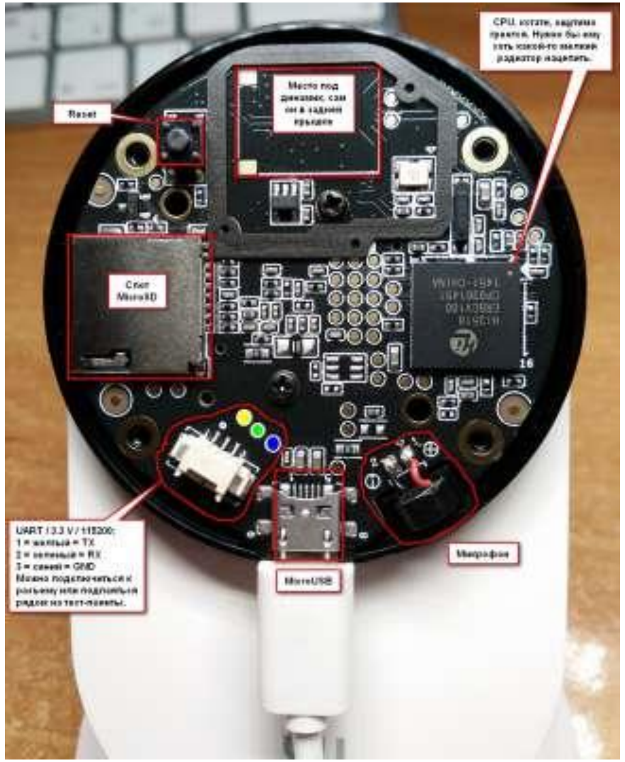
Try to Connect to USB TTL



- Power
- USB TTL
- No Way To Get Near USB TTL

Solving Puzzle

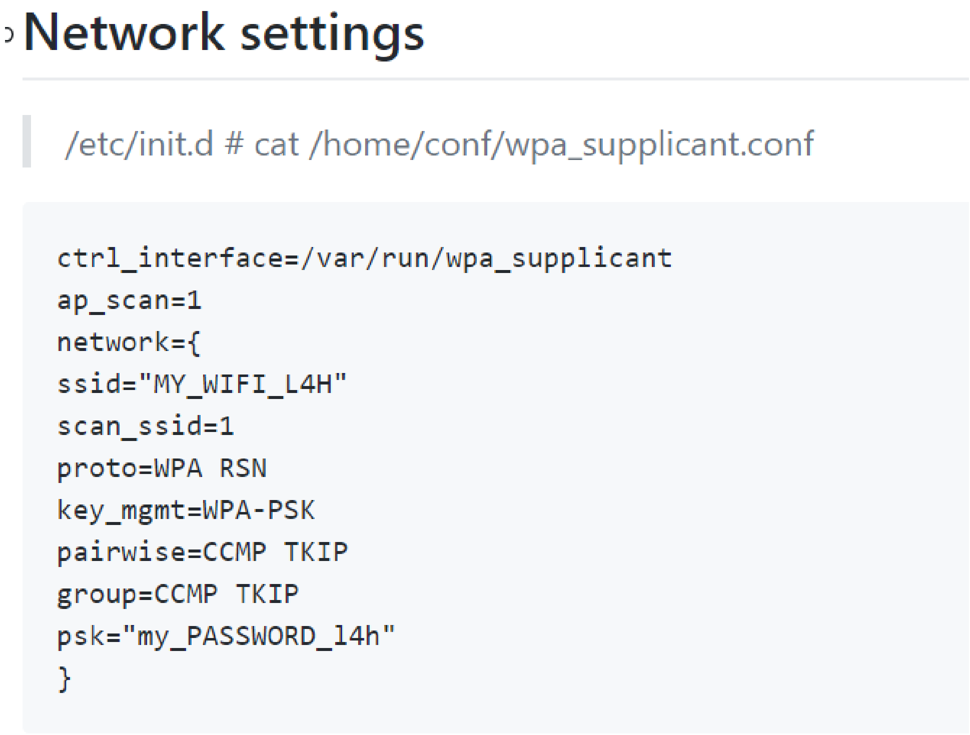




CPU, кстати, ощутимо греется. Нужно бы ему хоть какой-то маленький радиатор наклеить.
Reset
Место под динамик, сам он в задней крышке
Слот MicroSD
UART / 3.3 V / 115200:
1 = желтый = TX
2 = зеленый = RX
3 = синий = GND
Можно подключиться к разъему или подпаяться рядом на тест-поинты.
MicroUSB
Микрофон

- Finding GND
- Guessing RX TX
- Multi meter

What To We Want To Archive



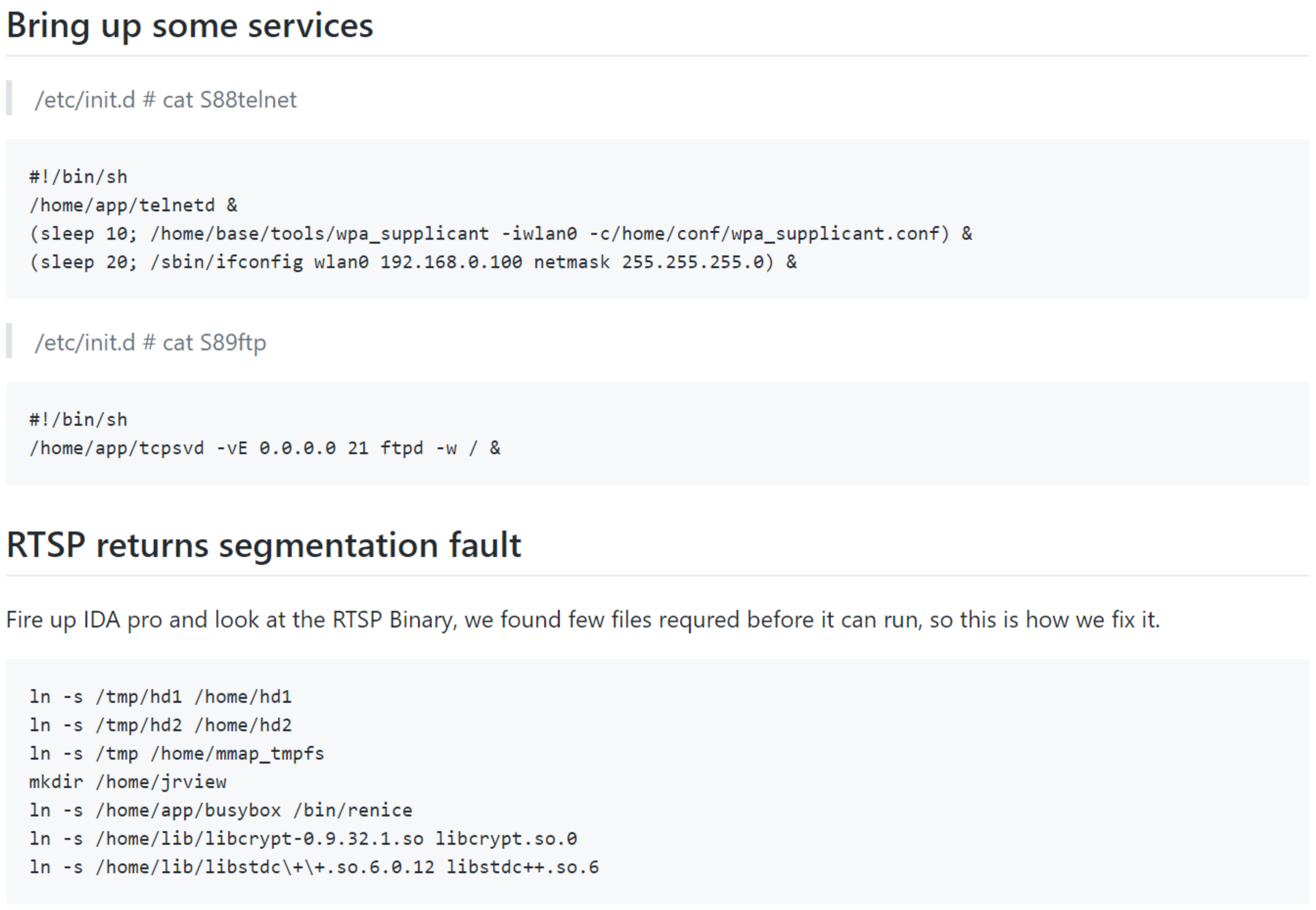
Network settings

> /etc/init.d # cat /home/conf/wpa_supplicant.conf

```
ctrl_interface=/var/run/wpa_supplicant
ap_scan=1
network={
ssid="MY_WIFI_L4H"
scan_ssid=1
proto=WPA RSN
key_mgmt=WPA-PSK
pairwise=CCMP TKIP
group=CCMP TKIP
psk="my_PASSWORD_l4h"
}
```

- Work without Xiaomi app
- Turn on WiFi while Boot
- Turn on telnet while boot
- Turn on ftp while boot
- Turn RTSP whole boot

Enabling Services



Bring up some services

> /etc/init.d # cat S88telnet

```
#!/bin/sh
/home/app/telnetd &
(sleep 10; /home/base/tools/wpa_supplicant -iwlan0 -c/home/conf/wpa_supplicant.conf) &
(sleep 20; /sbin/ifconfig wlan0 192.168.0.100 netmask 255.255.255.0) &
```

> /etc/init.d # cat S89ftp

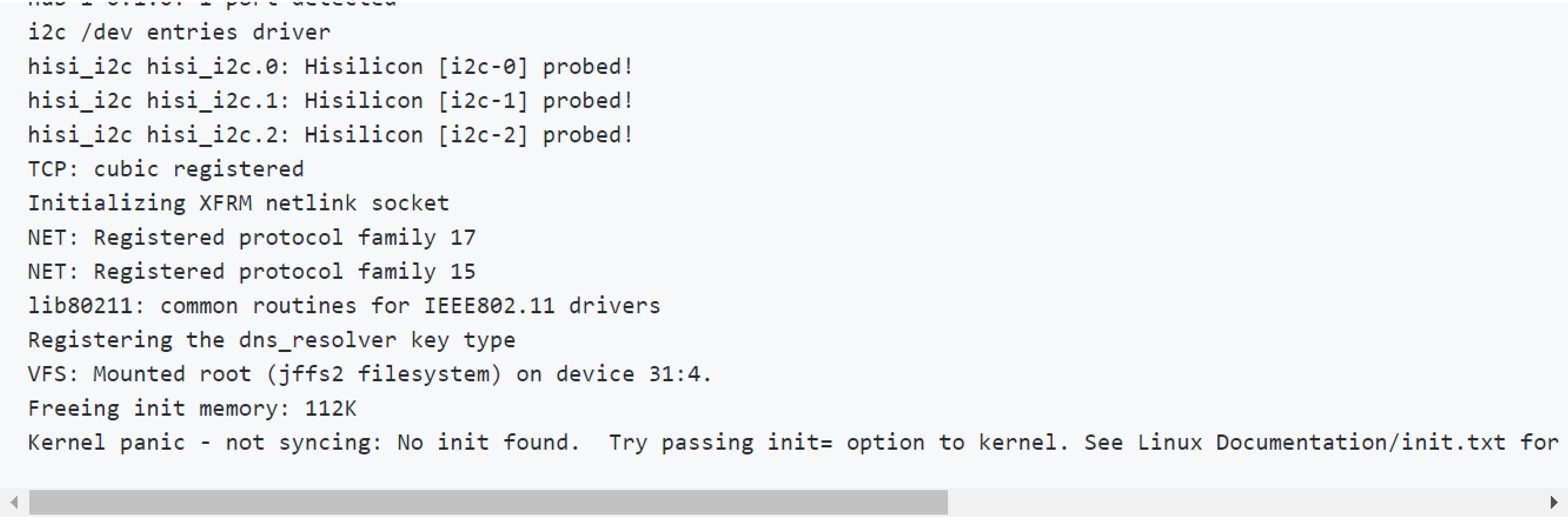
```
#!/bin/sh
/home/app/tcpsvd -vE 0.0.0.0 21 ftpd -w / &
```

RTSP returns segmentation fault

Fire up IDA pro and look at the RTSP Binary, we found few files requred before it can run, so this is how we fix it.

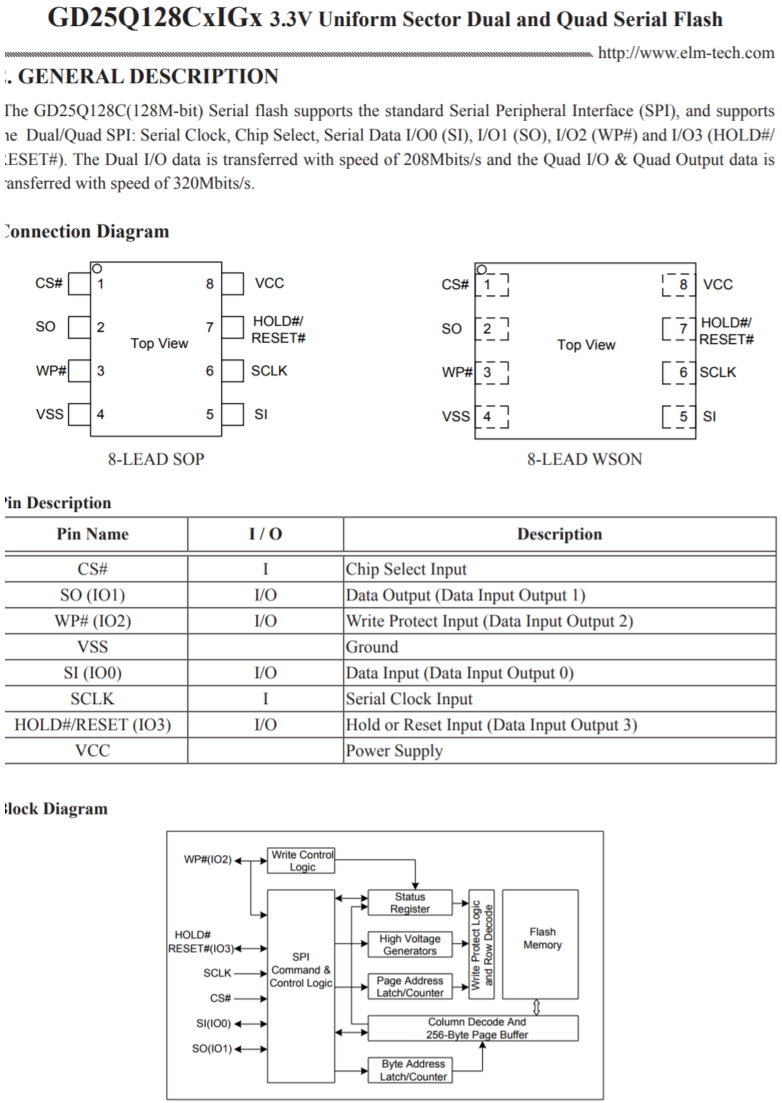
```
ln -s /tmp/hd1 /home/hd1
ln -s /tmp/hd2 /home/hd2
ln -s /tmp /home/mmap_tmpfs
mkdir /home/jrview
ln -s /home/app/busybox /bin/renice
ln -s /home/lib/libcrypt-0.9.32.1.so libcrypt.so.0
ln -s /home/lib/libstdc\+\+.so.6.0.12 libstdc++.so.6
```

Forgotten to mount FS after boot



```
hub 1-0:1.0: 1 port detected
i2c /dev entries driver
hisi_i2c hisi_i2c.0: Hisilicon [i2c-0] probed!
hisi_i2c hisi_i2c.1: Hisilicon [i2c-1] probed!
hisi_i2c hisi_i2c.2: Hisilicon [i2c-2] probed!
TCP: cubic registered
Initializing XFRM netlink socket
NET: Registered protocol family 17
NET: Registered protocol family 15
lib80211: common routines for IEEE802.11 drivers
Registering the dns_resolver key type
VFS: Mounted root (jffs2 filesystem) on device 31:4.
Freeing init memory: 112K
Kernel panic - not syncing: No init found.  Try passing init= option to kernel. See Linux Documentation/init.txt for
```

Back To Data Sheet



GD25Q128CxIGx 3.3V Uniform Sector Dual and Quad Serial Flash

http://www.elm-tech.com

. GENERAL DESCRIPTION

The GD25Q128C(128M-bit) Serial flash supports the standard Serial Peripheral Interface (SPI), and supports he Dual/Quad SPI: Serial Clock, Chip Select, Serial Data I/O0 (SI), I/O1 (SO), I/O2 (WP#) and I/O3 (HOLD#/ ESET#). The Dual I/O data is transferred with speed of 208Mbits/s and the Quad I/O & Quad Output data is ansferred with speed of 320Mbits/s.

onnection Diagram

8-LEAD SOP

8-LEAD WSON

in Description

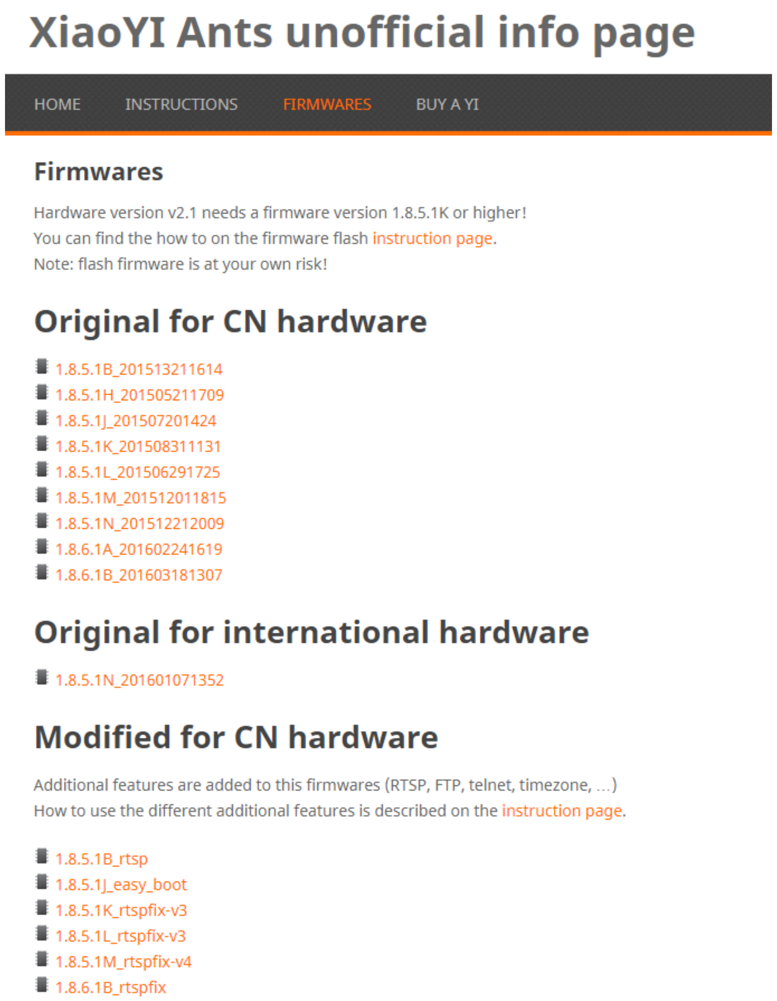
Pin Name	I / O	Description
CS#	I	Chip Select Input
SO (IO1)	I/O	Data Output (Data Input Output 1)
WP# (IO2)	I/O	Write Protect Input (Data Input Output 2)
VSS		Ground
SI (IO0)	I/O	Data Input (Data Input Output 0)
SCLK	I	Serial Clock Input
HOLD#/RESET (IO3)	I/O	Hold or Reset Input (Data Input Output 3)
VCC		Power Supply

lock Diagram

WP#(IO2)
Write Control Logic
Status Register
High Voltage Generators
Page Address Latch/Counter
Write Protect Logic and Row Decode
Flash Memory
HOLD# RESET#(IO3)
SCLK
CS#
SI(IO0)
SO(IO1)
SPI Command & Control Logic
Column Decode And 256-Byte Page Buffer
Byte Address Latch/Counter

- sdcard Is not readable while boot

Analyzing The Actual Firmware



XiaoYI Ants unofficial info page

Firmwares

Hardware version v2.1 needs a firmware version 1.8.5.1K or higher!
You can find the how to on the firmware flash instruction page.
Note: flash firmware is at your own risk!

Original for CN hardware

- 1.8.5.1B_201513211614
- 1.8.5.1H_201505211709
- 1.8.5.1J_201507201424
- 1.8.5.1K_201508311131
- 1.8.5.1L_201506291725
- 1.8.5.1M_201512011815
- 1.8.5.1N_201512212009
- 1.8.6.1A_201602241619
- 1.8.6.1B_201603181307

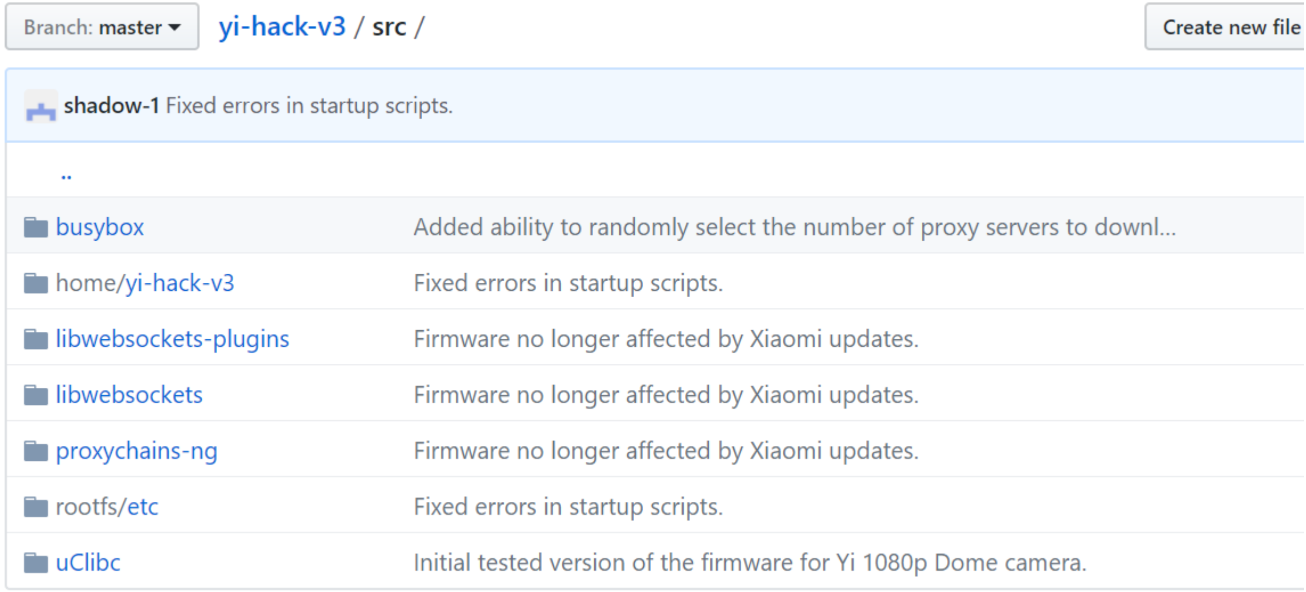
Original for international hardware

- 1.8.5.1N_201601071352

Modified for CN hardware

Additional features are added to this firmwares (RTSP, FTP, telnet, timezone, ...)
How to use the different additional features is described on the instruction page.

- 1.8.5.1B_rtsp
- 1.8.5.1J_easy_boot
- 1.8.5.1K_rtspfix-v3
- 1.8.5.1L_rtspfix-v3
- 1.8.5.1M_rtspfix-v4
- 1.8.6.1B_rtspfix

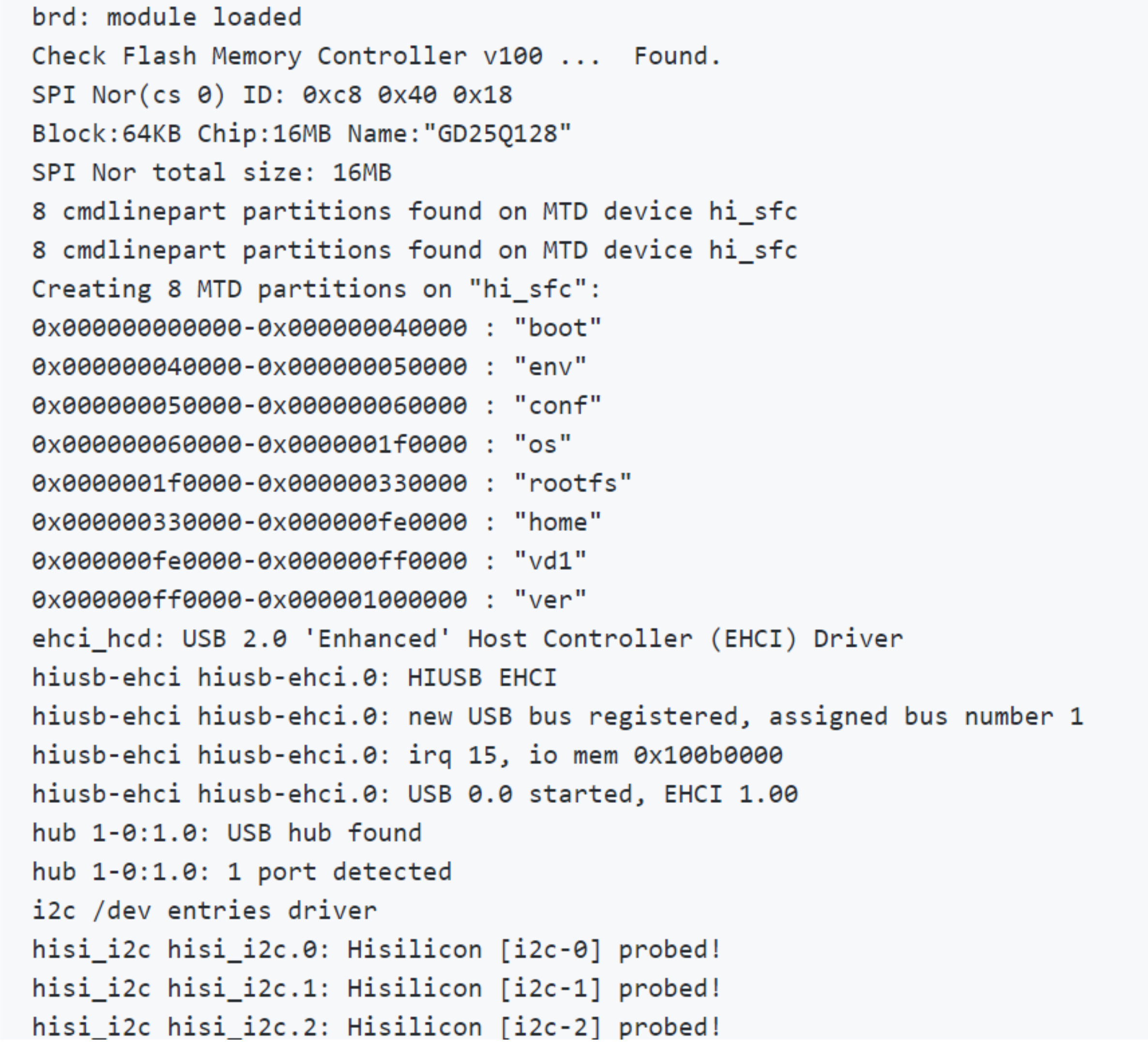


yi-hack-v3 / src /

shadow-1 Fixed errors in startup scripts.

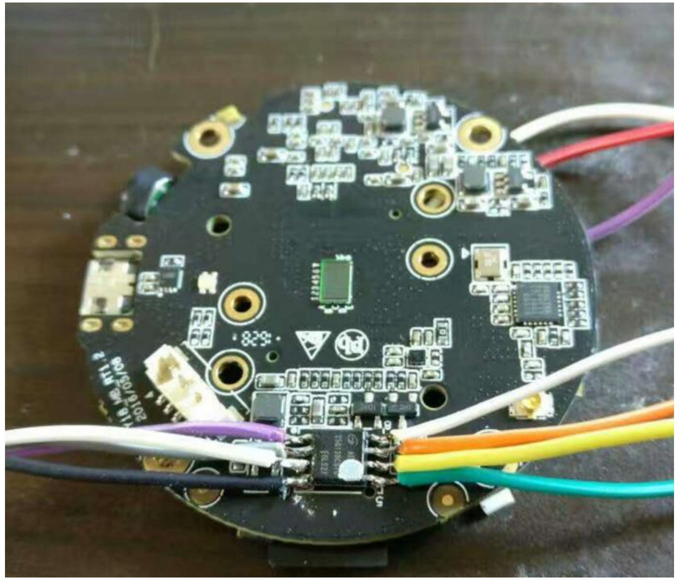
..	
busybox	Added ability to randomly select the number of proxy servers to downl...
home/yi-hack-v3	Fixed errors in startup scripts.
libwebsockets-plugins	Firmware no longer affected by Xiaomi updates.
libwebsockets	Firmware no longer affected by Xiaomi updates.
proxychains-ng	Firmware no longer affected by Xiaomi updates.
rootfs/etc	Fixed errors in startup scripts.
uClibc	Initial tested version of the firmware for Yi 1080p Dome camera.

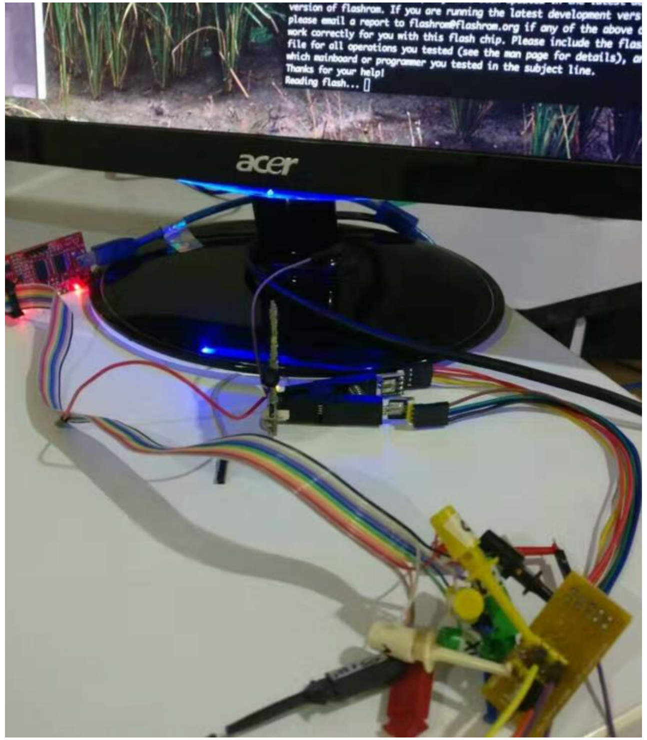
Understanding dmesg



```
brd: module loaded
Check Flash Memory Controller v100 ...  Found.
SPI Nor(cs 0) ID: 0xc8 0x40 0x18
Block:64KB Chip:16MB Name:"GD25Q128"
SPI Nor total size: 16MB
8 cmdlinepart partitions found on MTD device hi_sfc
8 cmdlinepart partitions found on MTD device hi_sfc
Creating 8 MTD partitions on "hi_sfc":
0x000000000000-0x000000040000 : "boot"
0x000000040000-0x000000050000 : "env"
0x000000050000-0x000000060000 : "conf"
0x000000060000-0x0000001f0000 : "os"
0x0000001f0000-0x000000330000 : "rootfs"
0x000000330000-0x000000fe0000 : "home"
0x000000fe0000-0x000000ff0000 : "vd1"
0x000000ff0000-0x000001000000 : "ver"
ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver
hiusb-ehci hiusb-ehci.0: HIUSB EHCI
hiusb-ehci hiusb-ehci.0: new USB bus registered, assigned bus number 1
hiusb-ehci hiusb-ehci.0: irq 15, io mem 0x100b0000
hiusb-ehci hiusb-ehci.0: USB 0.0 started, EHCI 1.00
hub 1-0:1.0: USB hub found
hub 1-0:1.0: 1 port detected
i2c /dev entries driver
hisi_i2c hisi_i2c.0: Hisilicon [i2c-0] probed!
hisi_i2c hisi_i2c.1: Hisilicon [i2c-1] probed!
hisi_i2c hisi_i2c.2: Hisilicon [i2c-2] probed!
```

Dumping The Firmware



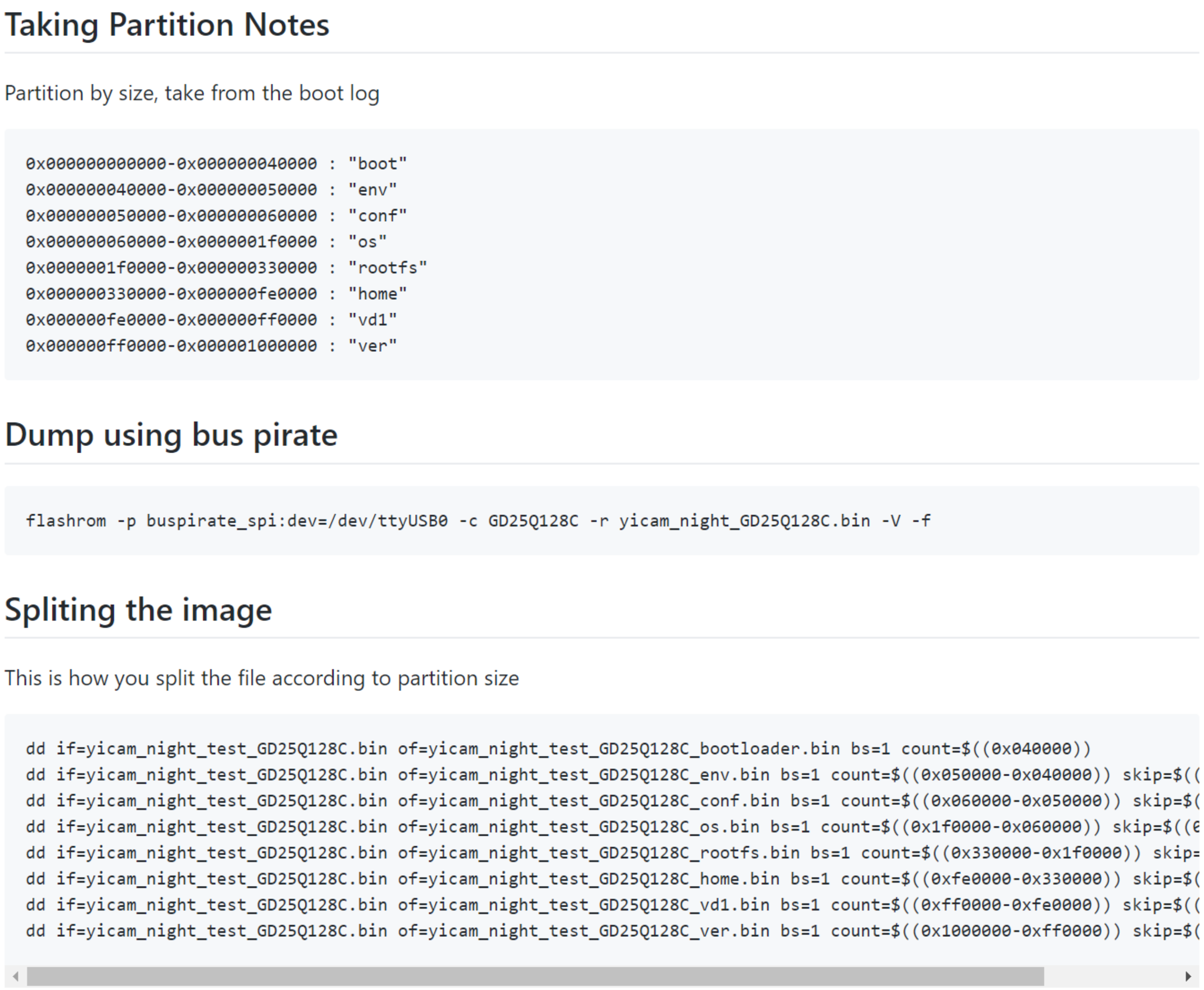


```
version of flashrom. If you are running the latest development vers
please email a report to flashrom@flashrom.org if any of the above c
work correctly for you with this flash chip. Please include the flas
file for all operations you tested (see the man page for details), an
which mainboard or programmer you tested in the subject line.
Thanks for your help!
Reading flash... ▯
```

- Making sure the firmware is the same with the one on the internet

Debug and Patch

Extract !



Taking Partition Notes

Partition by size, take from the boot log

```
0x000000000000-0x000000040000 : "boot"
0x000000040000-0x000000050000 : "env"
0x000000050000-0x000000060000 : "conf"
0x000000060000-0x0000001f0000 : "os"
0x0000001f0000-0x000000330000 : "rootfs"
0x000000330000-0x000000fe0000 : "home"
0x000000fe0000-0x000000ff0000 : "vd1"
0x000000ff0000-0x000001000000 : "ver"
```

Dump using bus pirate

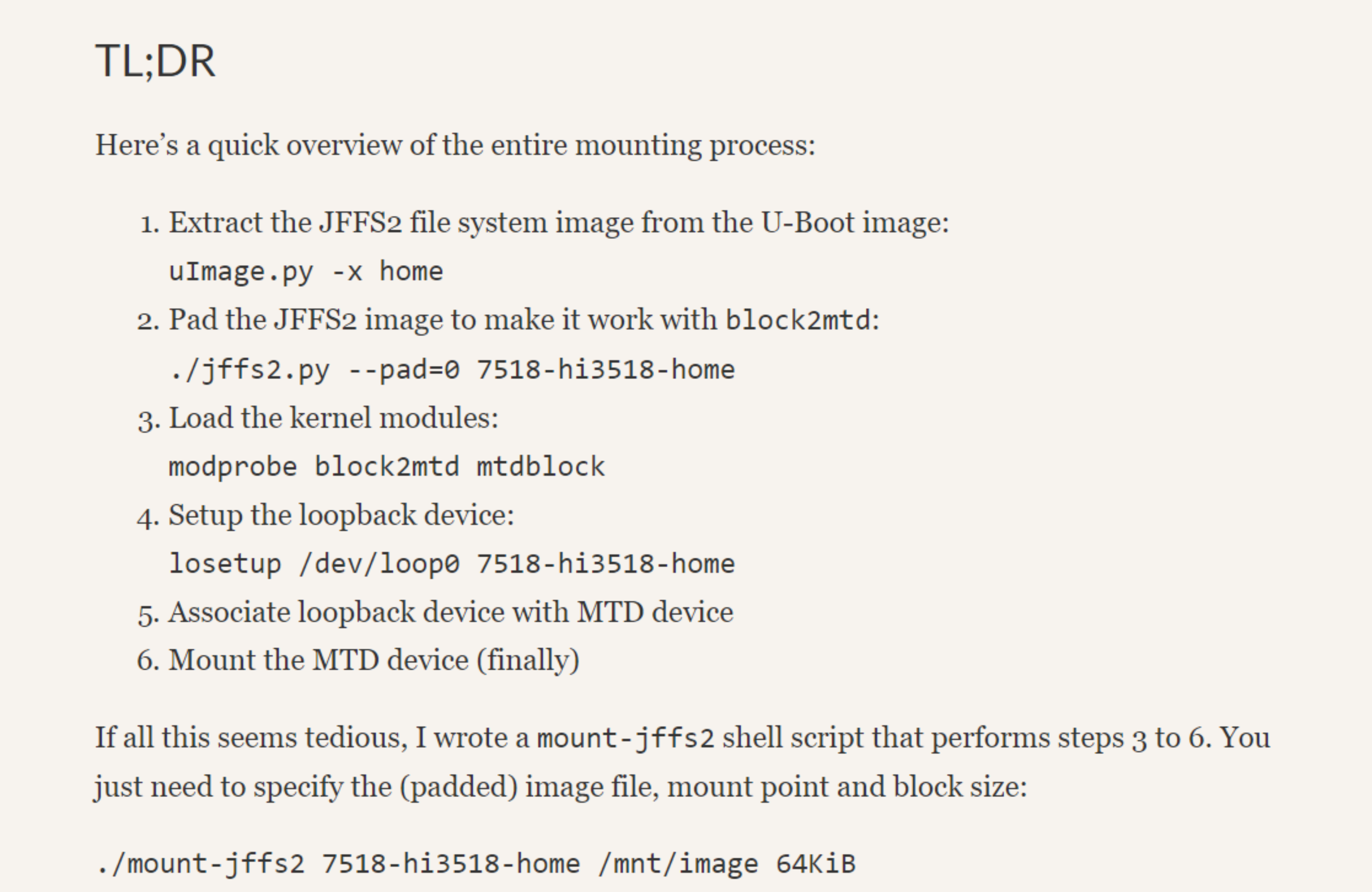
```
flashrom -p buspirate_spi:dev=/dev/ttyUSB0 -c GD25Q128C -r yicam_night_GD25Q128C.bin -V -f
```

Spliting the image

This is how you split the file according to partition size

```
dd if=yicam_night_test_GD25Q128C.bin of=yicam_night_test_GD25Q128C_bootloader.bin bs=1 count=$((0x040000))
dd if=yicam_night_test_GD25Q128C.bin of=yicam_night_test_GD25Q128C_env.bin bs=1 count=$((0x050000-0x040000)) skip=$((
dd if=yicam_night_test_GD25Q128C.bin of=yicam_night_test_GD25Q128C_conf.bin bs=1 count=$((0x060000-0x050000)) skip=$(
dd if=yicam_night_test_GD25Q128C.bin of=yicam_night_test_GD25Q128C_os.bin bs=1 count=$((0x1f0000-0x060000)) skip=$((
dd if=yicam_night_test_GD25Q128C.bin of=yicam_night_test_GD25Q128C_rootfs.bin bs=1 count=$((0x330000-0x1f0000)) skip=
dd if=yicam_night_test_GD25Q128C.bin of=yicam_night_test_GD25Q128C_home.bin bs=1 count=$((0xfe0000-0x330000)) skip=$(
dd if=yicam_night_test_GD25Q128C.bin of=yicam_night_test_GD25Q128C_vd1.bin bs=1 count=$((0xff0000-0xfe0000)) skip=$((
dd if=yicam_night_test_GD25Q128C.bin of=yicam_night_test_GD25Q128C_ver.bin bs=1 count=$((0x1000000-0xff0000)) skip=$(
```

Extract JFFS



TL;DR

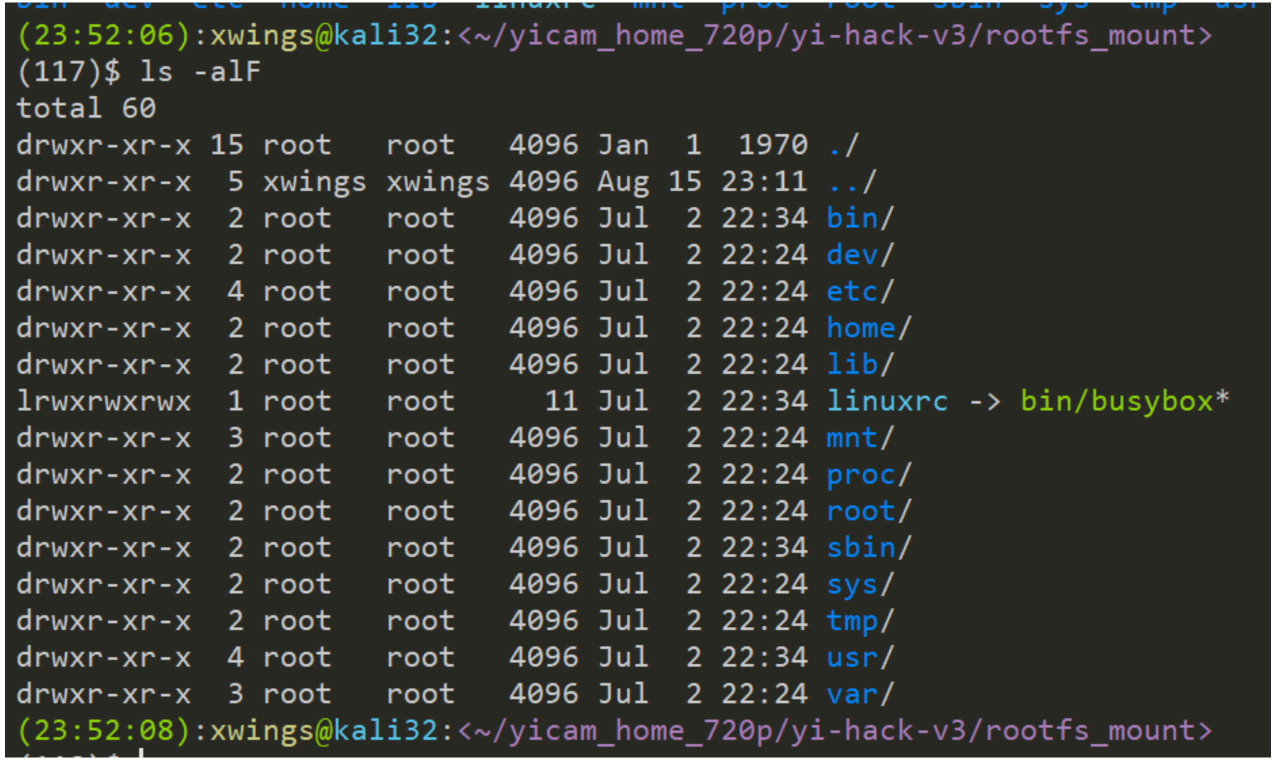
Here’s a quick overview of the entire mounting process:

1. Extract the JFFS2 file system image from the U-Boot image:
   ```
   uImage.py -x home
   ```
2. Pad the JFFS2 image to make it work with `block2mtd`:
   ```
   ./jffs2.py --pad=0 7518-hi3518-home
   ```
3. Load the kernel modules:
   ```
   modprobe block2mtd mtdblock
   ```
4. Setup the loopback device:
   ```
   losetup /dev/loop0 7518-hi3518-home
   ```
5. Associate loopback device with MTD device
6. Mount the MTD device (finally)

If all this seems tedious, I wrote a `mount-jffs2` shell script that performs steps 3 to 6. You just need to specify the (padded) image file, mount point and block size:

```
./mount-jffs2 7518-hi3518-home /mnt/image 64KiB
```

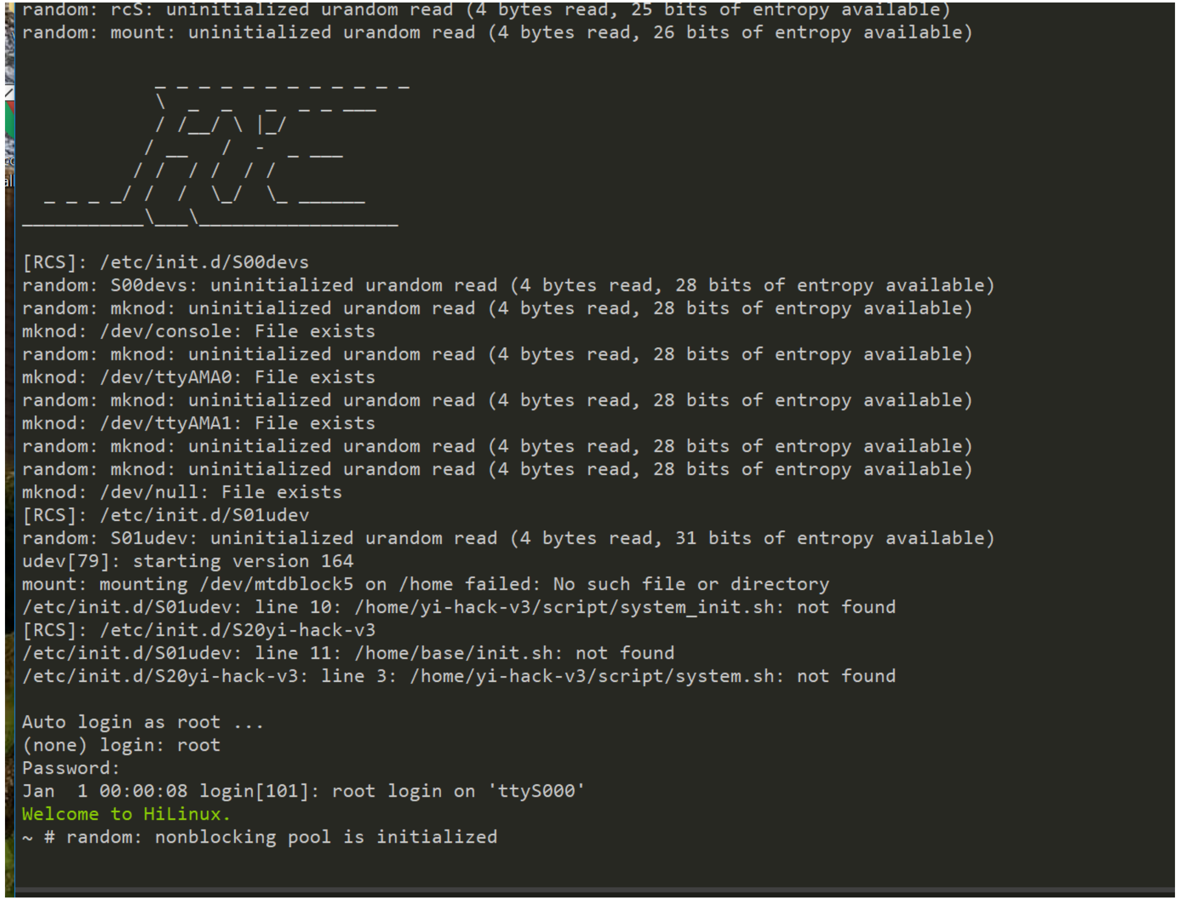
Making The Firmware



```
(23:52:06):xwings@kali32:<~/yicam_home_720p/yi-hack-v3/rootfs_mount>
(117)$ ls -alF
total 60
drwxr-xr-x 15 root   root   4096 Jan  1  1970 ./
drwxr-xr-x  5 xwings xwings 4096 Aug 15 23:11 ../
drwxr-xr-x  2 root   root   4096 Jul  2 22:34 bin/
drwxr-xr-x  2 root   root   4096 Jul  2 22:24 dev/
drwxr-xr-x  4 root   root   4096 Jul  2 22:24 etc/
drwxr-xr-x  2 root   root   4096 Jul  2 22:24 home/
drwxr-xr-x  2 root   root   4096 Jul  2 22:24 lib/
lrwxrwxrwx  1 root   root     11 Jul  2 22:34 linuxrc -> bin/busybox*
drwxr-xr-x  3 root   root   4096 Jul  2 22:24 mnt/
drwxr-xr-x  2 root   root   4096 Jul  2 22:24 proc/
drwxr-xr-x  2 root   root   4096 Jul  2 22:24 root/
drwxr-xr-x  2 root   root   4096 Jul  2 22:34 sbin/
drwxr-xr-x  2 root   root   4096 Jul  2 22:24 sys/
drwxr-xr-x  2 root   root   4096 Jul  2 22:24 tmp/
drwxr-xr-x  4 root   root   4096 Jul  2 22:34 usr/
drwxr-xr-x  3 root   root   4096 Jul  2 22:24 var/
(23:52:08):xwings@kali32:<~/yicam_home_720p/yi-hack-v3/rootfs_mount>
```

- # qemu-img create test.img 1024M
- # mkfs.ext2 –F test.img
- # mount –t ext2 –o loop,rw test.img /mnt/test
- Copy all files
- umount

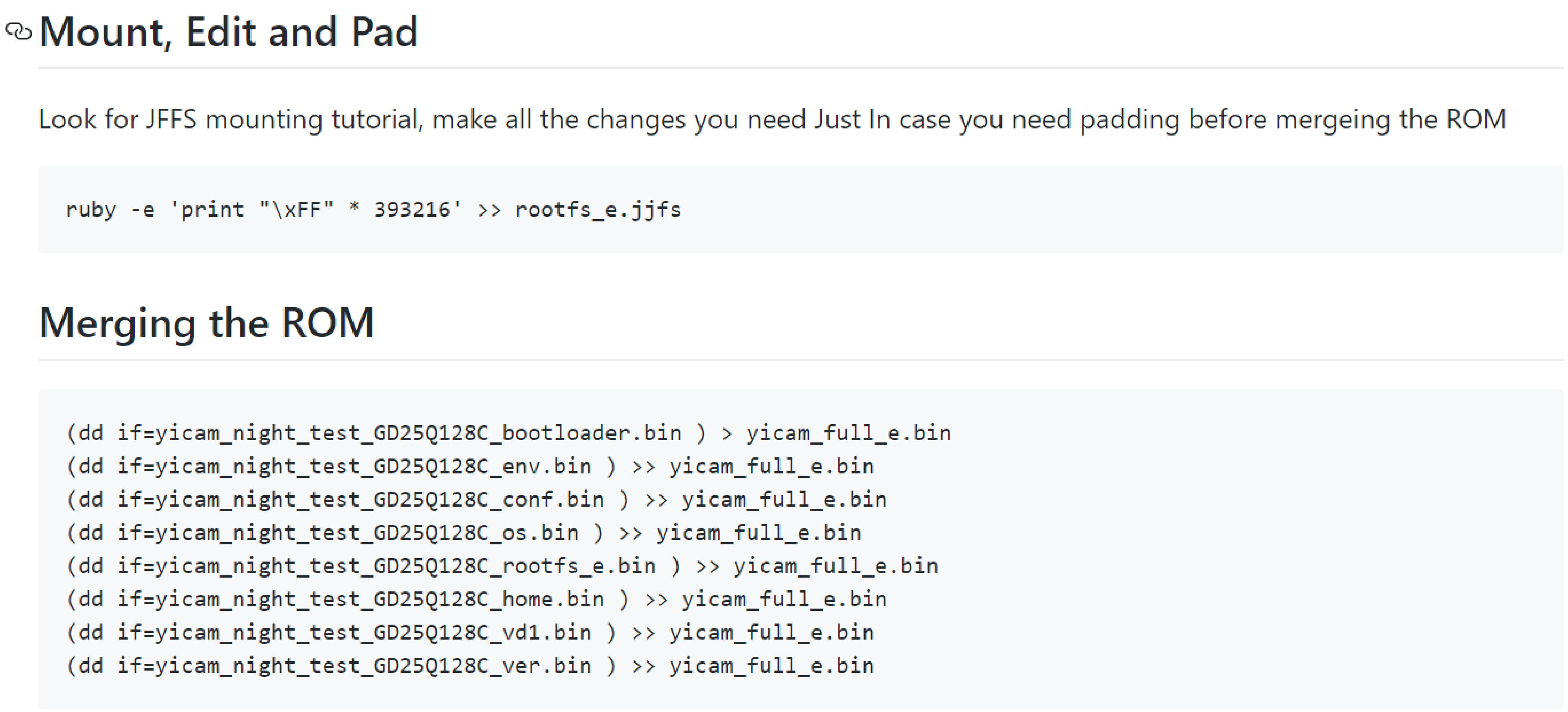
Test Booting with QEMU



```
random: rcS: uninitialized urandom read (4 bytes read, 25 bits of entropy available)
random: mount: uninitialized urandom read (4 bytes read, 26 bits of entropy available)


            _ _ _ _ _ _ _ _ _ _ _ _
            \  _  _   _  _ _ ___
            / /__/ \ |_/
           / __   /  -  _ ___
          / /  / /  / /
  _ _ _ _/ /  /  \_/  \_ _____
___________\___\_________________

[RCS]: /etc/init.d/S00devs
random: S00devs: uninitialized urandom read (4 bytes read, 28 bits of entropy available)
random: mknod: uninitialized urandom read (4 bytes read, 28 bits of entropy available)
mknod: /dev/console: File exists
random: mknod: uninitialized urandom read (4 bytes read, 28 bits of entropy available)
mknod: /dev/ttyAMA0: File exists
random: mknod: uninitialized urandom read (4 bytes read, 28 bits of entropy available)
mknod: /dev/ttyAMA1: File exists
random: mknod: uninitialized urandom read (4 bytes read, 28 bits of entropy available)
random: mknod: uninitialized urandom read (4 bytes read, 28 bits of entropy available)
mknod: /dev/null: File exists
[RCS]: /etc/init.d/S01udev
random: S01udev: uninitialized urandom read (4 bytes read, 31 bits of entropy available)
udev[79]: starting version 164
mount: mounting /dev/mtdblock5 on /home failed: No such file or directory
/etc/init.d/S01udev: line 10: /home/yi-hack-v3/script/system_init.sh: not found
[RCS]: /etc/init.d/S20yi-hack-v3
/etc/init.d/S01udev: line 11: /home/base/init.sh: not found
/etc/init.d/S20yi-hack-v3: line 3: /home/yi-hack-v3/script/system.sh: not found

Auto login as root ...
(none) login: root
Password:
Jan  1 00:00:08 login[101]: root login on 'ttyS000'
Welcome to HiLinux.
~ # random: nonblocking pool is initialized
```

- /home/xwings/qemu-2.9.0/arm-softmmu/qemu-system-arm -cpu arm1176 -M versatilepb -kernel /home/xwings/yicam_home_720p/testrun/kernel-qemu-4.4.34-jessie -append "console=ttyAMA0 root=/dev/sda rootfstype=ext2 rw" -hda /home/xwings/yicam_home_720p/yi-hack-v3/rootrootfs.img -nographic

Firmware Repacking



Mount, Edit and Pad

Look for JFFS mounting tutorial, make all the changes you need Just In case you need padding before mergeing the ROM

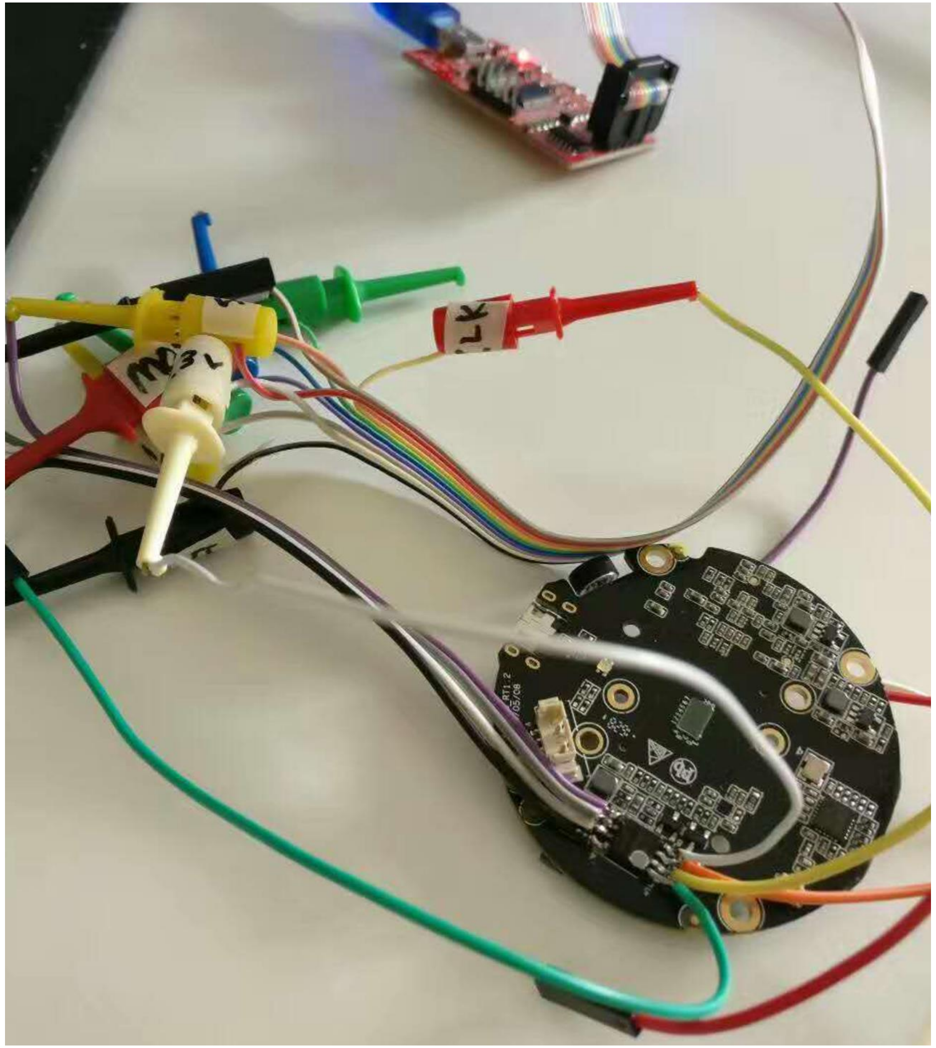
```
ruby -e 'print "\xFF" * 393216' >> rootfs_e.jjfs
```

Merging the ROM

```
(dd if=yicam_night_test_GD25Q128C_bootloader.bin ) > yicam_full_e.bin
(dd if=yicam_night_test_GD25Q128C_env.bin ) >> yicam_full_e.bin
(dd if=yicam_night_test_GD25Q128C_conf.bin ) >> yicam_full_e.bin
(dd if=yicam_night_test_GD25Q128C_os.bin ) >> yicam_full_e.bin
(dd if=yicam_night_test_GD25Q128C_rootfs_e.bin ) >> yicam_full_e.bin
(dd if=yicam_night_test_GD25Q128C_home.bin ) >> yicam_full_e.bin
(dd if=yicam_night_test_GD25Q128C_vd1.bin ) >> yicam_full_e.bin
(dd if=yicam_night_test_GD25Q128C_ver.bin ) >> yicam_full_e.bin
```

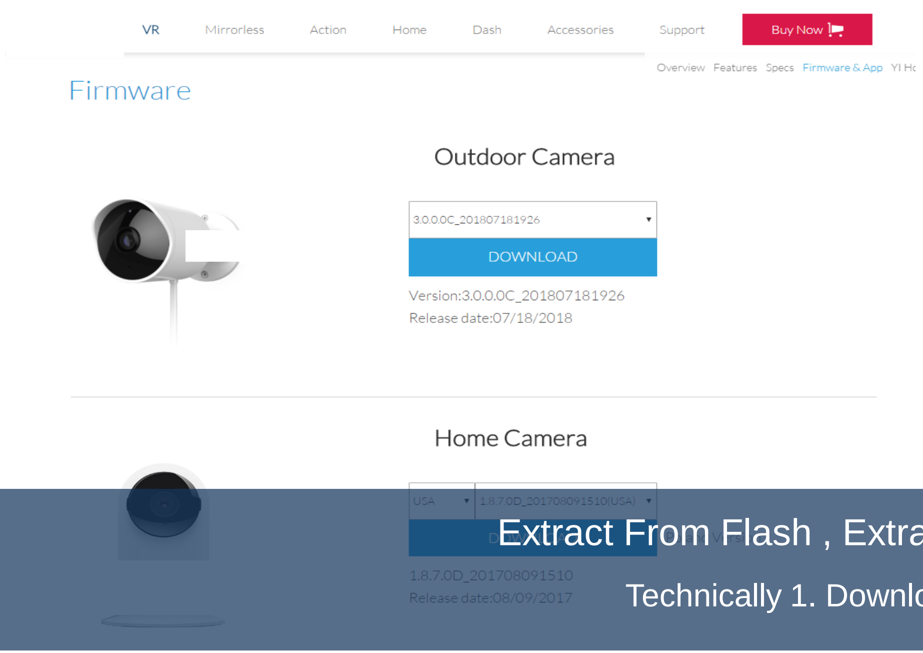
Seal

Flashing Back The Firmware



Getting Firmware

Firmware and Hardware



Firmware

Outdoor Camera

3.0.0.0C_201807181926

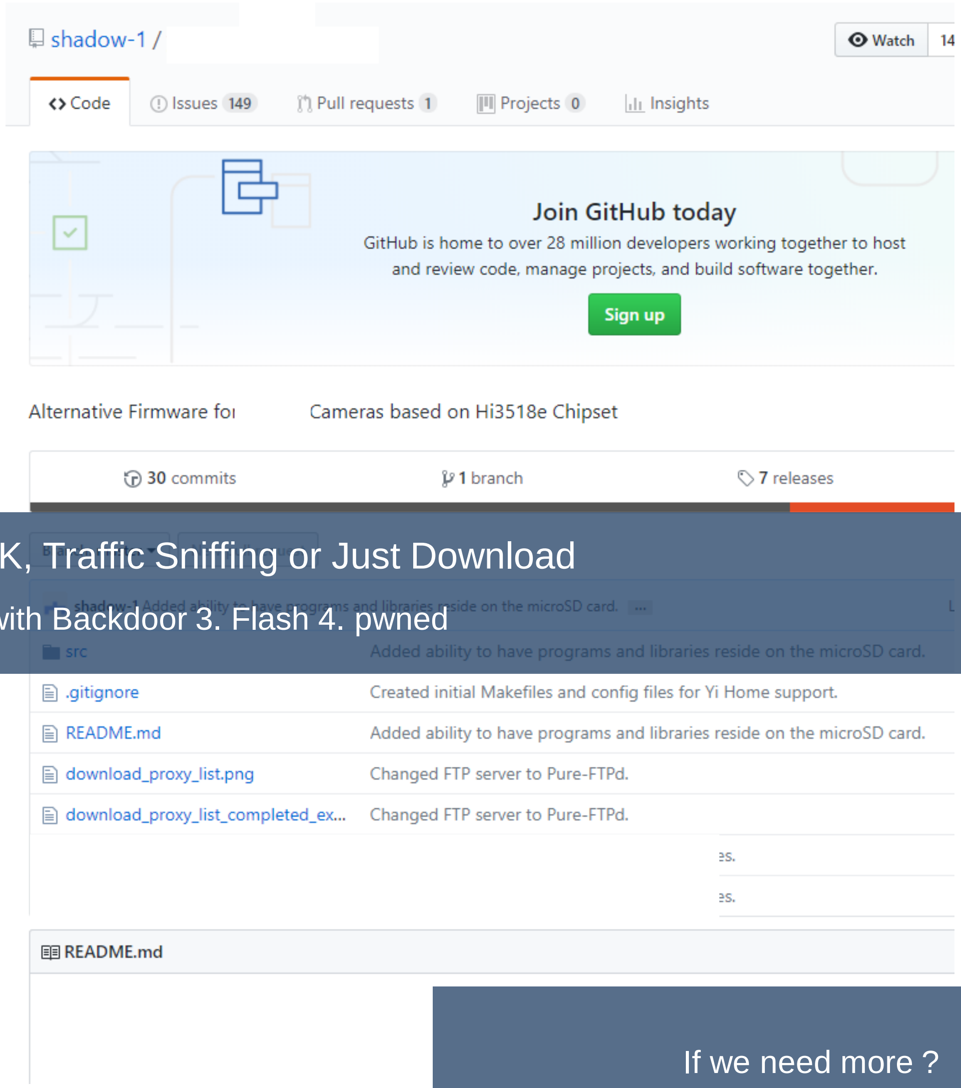
DOWNLOAD

Version:3.0.0.0C_201807181926
Release date:07/18/2018

Home Camera

USA | 1.8.7.0D_201708091510(USA)

1.8.7.0D_201708091510
Release date:08/09/2017



Join GitHub today

GitHub is home to over 28 million developers working together to host and review code, manage projects, and build software together.

Alternative Firmware for Cameras based on Hi3518e Chipset

30 commits | 1 branch | 7 releases

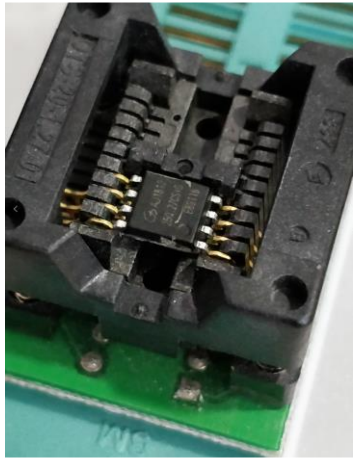
shadow-1 Added ability to have programs and libraries reside on the microSD card. ...

src	Added ability to have programs and libraries reside on the microSD card.
.gitignore	Created initial Makefiles and config files for Yi Home support.
README.md	Added ability to have programs and libraries reside on the microSD card.
download_proxy_list.png	Changed FTP server to Pure-FTPd.
download_proxy_list_completed_ex...	Changed FTP server to Pure-FTPd.
	es.
	es.

README.md

Extract From Flash , Extract From APK, Traffic Sniffing or Just Download

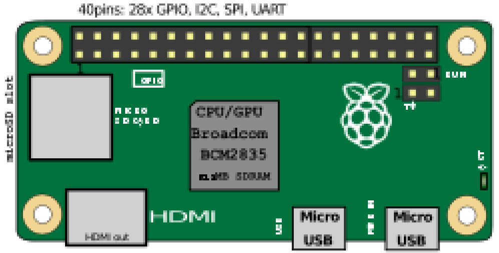
Technically 1. Download 2. Patch with Backdoor 3. Flash 4. pwned

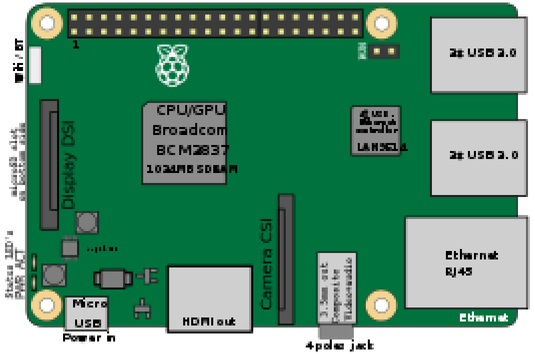


If we need more ?
1. RCE 2. Fuzz

Work Around

Complete Kit to Success





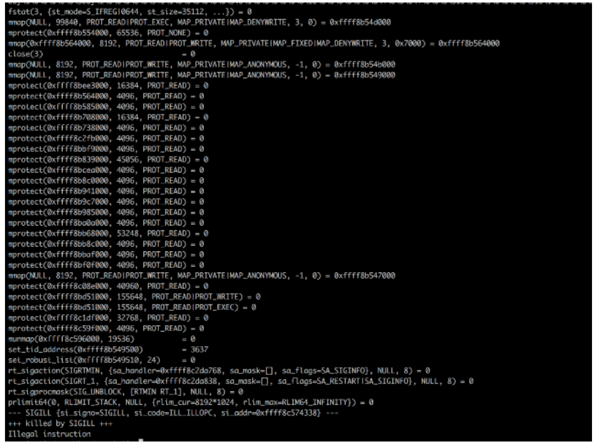
MIPS

ARM

AARCH64

How Many Dev Board

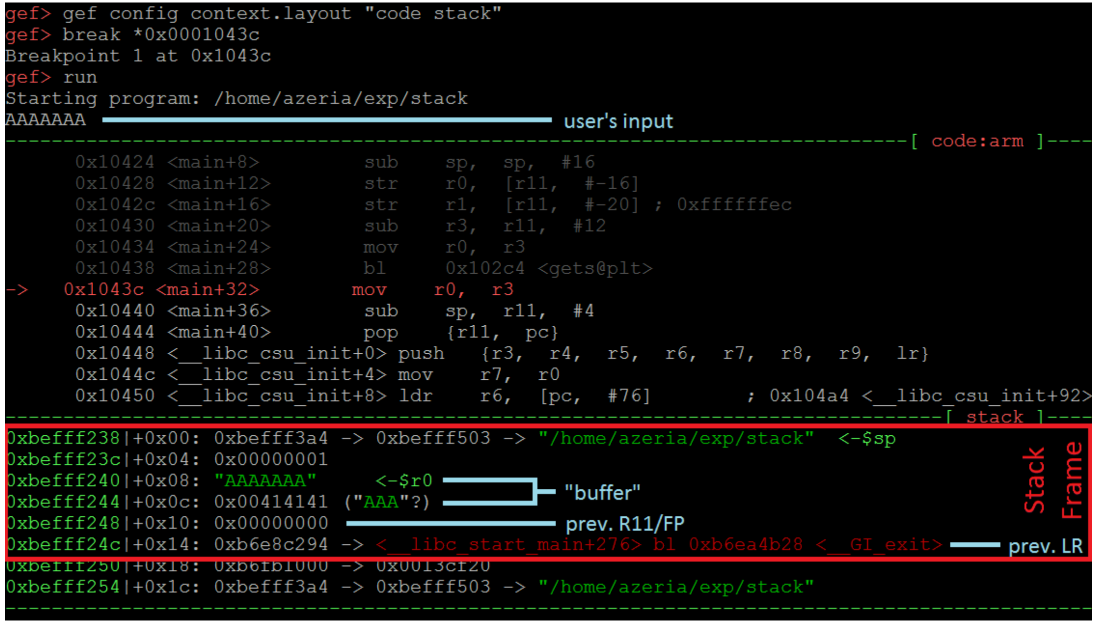
Classic LIBC Issue



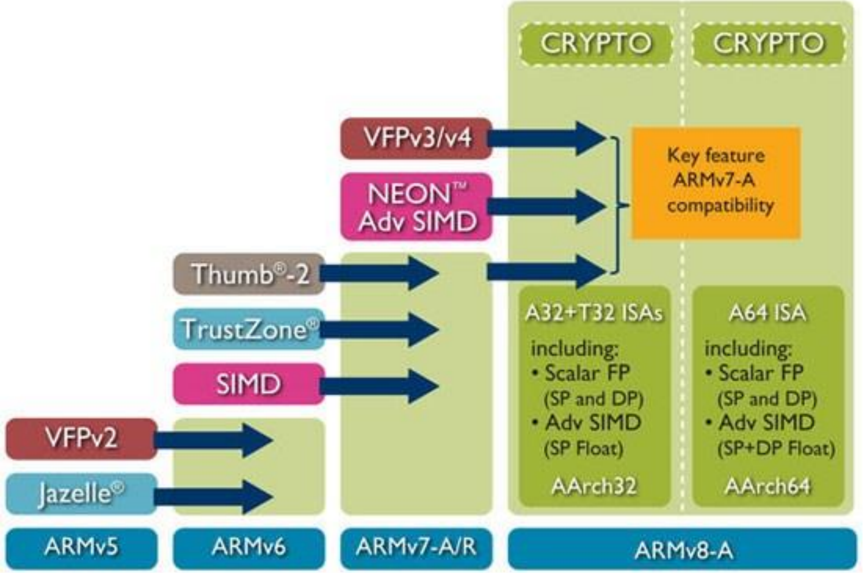
```
fstat(3, {st_mode=S_IFREG|0644, st_size=35112, ...}) = 0
mmap(NULL, 99840, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0xffff8b54d000
mprotect(0xffff8b554000, 65536, PROT_NONE) = 0
mmap(0xffff8b564000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x7000) = 0xffff8b564000
close(3)                                = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xffff8b54b000
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xffff8b549000
mprotect(0xffff8bee3000, 16384, PROT_READ) = 0
mprotect(0xffff8b564000, 4096, PROT_READ) = 0
mprotect(0xffff8b585000, 4096, PROT_READ) = 0
mprotect(0xffff8b708000, 16384, PROT_READ) = 0
mprotect(0xffff8b738000, 4096, PROT_READ) = 0
mprotect(0xffff8c2fb000, 4096, PROT_READ) = 0
mprotect(0xffff8bbf9000, 4096, PROT_READ) = 0
mprotect(0xffff8b839000, 45056, PROT_READ) = 0
mprotect(0xffff8bcea000, 4096, PROT_READ) = 0
mprotect(0xffff8b8c0000, 4096, PROT_READ) = 0
mprotect(0xffff8b941000, 4096, PROT_READ) = 0
mprotect(0xffff8b9c7000, 4096, PROT_READ) = 0
mprotect(0xffff8b985000, 4096, PROT_READ) = 0
mprotect(0xffff8ba0a000, 4096, PROT_READ) = 0
mprotect(0xffff8bb68000, 53248, PROT_READ) = 0
mprotect(0xffff8bb8c000, 4096, PROT_READ) = 0
mprotect(0xffff8bbaf000, 4096, PROT_READ) = 0
mprotect(0xffff8bf0f000, 4096, PROT_READ) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xffff8b547000
mprotect(0xffff8c08e000, 40960, PROT_READ) = 0
mprotect(0xffff8bd51000, 155648, PROT_READ|PROT_WRITE) = 0
mprotect(0xffff8bd51000, 155648, PROT_READ|PROT_EXEC) = 0
mprotect(0xffff8c1df000, 32768, PROT_READ) = 0
mprotect(0xffff8c59f000, 4096, PROT_READ) = 0
munmap(0xffff8c596000, 19536)           = 0
set_tid_address(0xffff8b549500)         = 3637
set_robust_list(0xffff8b549510, 24)     = 0
rt_sigaction(SIGRTMIN, {sa_handler=0xffff8c2da768, sa_mask=[], sa_flags=SA_SIGINFO}, NULL, 8) = 0
rt_sigaction(SIGRT_1, {sa_handler=0xffff8c2da838, sa_mask=[], sa_flags=SA_RESTART|SA_SIGINFO}, NULL, 8) = 0
rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
--- SIGILL {si_signo=SIGILL, si_code=ILL_ILLOPC, si_addr=0xffff8c574338} ---
+++ killed by SIGILL +++
Illegal instruction
```

Hardware is not “down gradable”

Assembly Instruction Compatibility



```
gef> gef config context.layout "code stack"
gef> break *0x0001043c
Breakpoint 1 at 0x1043c
gef> run
Starting program: /home/azeria/exp/stack
AAAAAAA
-------------------------------------------------------------------------------[ code:arm ]----
      0x10424 <main+8>         sub    sp,  sp,  #16
      0x10428 <main+12>        str    r0,  [r11,  #-16]
      0x1042c <main+16>        str    r1,  [r11,  #-20] ; 0xffffffec
      0x10430 <main+20>        sub    r3,  r11,  #12
      0x10434 <main+24>        mov    r0,  r3
      0x10438 <main+28>        bl     0x102c4 <gets@plt>
->   0x1043c <main+32>        mov    r0,  r3
      0x10440 <main+36>        sub    sp,  r11,  #4
      0x10444 <main+40>        pop    {r11,  pc}
      0x10448 <__libc_csu_init+0> push   {r3,  r4,  r5,  r6,  r7,  r8,  r9,  lr}
      0x1044c <__libc_csu_init+4> mov    r7,  r0
      0x10450 <__libc_csu_init+8> ldr    r6,  [pc,  #76]        ; 0x104a4 <__libc_csu_init+92>
--------------------------------------------------------------------------------[ stack ]----
0xbefff238|+0x00: 0xbefff3a4 -> 0xbefff503 -> "/home/azeria/exp/stack"  <-$sp
0xbefff23c|+0x04: 0x00000001
0xbefff240|+0x08: "AAAAAAA"     <-$r0
0xbefff244|+0x0c: 0x00414141 ("AAA"?)
0xbefff248|+0x10: 0x00000000
0xbefff24c|+0x14: 0xb6e8c294 -> <__libc_start_main+276> bl 0xb6ea4b28 <__GI_exit>
0xbefff250|+0x18: 0xb6fb1000 -> 0x0013cf20
0xbefff254|+0x1c: 0xbefff3a4 -> 0xbefff503 -> "/home/azeria/exp/stack"
-----------------------------------------------------------------------------------------------
```



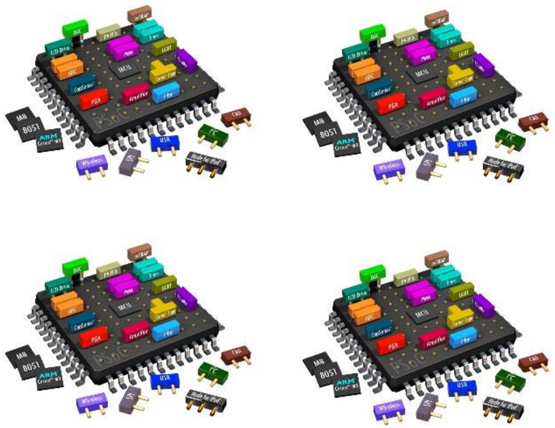
ARM

AARCH64

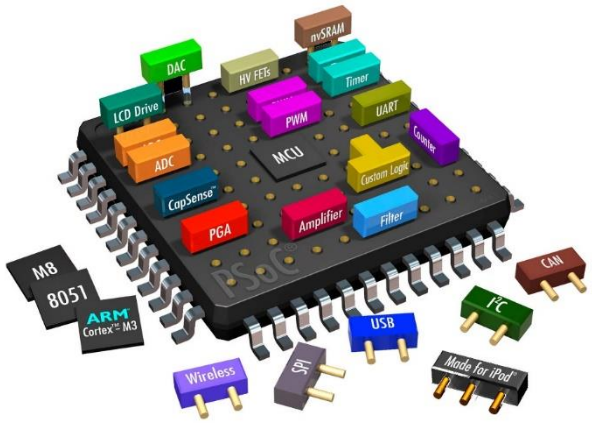
Why Firmware Emulation

More Resources = More Power

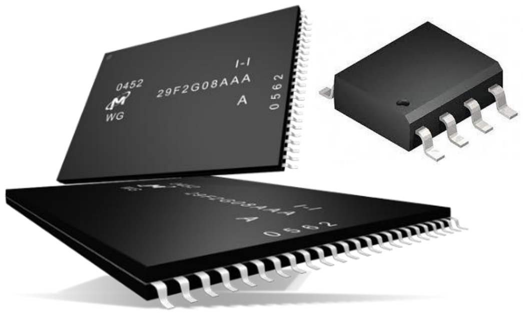
Multicore



MAX RAM



MAX Space



Processor

Normally 1-2 Core

RAM

Normally
256MB/512MB

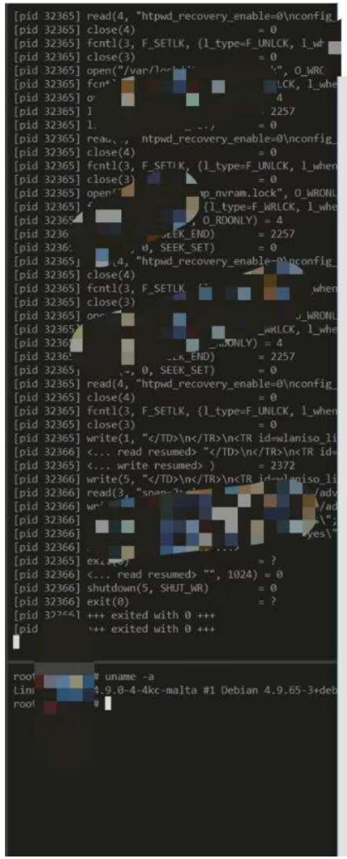
FLASH

Normally
8MB/16MB/32MB/256MB

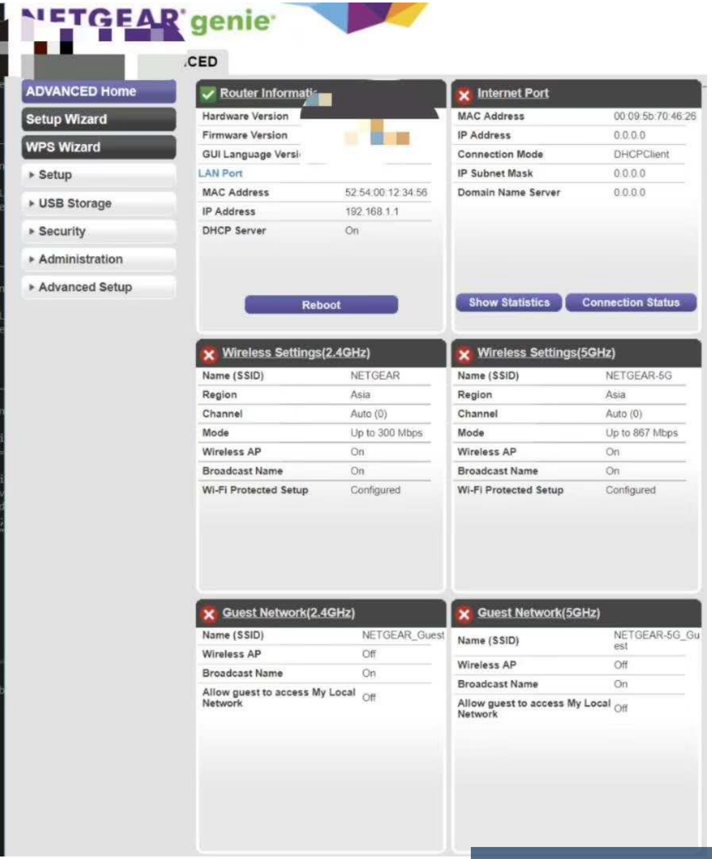
Most Important, we got apt-get

Objectives

Only One Process with Interaction



```
[pid 32365] read(4, "htpwd_recovery_enable=0\nconfig_
[pid 32365] close(4)                    = 0
[pid 32365] fcntl(3, F_SETLK, {l_type=F_UNLCK, l_wh
[pid 32365] close(3)                    = 0
[pid 32365] open("/var/lo[redacted]", O_WRC
[pid 32365] fcn[redacted]LCK, l_whe
[pid 32365] o[redacted]                   4
[pid 32365] l[redacted]                   2257
[pid 32365] l[redacted]                 = 0
[pid 32365] rea[redacted] htpwd_recovery_enable=0\nconfig_
[pid 32365] close(4)                    = 0
[pid 32365] fcntl(3, F_SETLK, {l_type=F_UNLCK, l_when
[pid 32365] close(3)                    = 0
[pid 32365] open[redacted]p_nvram.lock", O_WRONL
[pid 32365] [redacted] {l_type=F_WRLCK, l_whe
[pid 3236[redacted], O_RDONLY) = 4
[pid 3236[redacted]EEK_END)             = 2257
[pid 3236[redacted], 0, SEEK_SET)       = 0
[pid 32365] [redacted]4, "htpwd_recovery_enable=[redacted]nconfig_
[pid 32365] close(4)
[pid 32365] fcntl(3, F_SETLK [redacted] when
[pid 32365] close(3)
[pid 32365] ope[redacted]_WRONL
[pid 32365][redacted]_WRLCK, l_whe
[pid 3236[redacted]ONLY) = 4
[pid 3236[redacted]_END)                = 2257
[pid 32365][redacted], 0, SEEK_SET)     = 0
[pid 32365] read(4, "htpwd_recovery_enable=0\nconfig_
[pid 32365] close(4)                    = 0
[pid 32365] fcntl(3, F_SETLK, {l_type=F_UNLCK, l_when
[pid 32365] close(3)                    = 0
[pid 32365] write(1, "</TD>\n</TR>\n<TR id=wlaniso_li
[pid 32366] <... read resumed> "</TD>\n</TR>\n<TR id=
[pid 32365] <... write resumed> )       = 2372
[pid 32366] write(5, "</TD>\n</TR>\n<TR id=[redacted]aniso_li
[pid 32366] read(3, "snap-[redacted]/adv
[pid 32366] wr[redacted]/ad
[pid 32366] [redacted]\";
[pid 32366] [redacted]yes\"
[pid 32366] [redacted]
[pid 32365] ex[redacted]                = ?
[pid 32366] <... read resumed> "", 1024) = 0
[pid 32366] shutdown(5, SHUT_WR)        = 0
[pid 32366] exit(0)                     = ?
[pid 32366] +++ exited with 0 +++
[pid [redacted] +++ exited with 0 +++

root[redacted]# uname -a
Lin[redacted]4.9.0-4-4kc-malta #1 Debian 4.9.65-3+deb
root[redacted]#
```



Router Information

Hardware Version	
Firmware Version	
GUI Language Version	
LAN Port	
MAC Address	52:54:00:12:34:56
IP Address	192.168.1.1
DHCP Server	On

Reboot

Internet Port

MAC Address	00:09:5b:70:46:26
IP Address	0.0.0.0
Connection Mode	DHCPClient
IP Subnet Mask	0.0.0.0
Domain Name Server	0.0.0.0

Show Statistics | Connection Status

Wireless Settings(2.4GHz)

Name (SSID)	NETGEAR
Region	Asia
Channel	Auto (0)
Mode	Up to 300 Mbps
Wireless AP	On
Broadcast Name	On
Wi-Fi Protected Setup	Configured

Wireless Settings(5GHz)

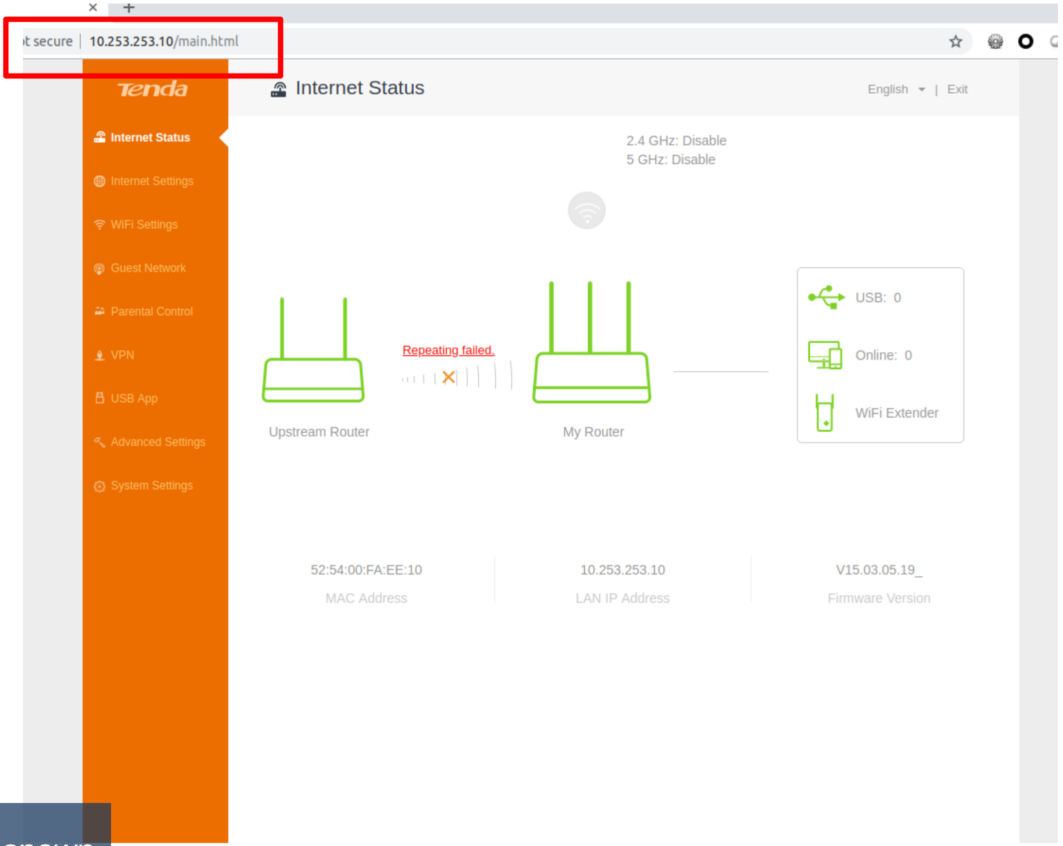
Name (SSID)	NETGEAR-5G
Region	Asia
Channel	Auto (0)
Mode	Up to 867 Mbps
Wireless AP	On
Broadcast Name	On
Wi-Fi Protected Setup	Configured

Guest Network(2.4GHz)

Name (SSID)	NETGEAR_Guest
Wireless AP	Off
Broadcast Name	On
Allow guest to access My Local Network	Off

Guest Network(5GHz)

Name (SSID)	NETGEAR-5G_Guest
Wireless AP	Off
Broadcast Name	On
Allow guest to access My Local Network	Off



t secure | 10.253.253.10/main.html

Internet Status

2.4 GHz: Disable
5 GHz: Disable

52:54:00:FA:EE:10	10.253.253.10	V15.03.05.19_
MAC Address	LAN IP Address	Firmware Version

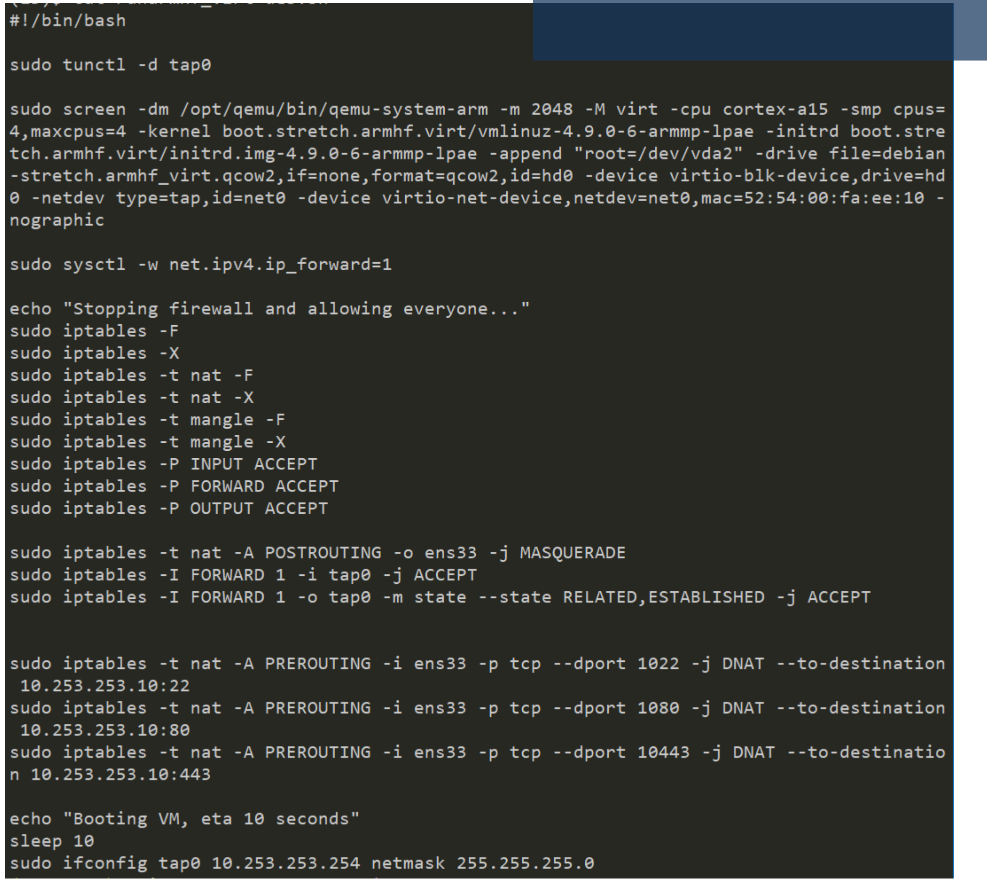
Hunt for the one that spawn listener port

most of the devices comes with one big binary

Boot

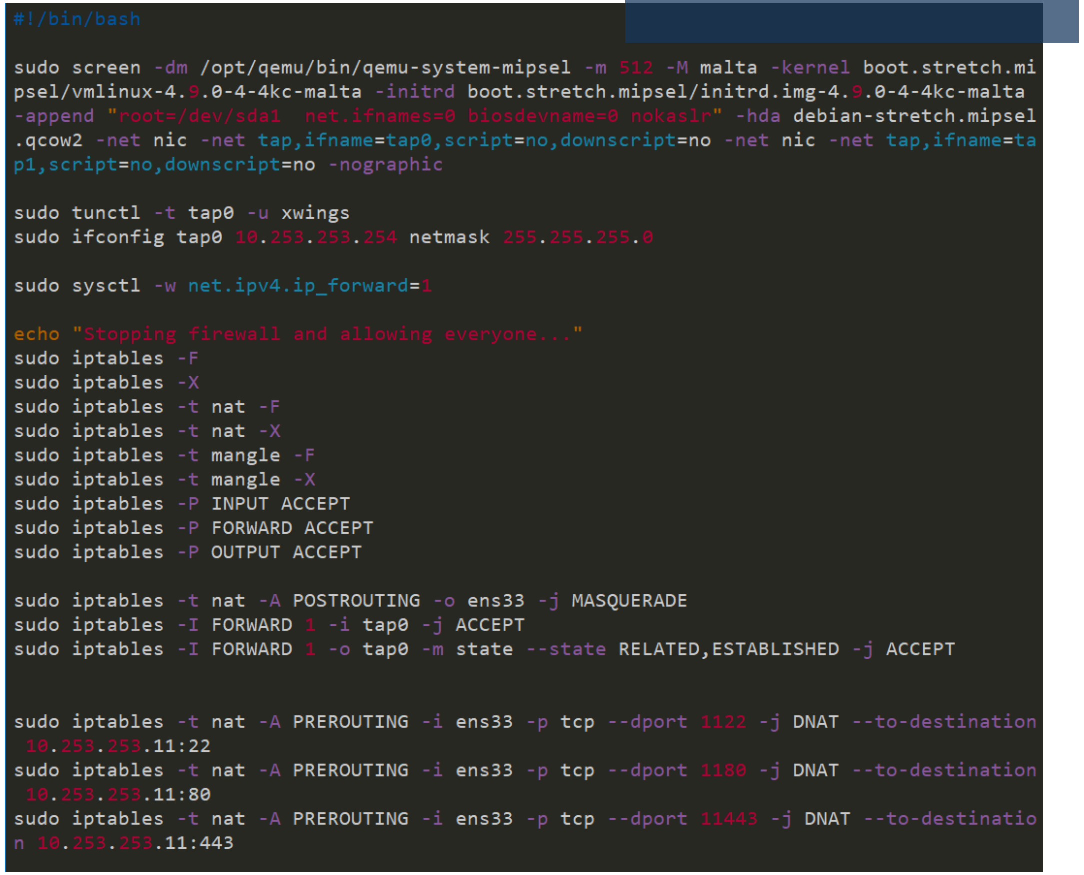
Distro and Kernel Mix and Match

script to boot arm



```
#!/bin/bash

sudo tunctl -d tap0

sudo screen -dm /opt/qemu/bin/qemu-system-arm -m 2048 -M virt -cpu cortex-a15 -smp cpus=
4,maxcpus=4 -kernel boot.stretch.armhf.virt/vmlinuz-4.9.0-6-armmp-lpae -initrd boot.stre
tch.armhf.virt/initrd.img-4.9.0-6-armmp-lpae -append "root=/dev/vda2" -drive file=debian
-stretch.armhf_virt.qcow2,if=none,format=qcow2,id=hd0 -device virtio-blk-device,drive=hd
0 -netdev type=tap,id=net0 -device virtio-net-device,netdev=net0,mac=52:54:00:fa:ee:10 -
nographic

sudo sysctl -w net.ipv4.ip_forward=1

echo "Stopping firewall and allowing everyone..."
sudo iptables -F
sudo iptables -X
sudo iptables -t nat -F
sudo iptables -t nat -X
sudo iptables -t mangle -F
sudo iptables -t mangle -X
sudo iptables -P INPUT ACCEPT
sudo iptables -P FORWARD ACCEPT
sudo iptables -P OUTPUT ACCEPT

sudo iptables -t nat -A POSTROUTING -o ens33 -j MASQUERADE
sudo iptables -I FORWARD 1 -i tap0 -j ACCEPT
sudo iptables -I FORWARD 1 -o tap0 -m state --state RELATED,ESTABLISHED -j ACCEPT


sudo iptables -t nat -A PREROUTING -i ens33 -p tcp --dport 1022 -j DNAT --to-destination
 10.253.253.10:22
sudo iptables -t nat -A PREROUTING -i ens33 -p tcp --dport 1080 -j DNAT --to-destination
 10.253.253.10:80
sudo iptables -t nat -A PREROUTING -i ens33 -p tcp --dport 10443 -j DNAT --to-destinatio
n 10.253.253.10:443

echo "Booting VM, eta 10 seconds"
sleep 10
sudo ifconfig tap0 10.253.253.254 netmask 255.255.255.0
```

script to boot mips

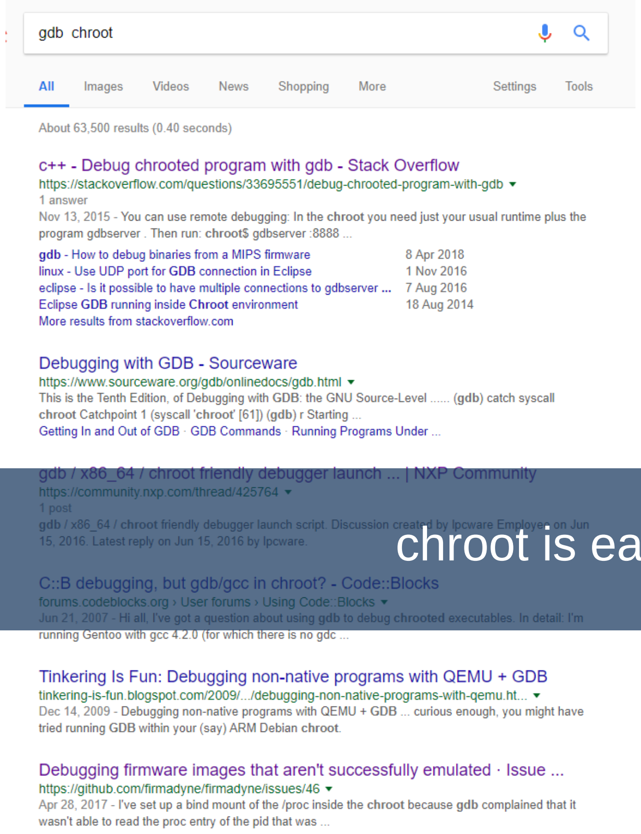


```
#!/bin/bash

sudo screen -dm /opt/qemu/bin/qemu-system-mipsel -m 512 -M malta -kernel boot.stretch.mi
psel/vmlinux-4.9.0-4-4kc-malta -initrd boot.stretch.mipsel/initrd.img-4.9.0-4-4kc-malta
-append "root=/dev/sda1  net.ifnames=0 biosdevname=0 nokaslr" -hda debian-stretch.mipsel
.qcow2 -net nic -net tap,ifname=tap0,script=no,downscript=no -net nic -net tap,ifname=ta
p1,script=no,downscript=no -nographic

sudo tunctl -t tap0 -u xwings
sudo ifconfig tap0 10.253.253.254 netmask 255.255.255.0

sudo sysctl -w net.ipv4.ip_forward=1

echo "Stopping firewall and allowing everyone..."
sudo iptables -F
sudo iptables -X
sudo iptables -t nat -F
sudo iptables -t nat -X
sudo iptables -t mangle -F
sudo iptables -t mangle -X
sudo iptables -P INPUT ACCEPT
sudo iptables -P FORWARD ACCEPT
sudo iptables -P OUTPUT ACCEPT

sudo iptables -t nat -A POSTROUTING -o ens33 -j MASQUERADE
sudo iptables -I FORWARD 1 -i tap0 -j ACCEPT
sudo iptables -I FORWARD 1 -o tap0 -m state --state RELATED,ESTABLISHED -j ACCEPT


sudo iptables -t nat -A PREROUTING -i ens33 -p tcp --dport 1122 -j DNAT --to-destination
 10.253.253.11:22
sudo iptables -t nat -A PREROUTING -i ens33 -p tcp --dport 1180 -j DNAT --to-destination
 10.253.253.11:80
sudo iptables -t nat -A PREROUTING -i ens33 -p tcp --dport 11443 -j DNAT --to-destinatio
n 10.253.253.11:443
```

argument: running new or old distro + kernel

chroot

Easy Way Out, chroot



About 63,500 results (0.40 seconds)

c++ - Debug chrooted program with gdb - Stack Overflow
https://stackoverflow.com/questions/33695551/debug-chrooted-program-with-gdb
1 answer
Nov 13, 2015 - You can use remote debugging: In the **chroot** you need just your usual runtime plus the program gdbserver . Then run: **chroot$** gdbserver :8888 ...

gdb - How to debug binaries from a MIPS firmware	8 Apr 2018
linux - Use UDP port for **GDB** connection in Eclipse	1 Nov 2016
eclipse - Is it possible to have multiple connections to gdbserver ...	7 Aug 2016
Eclipse **GDB** running inside **Chroot** environment	18 Aug 2014

More results from stackoverflow.com

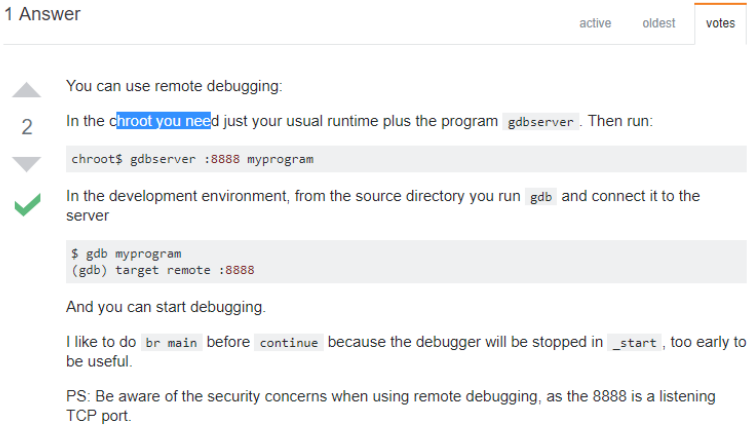
Debugging with GDB - Sourceware
https://www.sourceware.org/gdb/onlinedocs/gdb.html
This is the Tenth Edition, of Debugging with GDB: the GNU Source-Level ...... (**gdb**) catch syscall **chroot** Catchpoint 1 (syscall '**chroot**' [61]) (**gdb**) r Starting ...
Getting In and Out of GDB · GDB Commands · Running Programs Under ...

gdb / x86_64 / chroot friendly debugger launch ... | NXP Community
https://community.nxp.com/thread/425764
1 post
gdb / x86_64 / **chroot** friendly debugger launch script. Discussion created by lpcware Employee on Jun 15, 2016. Latest reply on Jun 15, 2016 by lpcware.

C::B debugging, but gdb/gcc in chroot? - Code::Blocks
forums.codeblocks.org › User forums › Using Code::Blocks
Jun 21, 2007 - Hi all, I've got a question about using **gdb** to debug **chrooted** executables. In detail: I'm running Gentoo with gcc 4.2.0 (for which there is no gdc ...

Tinkering Is Fun: Debugging non-native programs with QEMU + GDB
tinkering-is-fun.blogspot.com/2009/.../debugging-non-native-programs-with-qemu.ht...
Dec 14, 2009 - Debugging non-native programs with QEMU + **GDB** ... curious enough, you might have tried running **GDB** within your (say) ARM Debian **chroot**.

Debugging firmware images that aren't successfully emulated · Issue ...
https://github.com/firmadyne/firmadyne/issues/46
Apr 28, 2017 - I've set up a bind mount of the /proc inside the **chroot** because **gdb** complained that it wasn't able to read the proc entry of the pid that was ...



You can use remote debugging:

In the chroot you need just your usual runtime plus the program `gdbserver`. Then run:

```
chroot$ gdbserver :8888 myprogram
```

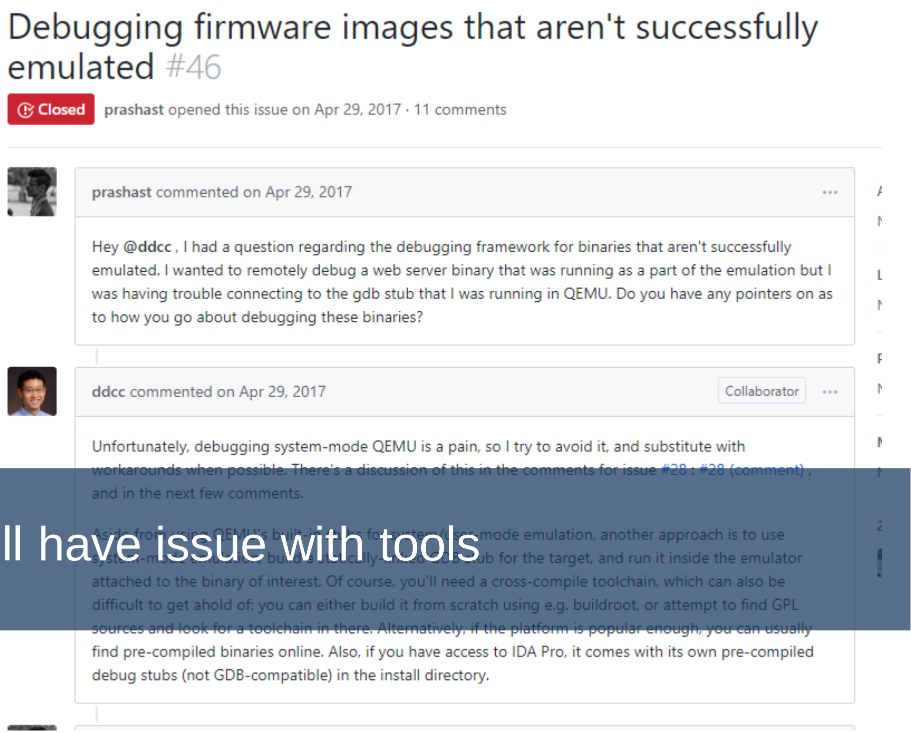
In the development environment, from the source directory you run `gdb` and connect it to the server

```
$ gdb myprogram
(gdb) target remote :8888
```

And you can start debugging.

I like to do `br main` before `continue` because the debugger will be stopped in `_start`, too early to be useful.

PS: Be aware of the security concerns when using remote debugging, as the 8888 is a listening TCP port.



Debugging firmware images that aren't successfully emulated #46

prashast opened this issue on Apr 29, 2017 · 11 comments

prashast commented on Apr 29, 2017

Hey **@ddcc** , I had a question regarding the debugging framework for binaries that aren't successfully emulated. I wanted to remotely debug a web server binary that was running as a part of the emulation but I was having trouble connecting to the gdb stub that I was running in QEMU. Do you have any pointers on as to how you go about debugging these binaries?

ddcc commented on Apr 29, 2017

Unfortunately, debugging system-mode QEMU is a pain, so I try to avoid it, and substitute with workarounds when possible. There's a discussion of this in the comments for issue #28 : #28 (comment) , and in the next few comments.

Aside from using QEMU's built-in stubs for system/user-mode emulation, another approach is to use system-mode emulation, build a statically-linked GDB stub for the target, and run it inside the emulator attached to the binary of interest. Of course, you'll need a cross-compile toolchain, which can also be difficult to get ahold of; you can either build it from scratch using e.g. buildroot, or attempt to find GPL sources and look for a toolchain in there. Alternatively, if the platform is popular enough, you can usually find pre-compiled binaries online. Also, if you have access to IDA Pro, it comes with its own pre-compiled debug stubs (not GDB-compatible) in the install directory.

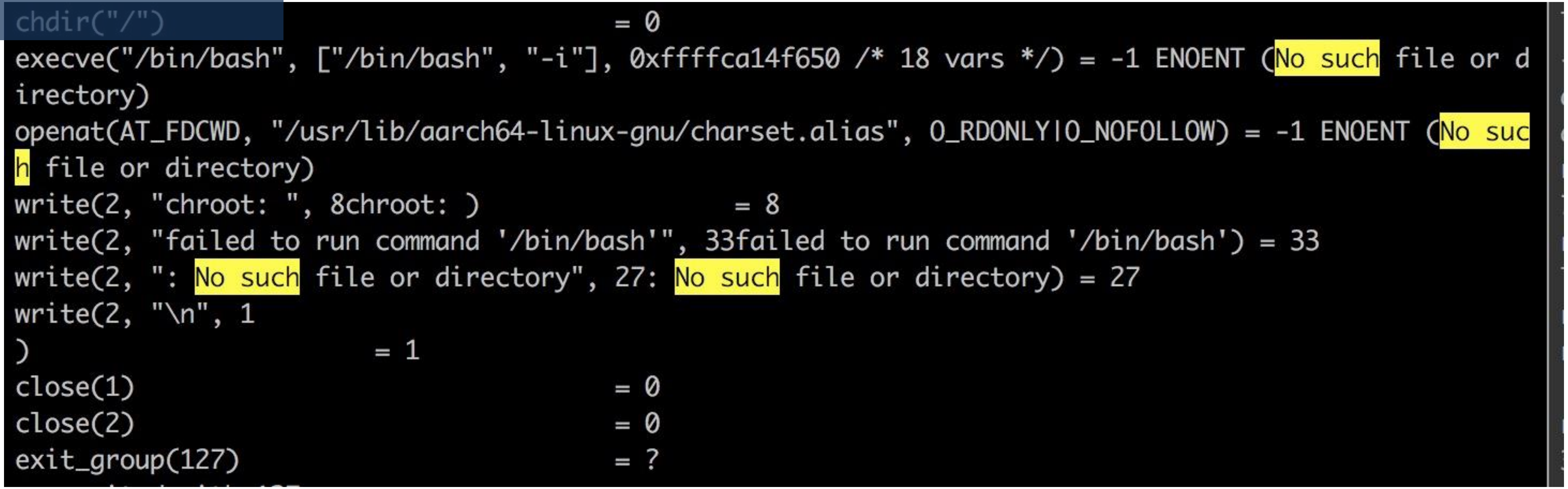
chroot is easy (still hardware dependent), but we will have issue with tools

Running without chroot

Stage 0 Issue: File Not Found

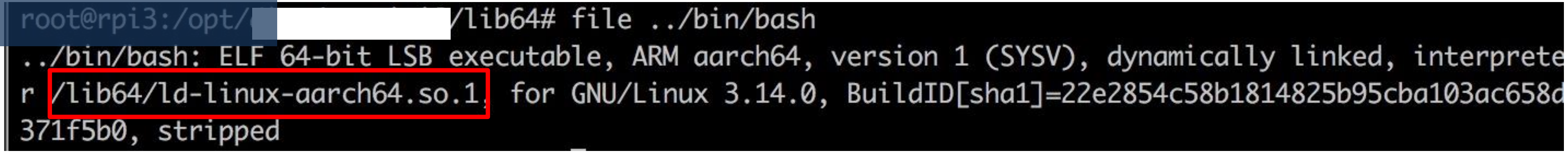
The File Missing Trick

We Missed You



```
chdir("/")                                      = 0
execve("/bin/bash", ["/bin/bash", "-i"], 0xffffca14f650 /* 18 vars */) = -1 ENOENT (No such file or d
irectory)
openat(AT_FDCWD, "/usr/lib/aarch64-linux-gnu/charset.alias", O_RDONLY|O_NOFOLLOW) = -1 ENOENT (No suc
h file or directory)
write(2, "chroot: ", 8chroot: )                         = 8
write(2, "failed to run command '/bin/bash'", 33failed to run command '/bin/bash') = 33
write(2, ": No such file or directory", 27: No such file or directory) = 27
write(2, "\n", 1
)                              = 1
close(1)                                        = 0
close(2)                                        = 0
exit_group(127)                                 = ?
```

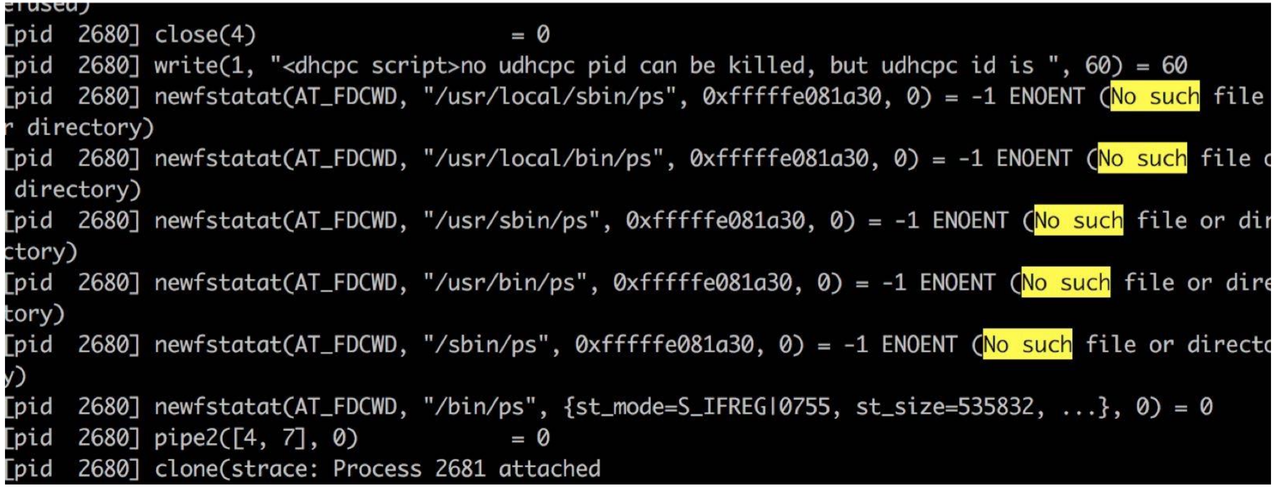
We found you



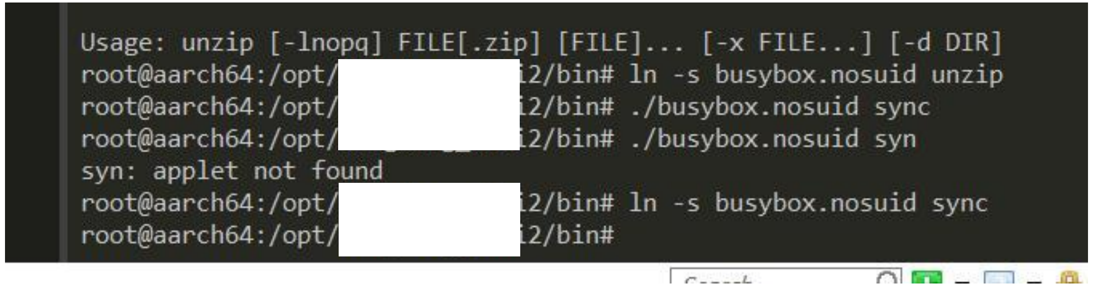
```
root@rpi3:/opt/[redacted]/lib64# file ../bin/bash
../bin/bash: ELF 64-bit LSB executable, ARM aarch64, version 1 (SYSV), dynamically linked, interprete
r /lib64/ld-linux-aarch64.so.1, for GNU/Linux 3.14.0, BuildID[sha1]=22e2854c58b1814825b95cba103ac658d
371f5b0, stripped
```

Stage 1 Issue: .SO Not Found

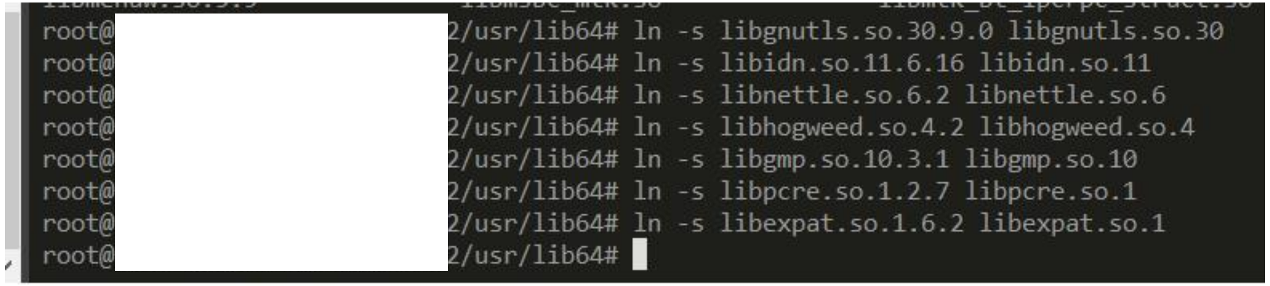
Out from chroot, we need feeeding



```
erused)
[pid  2680] close(4)                          = 0
[pid  2680] write(1, "<dhcpc script>no udhcpc pid can be killed, but udhcpc id is ", 60) = 60
[pid  2680] newfstatat(AT_FDCWD, "/usr/local/sbin/ps", 0xfffffe081a30, 0) = -1 ENOENT (No such file
r directory)
[pid  2680] newfstatat(AT_FDCWD, "/usr/local/bin/ps", 0xfffffe081a30, 0) = -1 ENOENT (No such file o
 directory)
[pid  2680] newfstatat(AT_FDCWD, "/usr/sbin/ps", 0xfffffe081a30, 0) = -1 ENOENT (No such file or dir
ctory)
[pid  2680] newfstatat(AT_FDCWD, "/usr/bin/ps", 0xfffffe081a30, 0) = -1 ENOENT (No such file or dire
tory)
[pid  2680] newfstatat(AT_FDCWD, "/sbin/ps", 0xfffffe081a30, 0) = -1 ENOENT (No such file or directo
y)
[pid  2680] newfstatat(AT_FDCWD, "/bin/ps", {st_mode=S_IFREG|0755, st_size=535832, ...}, 0) = 0
[pid  2680] pipe2([4, 7], 0)                  = 0
[pid  2680] clone(strace: Process 2681 attached
```



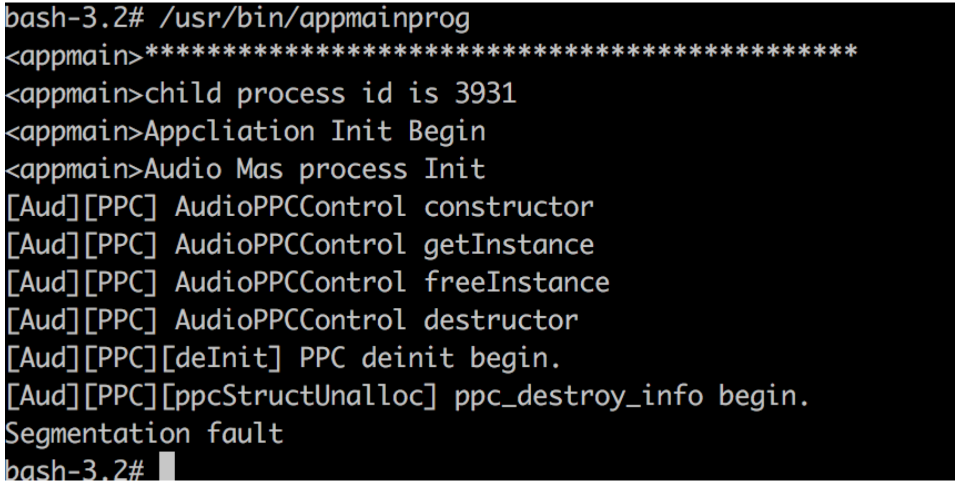
```
Usage: unzip [-lnopq] FILE[.zip] [FILE]... [-x FILE...] [-d DIR]
root@aarch64:/opt/[redacted]i2/bin# ln -s busybox.nosuid unzip
root@aarch64:/opt/[redacted]i2/bin# ./busybox.nosuid sync
root@aarch64:/opt/[redacted]i2/bin# ./busybox.nosuid syn
syn: applet not found
root@aarch64:/opt/[redacted]i2/bin# ln -s busybox.nosuid sync
root@aarch64:/opt/[redacted]i2/bin#
```



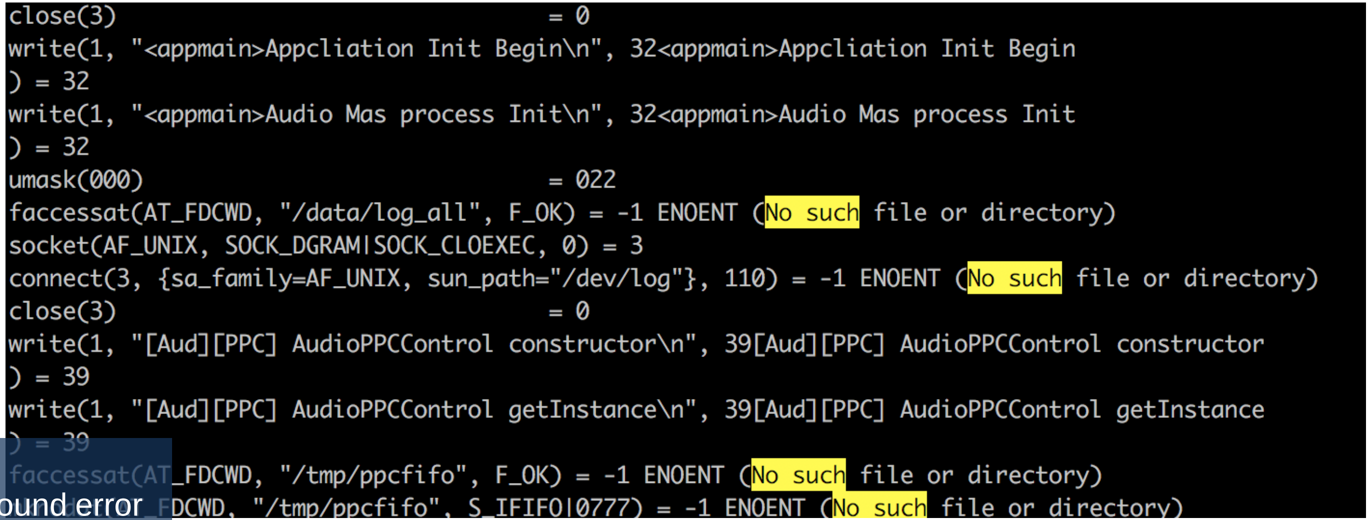
```
root@[redacted]2/usr/lib64# ln -s libgnutls.so.30.9.0 libgnutls.so.30
root@[redacted]2/usr/lib64# ln -s libidn.so.11.6.16 libidn.so.11
root@[redacted]2/usr/lib64# ln -s libnettle.so.6.2 libnettle.so.6
root@[redacted]2/usr/lib64# ln -s libhogweed.so.4.2 libhogweed.so.4
root@[redacted]2/usr/lib64# ln -s libgmp.so.10.3.1 libgmp.so.10
root@[redacted]2/usr/lib64# ln -s libpcre.so.1.2.7 libpcre.so.1
root@[redacted]2/usr/lib64# ln -s libexpat.so.1.6.2 libexpat.so.1
root@[redacted]2/usr/lib64# 
```

Feeding all the required so and binary with “ln –s”

Out from chroot, we need feeding



```
bash-3.2# /usr/bin/appmainprog
<appmain>**********************************************
<appmain>child process id is 3931
<appmain>Appcliation Init Begin
<appmain>Audio Mas process Init
[Aud][PPC] AudioPPCControl constructor
[Aud][PPC] AudioPPCControl getInstance
[Aud][PPC] AudioPPCControl freeInstance
[Aud][PPC] AudioPPCControl destructor
[Aud][PPC][deInit] PPC deinit begin.
[Aud][PPC][ppcStructUnalloc] ppc_destroy_info begin.
Segmentation fault
bash-3.2#
```



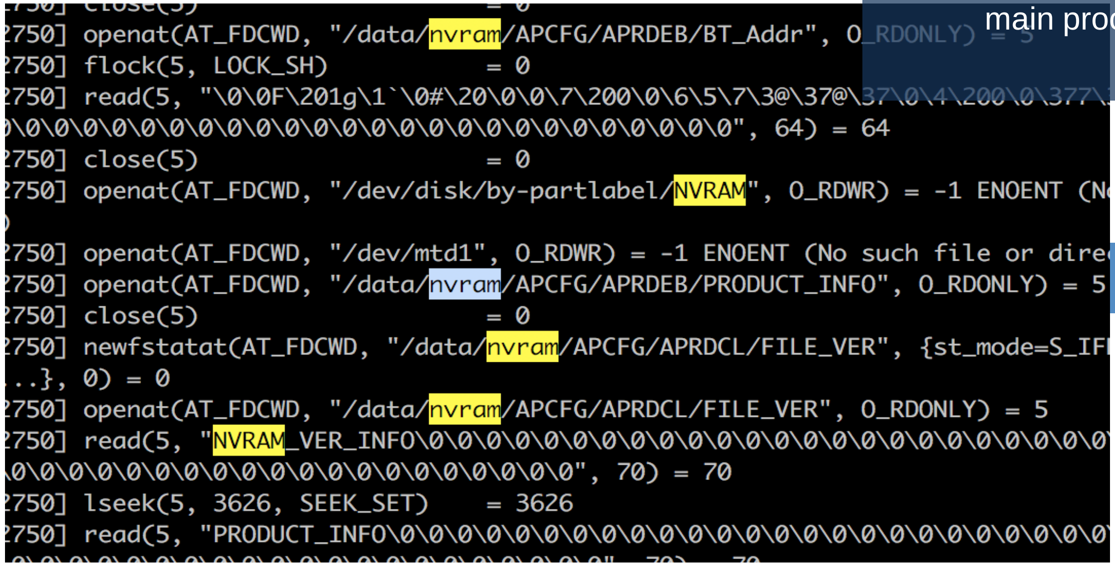
```
close(3)                                = 0
write(1, "<appmain>Appcliation Init Begin\n", 32<appmain>Appcliation Init Begin
) = 32
write(1, "<appmain>Audio Mas process Init\n", 32<appmain>Audio Mas process Init
) = 32
umask(000)                              = 022
faccessat(AT_FDCWD, "/data/log_all", F_OK) = -1 ENOENT (No such file or directory)
socket(AF_UNIX, SOCK_DGRAM|SOCK_CLOEXEC, 0) = 3
connect(3, {sa_family=AF_UNIX, sun_path="/dev/log"}, 110) = -1 ENOENT (No such file or directory)
close(3)                                = 0
write(1, "[Aud][PPC] AudioPPCControl constructor\n", 39[Aud][PPC] AudioPPCControl constructor
) = 39
write(1, "[Aud][PPC] AudioPPCControl getInstance\n", 39[Aud][PPC] AudioPPCControl getInstance
) = 39
faccessat(AT_FDCWD, "/tmp/ppcfifo", F_OK) = -1 ENOENT (No such file or directory)
mknodat(AT_FDCWD, "/tmp/ppcfifo", S_IFIFO|0777) = -1 ENOENT (No such file or directory)
```

Classical file not found error

“segfault” without clear error. strace come to rescue

NVram

Dark side of NVRAM



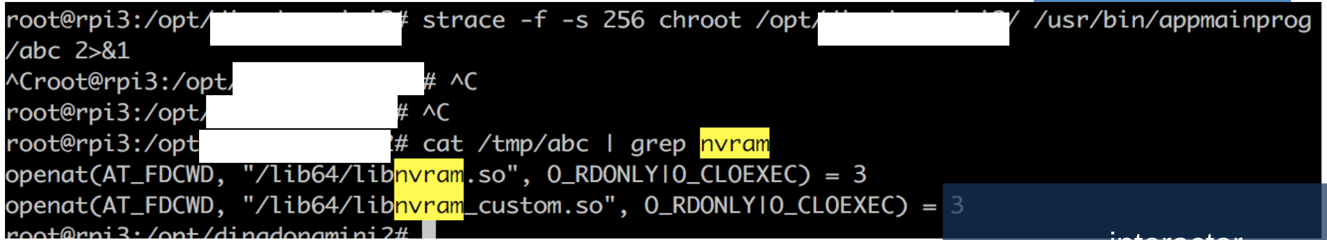
```
2750] close(5)                   = 0
2750] openat(AT_FDCWD, "/data/nvram/APCFG/APRDEB/BT_Addr", O_RDONLY) = 5
2750] flock(5, LOCK_SH)          = 0
2750] read(5, "\0\0F\201g\1`\0#\20\0\0\7\200\0\6\5\7\3@\37@\37\0\4\200\0\377\
)\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0", 64) = 64
2750] close(5)                   = 0
2750] openat(AT_FDCWD, "/dev/disk/by-partlabel/NVRAM", O_RDWR) = -1 ENOENT (N
)

2750] openat(AT_FDCWD, "/dev/mtd1", O_RDWR) = -1 ENOENT (No such file or dire
2750] openat(AT_FDCWD, "/data/nvram/APCFG/APRDEB/PRODUCT_INFO", O_RDONLY) = 5
2750] close(5)                   = 0
2750] newfstatat(AT_FDCWD, "/data/nvram/APCFG/APRDCL/FILE_VER", {st_mode=S_IF
...}, 0) = 0
2750] openat(AT_FDCWD, "/data/nvram/APCFG/APRDCL/FILE_VER", O_RDONLY) = 5
2750] read(5, "NVRAM_VER_INFO\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0", 70) = 70
2750] lseek(5, 3626, SEEK_SET)   = 3626
2750] read(5, "PRODUCT_INFO\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
```

main process

ask for nvram info

Relationship between main binary is so intimate, but in actual fact. Is just a hit and run

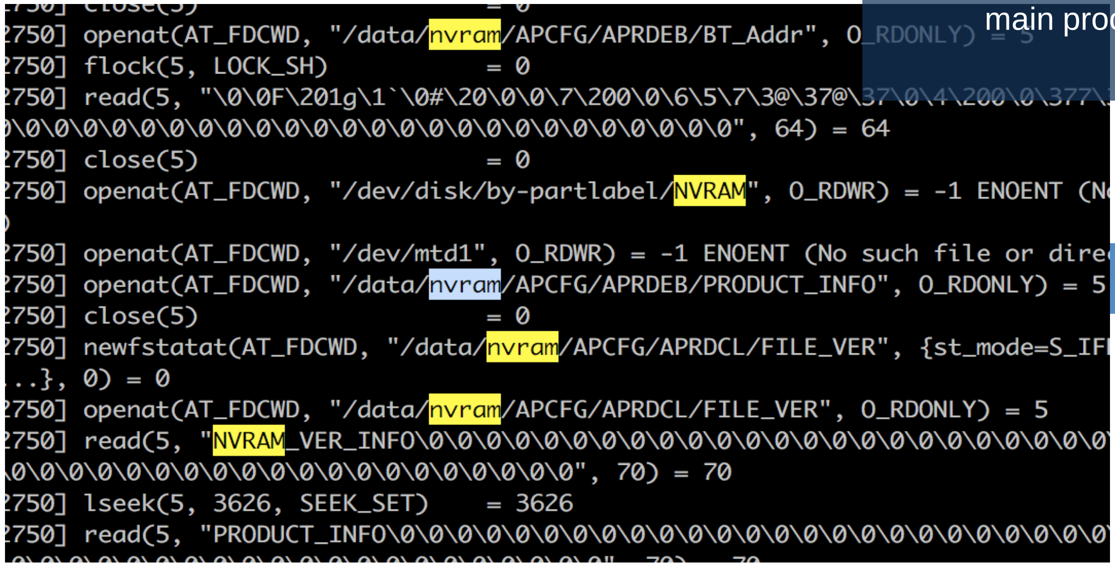
reply with nvram info



```
root@rpi3:/opt/[redacted]# strace -f -s 256 chroot /opt/[redacted]/ /usr/bin/appmainprog
/abc 2>&1
^Croot@rpi3:/opt/[redacted]# ^C
root@rpi3:/opt/[redacted]# ^C
root@rpi3:/opt[redacted]# cat /tmp/abc | grep nvram
openat(AT_FDCWD, "/lib64/libnvram.so", O_RDONLY|O_CLOEXEC) = 3
openat(AT_FDCWD, "/lib64/libnvram_custom.so", O_RDONLY|O_CLOEXEC) = 3
root@rpi3:/opt/dinadonamini2#
```

interactor

Dark Side of NVRAM



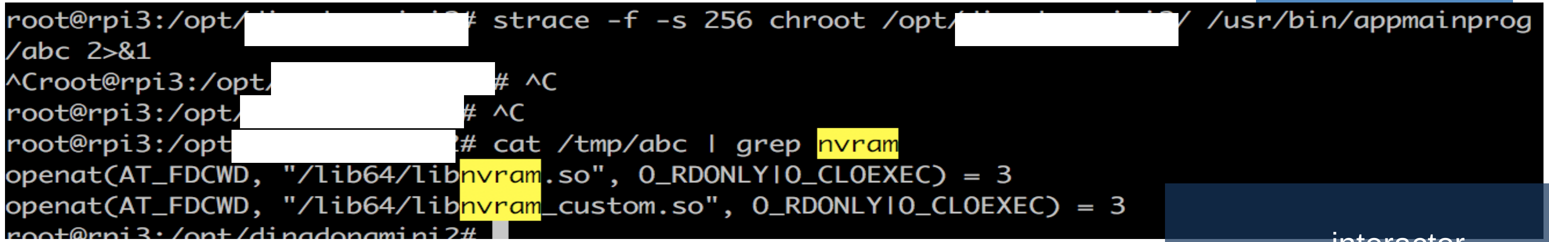
```
2750] openat(AT_FDCWD, "/data/nvram/APCFG/APRDEB/BT_Addr", O_RDONLY) = 5
2750] flock(5, LOCK_SH)                = 0
2750] read(5, "\0\0F\201g\1`\0#\20\0\0\7\200\0\6\5\7\3@\37@\37\0\4\200\0\377\
)\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0", 64) = 64
2750] close(5)                         = 0
2750] openat(AT_FDCWD, "/dev/disk/by-partlabel/NVRAM", O_RDWR) = -1 ENOENT (No
)
2750] openat(AT_FDCWD, "/dev/mtd1", O_RDWR) = -1 ENOENT (No such file or dire
2750] openat(AT_FDCWD, "/data/nvram/APCFG/APRDEB/PRODUCT_INFO", O_RDONLY) = 5
2750] close(5)                         = 0
2750] newfstatat(AT_FDCWD, "/data/nvram/APCFG/APRDCL/FILE_VER", {st_mode=S_IFI
...}, 0) = 0
2750] openat(AT_FDCWD, "/data/nvram/APCFG/APRDCL/FILE_VER", O_RDONLY) = 5
2750] read(5, "NVRAM_VER_INFO\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0", 70) = 70
2750] lseek(5, 3626, SEEK_SET)     = 3626
2750] read(5, "PRODUCT_INFO\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
```

main process

ask for nvram info

Relationship between main binary is so intimate, but in actual fact. Is just a hit and run

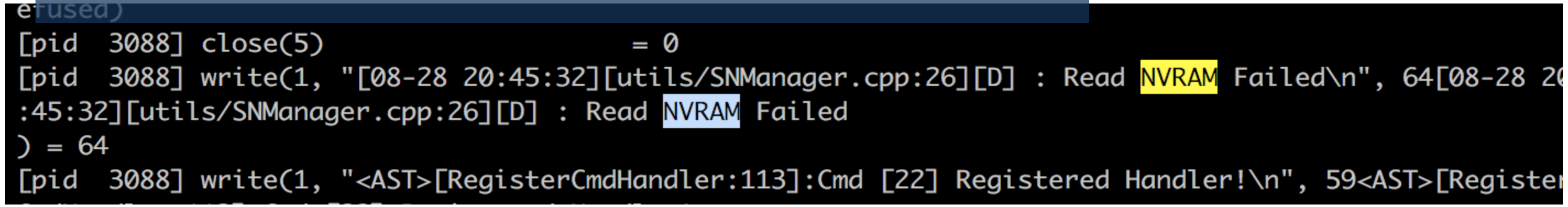
reply with nvram info



```
root@rpi3:/opt/[redacted]# strace -f -s 256 chroot /opt/[redacted]/ /usr/bin/appmainprog
/abc 2>&1
^Croot@rpi3:/opt/[redacted]# ^C
root@rpi3:/opt/[redacted]# ^C
root@rpi3:/opt[redacted]# cat /tmp/abc | grep nvram
openat(AT_FDCWD, "/lib64/libnvram.so", O_RDONLY|O_CLOEXEC) = 3
openat(AT_FDCWD, "/lib64/libnvram_custom.so", O_RDONLY|O_CLOEXEC) = 3
root@rpi3:/opt/dinadonamini2#
```

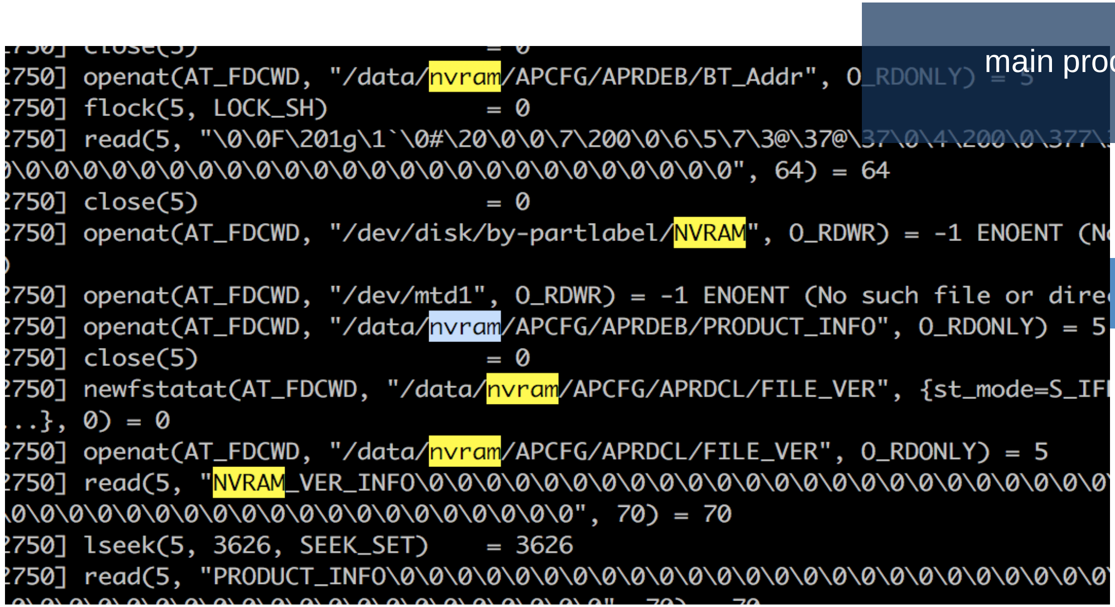
interactor

Dark Side of the main process, we ignore and con’t to next step



```
efused)
[pid  3088] close(5)                        = 0
[pid  3088] write(1, "[08-28 20:45:32][utils/SNManager.cpp:26][D] : Read NVRAM Failed\n", 64[08-28 20
:45:32][utils/SNManager.cpp:26][D] : Read NVRAM Failed
) = 64
[pid  3088] write(1, "<AST>[RegisterCmdHandler:113]:Cmd [22] Registered Handler!\n", 59<AST>[Registe
```

A fake NVRAM



```
2750] openat(AT_FDCWD, "/data/nvram/APCFG/APRDEB/BT_Addr", O_RDONLY) = 5
2750] flock(5, LOCK_SH)            = 0
2750] read(5, "\0\0F\201g\1`\0#\20\0\0\7\200\0\6\5\7\3@\37@\37\0\4\200\0\377\
)\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0", 64) = 64
2750] close(5)                     = 0
2750] openat(AT_FDCWD, "/dev/disk/by-partlabel/NVRAM", O_RDWR) = -1 ENOENT (No
)
2750] openat(AT_FDCWD, "/dev/mtd1", O_RDWR) = -1 ENOENT (No such file or direc
2750] openat(AT_FDCWD, "/data/nvram/APCFG/APRDEB/PRODUCT_INFO", O_RDONLY) = 5
2750] close(5)                     = 0
2750] newfstatat(AT_FDCWD, "/data/nvram/APCFG/APRDCL/FILE_VER", {st_mode=S_IFR
..}, 0) = 0
2750] openat(AT_FDCWD, "/data/nvram/APCFG/APRDCL/FILE_VER", O_RDONLY) = 5
2750] read(5, "NVRAM_VER_INFO\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\
\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0", 70) = 70
2750] lseek(5, 3626, SEEK_SET)     = 3626
2750] read(5, "PRODUCT_INFO\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
```

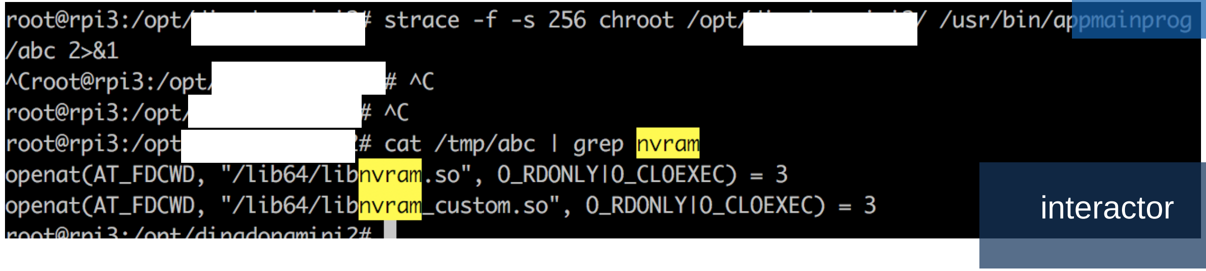
main process

ask for nvram info

IF interactor is the medium,

can we fake it ?

reply with
nvram info

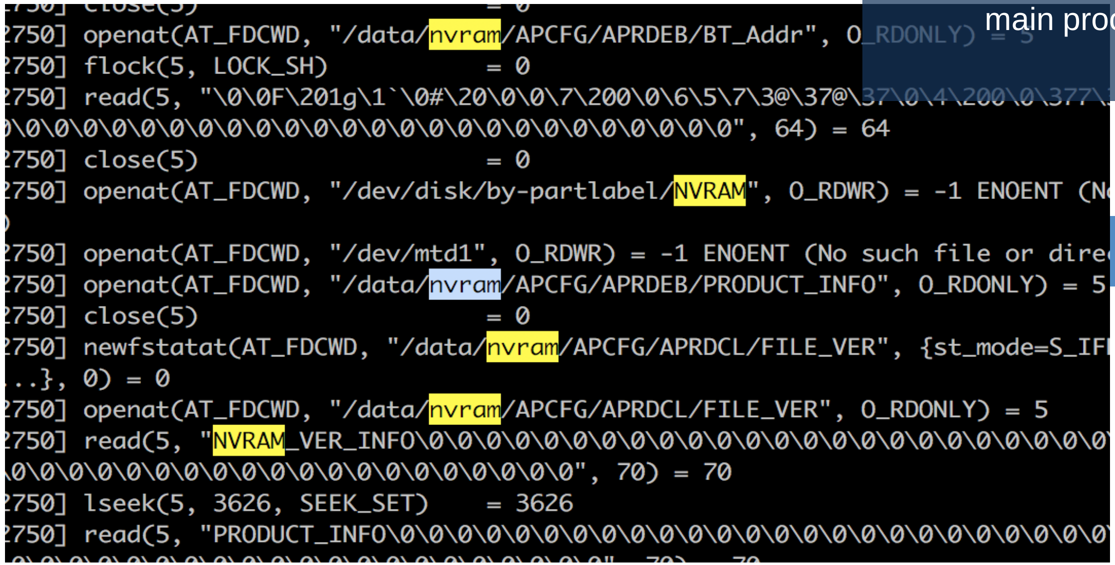


```
root@rpi3:/opt/[redacted]# strace -f -s 256 chroot /opt/[redacted]/ /usr/bin/appmainprog
/abc 2>&1
^Croot@rpi3:/opt/[redacted]# ^C
root@rpi3:/opt/[redacted]# ^C
root@rpi3:/opt[redacted]# cat /tmp/abc | grep nvram
openat(AT_FDCWD, "/lib64/libnvram.so", O_RDONLY|O_CLOEXEC) = 3
openat(AT_FDCWD, "/lib64/libnvram_custom.so", O_RDONLY|O_CLOEXEC) = 3
root@rpi3:/opt/dingdongmini2#
```

interactor

A fake NVRAM

main process



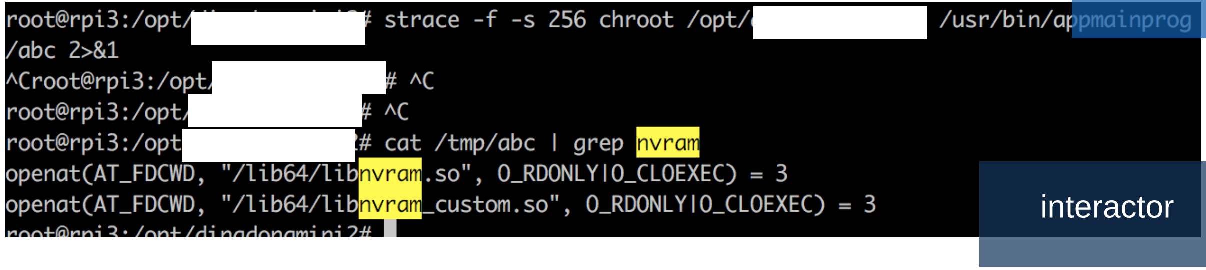
```
2750] close(5)                  = 0
2750] openat(AT_FDCWD, "/data/nvram/APCFG/APRDEB/BT_Addr", O_RDONLY) = 5
2750] flock(5, LOCK_SH)         = 0
2750] read(5, "\0\0F\201g\1`\0#\20\0\0\7\200\0\6\5\7\3@\37@\37\0\4\200\0\377\
)\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0", 64) = 64
2750] close(5)                  = 0
2750] openat(AT_FDCWD, "/dev/disk/by-partlabel/NVRAM", O_RDWR) = -1 ENOENT (N
)
2750] openat(AT_FDCWD, "/dev/mtd1", O_RDWR) = -1 ENOENT (No such file or dire
2750] openat(AT_FDCWD, "/data/nvram/APCFG/APRDEB/PRODUCT_INFO", O_RDONLY) = 5
2750] close(5)                  = 0
2750] newfstatat(AT_FDCWD, "/data/nvram/APCFG/APRDCL/FILE_VER", {st_mode=S_IF
...}, 0) = 0
2750] openat(AT_FDCWD, "/data/nvram/APCFG/APRDCL/FILE_VER", O_RDONLY) = 5
2750] read(5, "NVRAM_VER_INFO\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0", 70) = 70
2750] lseek(5, 3626, SEEK_SET)    = 3626
2750] read(5, "PRODUCT_INFO\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
```

ask for nvram info

IF interactor is the medium,

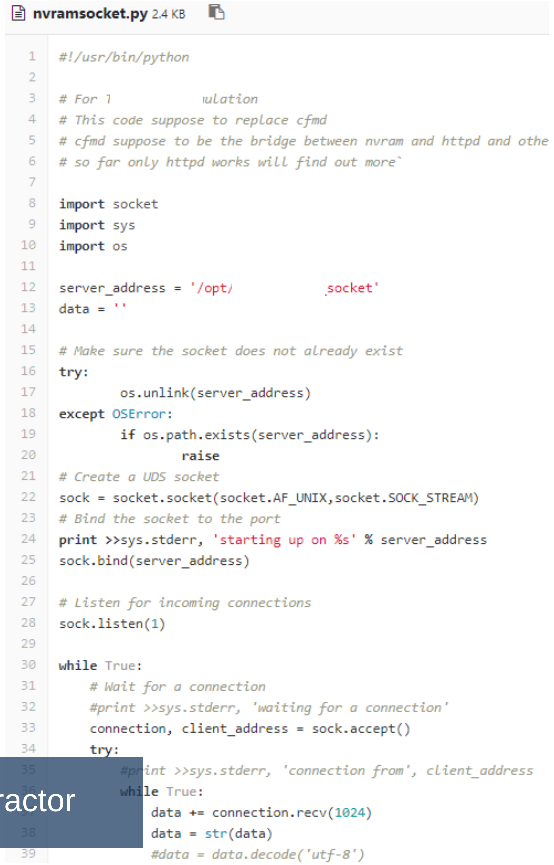
can we fake it ?

reply with nvram info



```
root@rpi3:/opt/[redacted]# strace -f -s 256 chroot /opt/[redacted] /usr/bin/appmainprog
/abc 2>&1
^Croot@rpi3:/opt/[redacted]# ^C
root@rpi3:/opt/[redacted]# ^C
root@rpi3:/opt[redacted]# cat /tmp/abc | grep nvram
openat(AT_FDCWD, "/lib64/libnvram.so", O_RDONLY|O_CLOEXEC) = 3
openat(AT_FDCWD, "/lib64/libnvram_custom.so", O_RDONLY|O_CLOEXEC) = 3
root@rpi3:/opt/[illegible]#
```

interactor



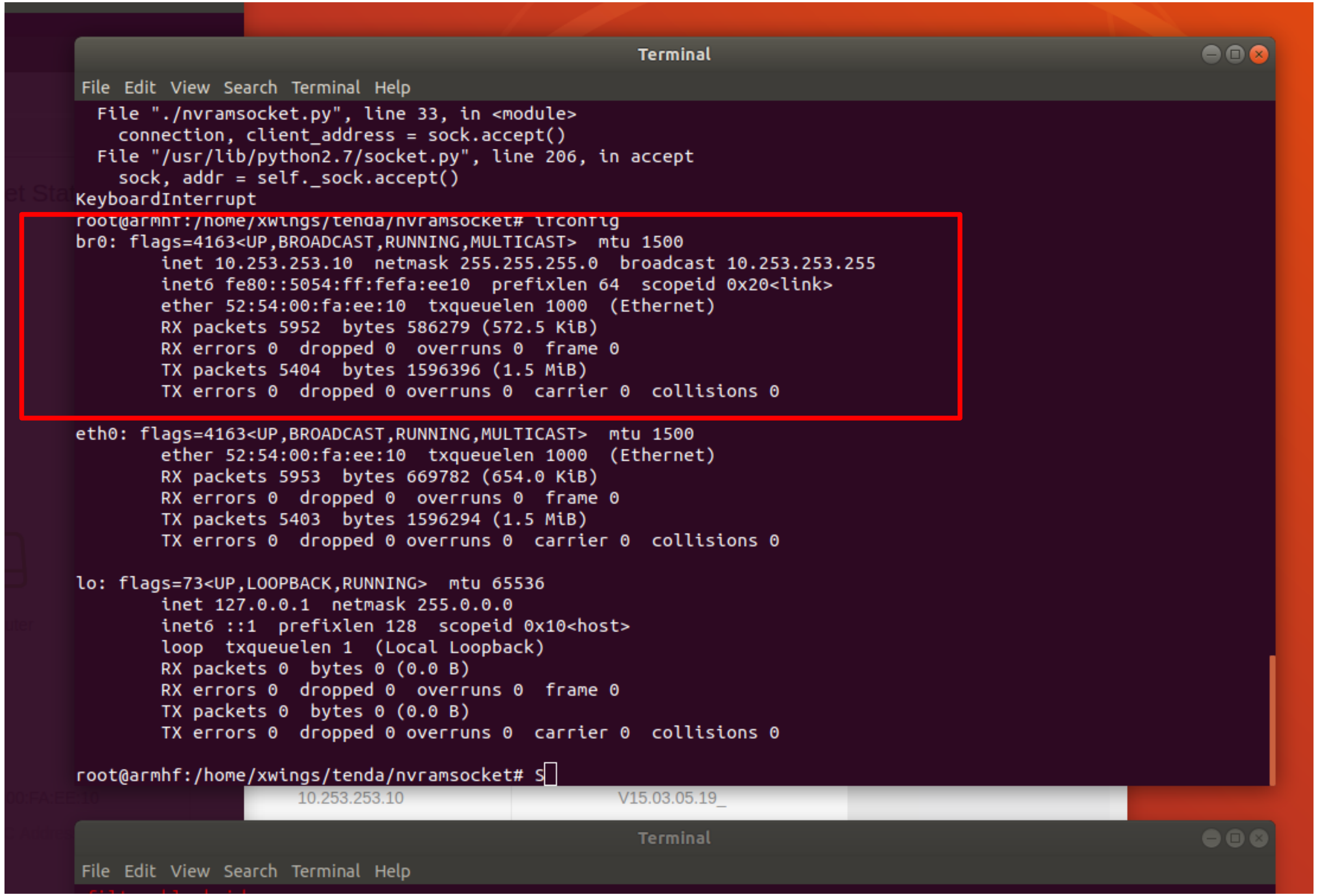
nvramsocket.py 2.4 KB

```python
#!/usr/bin/python

# For [redacted]ulation
# This code suppose to replace cfmd
# cfmd suppose to be the bridge between nvram and httpd and othe
# so far only httpd works will find out more`

import socket
import sys
import os

server_address = '/opt/[redacted]socket'
data = ''

# Make sure the socket does not already exist
try:
        os.unlink(server_address)
except OSError:
        if os.path.exists(server_address):
                raise
# Create a UDS socket
sock = socket.socket(socket.AF_UNIX,socket.SOCK_STREAM)
# Bind the socket to the port
print >>sys.stderr, 'starting up on %s' % server_address
sock.bind(server_address)

# Listen for incoming connections
sock.listen(1)

while True:
    # Wait for a connection
    #print >>sys.stderr, 'waiting for a connection'
    connection, client_address = sock.accept()
    try:
        #print >>sys.stderr, 'connection from', client_address
        while True:
            data += connection.recv(1024)
            data = str(data)
            #data = data.decode('utf-8')
```

Custom Interactor

br0

The bridge trick

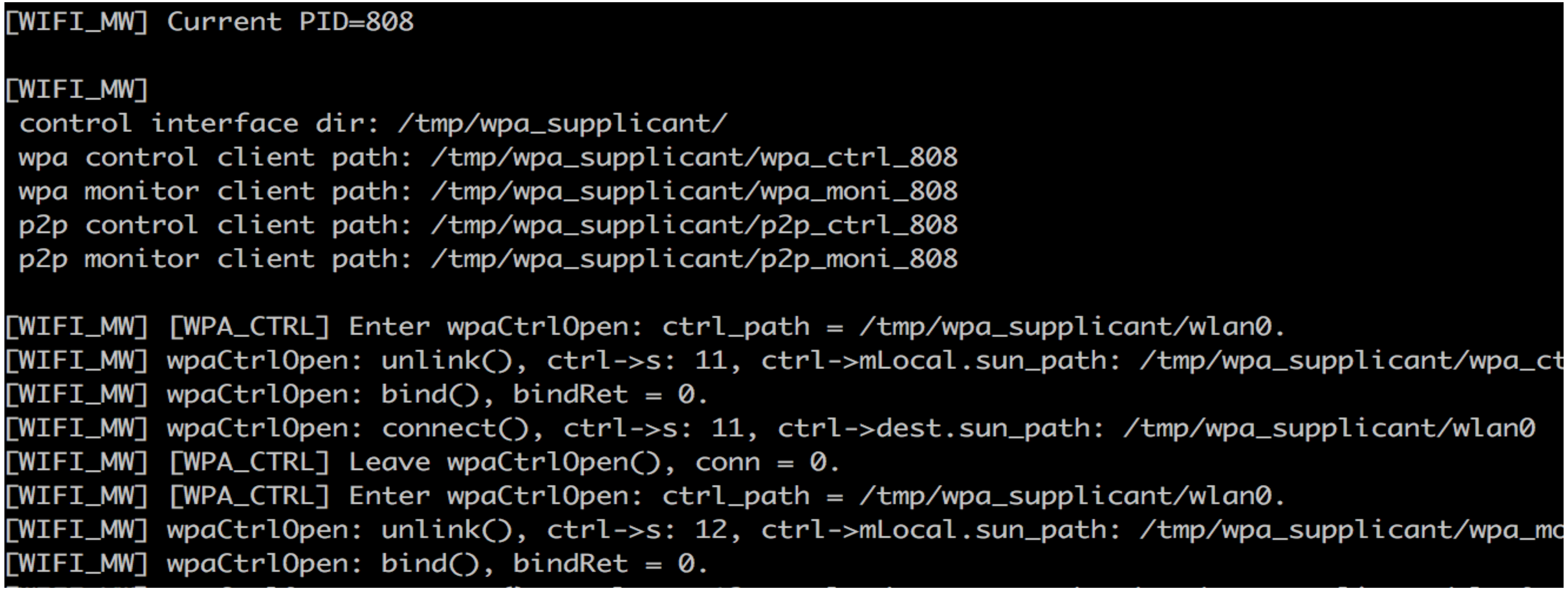

Terminal
File Edit View Search Terminal Help
10.253.253.10
V15.03.05.19_
Terminal
File Edit View Search Terminal Help

```
  File "./nvramsocket.py", line 33, in <module>
    connection, client_address = sock.accept()
  File "/usr/lib/python2.7/socket.py", line 206, in accept
    sock, addr = self._sock.accept()
KeyboardInterrupt
root@armhf:/home/xwings/tenda/nvramsocket# ifconfig
br0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.253.253.10  netmask 255.255.255.0  broadcast 10.253.253.255
        inet6 fe80::5054:ff:fefa:ee10  prefixlen 64  scopeid 0x20<link>
        ether 52:54:00:fa:ee:10  txqueuelen 1000  (Ethernet)
        RX packets 5952  bytes 586279 (572.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5404  bytes 1596396 (1.5 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        ether 52:54:00:fa:ee:10  txqueuelen 1000  (Ethernet)
        RX packets 5953  bytes 669782 (654.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5403  bytes 1596294 (1.5 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@armhf:/home/xwings/tenda/nvramsocket# S
```

The switch looking device

Wireless Devices

Faking wpa_supplicant



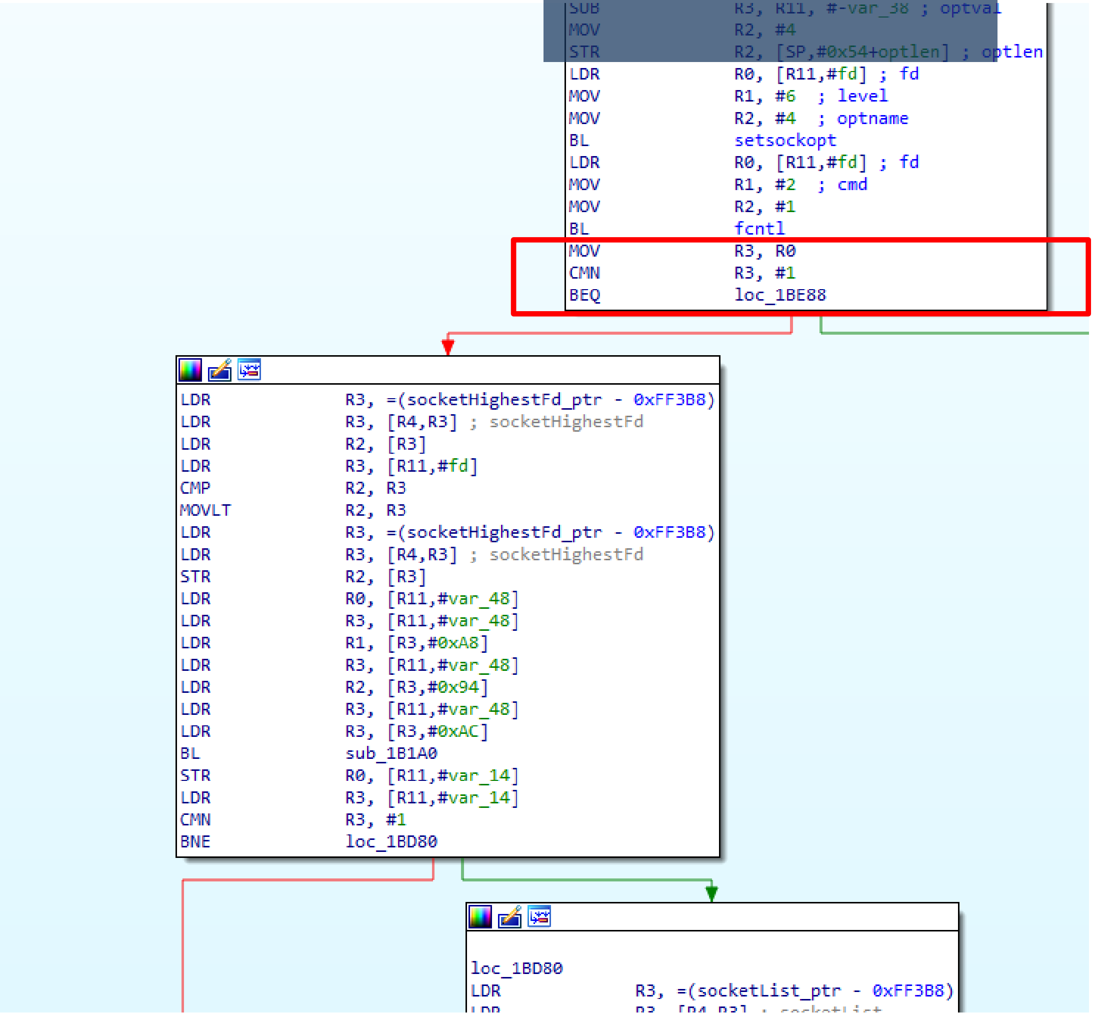
```
[WIFI_MW] Current PID=808

[WIFI_MW]
 control interface dir: /tmp/wpa_supplicant/
 wpa control client path: /tmp/wpa_supplicant/wpa_ctrl_808
 wpa monitor client path: /tmp/wpa_supplicant/wpa_moni_808
 p2p control client path: /tmp/wpa_supplicant/p2p_ctrl_808
 p2p monitor client path: /tmp/wpa_supplicant/p2p_moni_808

[WIFI_MW] [WPA_CTRL] Enter wpaCtrlOpen: ctrl_path = /tmp/wpa_supplicant/wlan0.
[WIFI_MW] wpaCtrlOpen: unlink(), ctrl->s: 11, ctrl->mLocal.sun_path: /tmp/wpa_supplicant/wpa_ct
[WIFI_MW] wpaCtrlOpen: bind(), bindRet = 0.
[WIFI_MW] wpaCtrlOpen: connect(), ctrl->s: 11, ctrl->dest.sun_path: /tmp/wpa_supplicant/wlan0
[WIFI_MW] [WPA_CTRL] Leave wpaCtrlOpen(), conn = 0.
[WIFI_MW] [WPA_CTRL] Enter wpaCtrlOpen: ctrl_path = /tmp/wpa_supplicant/wlan0.
[WIFI_MW] wpaCtrlOpen: unlink(), ctrl->s: 12, ctrl->mLocal.sun_path: /tmp/wpa_supplicant/wpa_mo
[WIFI_MW] wpaCtrlOpen: bind(), bindRet = 0.
```

making eth0 looks like wlan0 works too

Every Thing Else Fail

BL, BNE, BEQ and friends

Original BIN

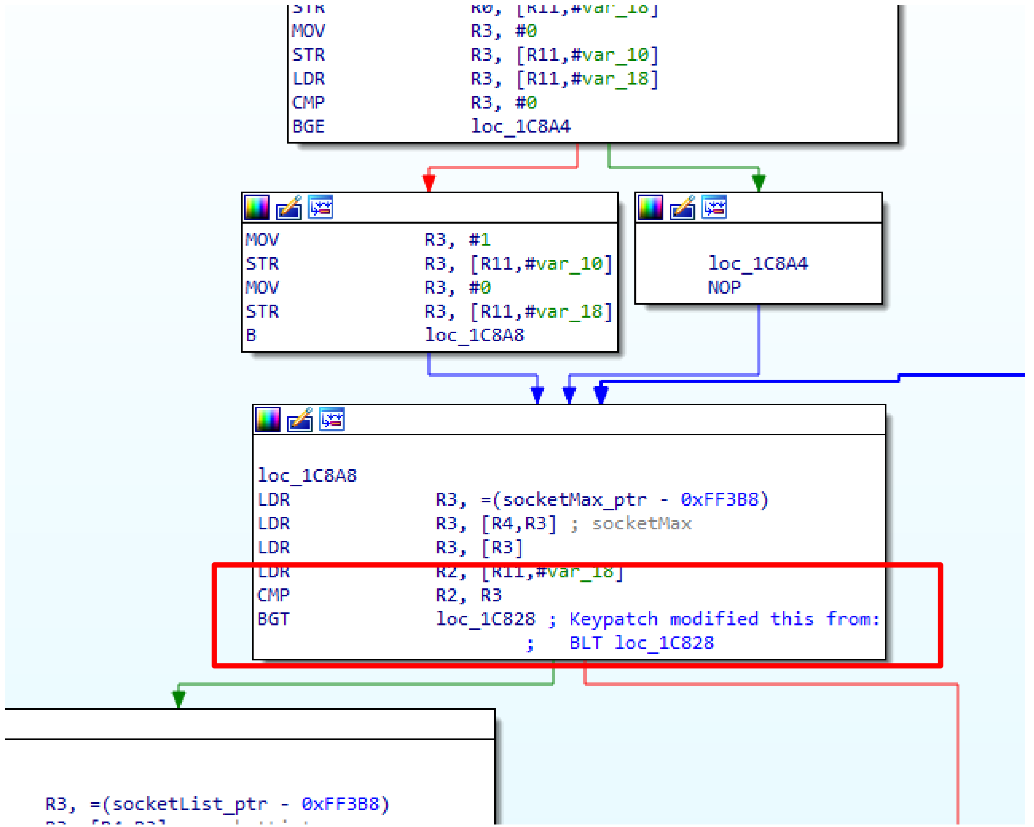


```
SUB             R3, R11, #-var_38 ; optval
MOV             R2, #4
STR             R2, [SP,#0x54+optlen] ; optlen
LDR             R0, [R11,#fd] ; fd
MOV             R1, #6  ; level
MOV             R2, #4  ; optname
BL              setsockopt
LDR             R0, [R11,#fd] ; fd
MOV             R1, #2  ; cmd
MOV             R2, #1
BL              fcntl
MOV             R3, R0
CMN             R3, #1
BEQ             loc_1BE88
```

```
LDR             R3, =(socketHighestFd_ptr - 0xFF3B8)
LDR             R3, [R4,R3] ; socketHighestFd
LDR             R2, [R3]
LDR             R3, [R11,#fd]
CMP             R2, R3
MOVLT           R2, R3
LDR             R3, =(socketHighestFd_ptr - 0xFF3B8)
LDR             R3, [R4,R3] ; socketHighestFd
STR             R2, [R3]
LDR             R0, [R11,#var_48]
LDR             R3, [R11,#var_48]
LDR             R1, [R3,#0xA8]
LDR             R3, [R11,#var_48]
LDR             R2, [R3,#0x94]
LDR             R3, [R11,#var_48]
LDR             R3, [R3,#0xAC]
BL              sub_1B1A0
STR             R0, [R11,#var_14]
LDR             R3, [R11,#var_14]
CMN             R3, #1
BNE             loc_1BD80
```

```
loc_1BD80
LDR             R3, =(socketList_ptr - 0xFF3B8)
LDR             R3, [R4,R3] ; socketList
```

Patched BIN



```
STR             R0, [R11,#var_18]
MOV             R3, #0
STR             R3, [R11,#var_10]
LDR             R3, [R11,#var_18]
CMP             R3, #0
BGE             loc_1C8A4
```

```
MOV             R3, #1
STR             R3, [R11,#var_10]
MOV             R3, #0
STR             R3, [R11,#var_18]
B               loc_1C8A8
```

```
loc_1C8A4
NOP
```

```
loc_1C8A8
LDR             R3, =(socketMax_ptr - 0xFF3B8)
LDR             R3, [R4,R3] ; socketMax
LDR             R3, [R3]
LDR             R2, [R11,#var_18]
CMP             R2, R3
BGT             loc_1C828 ; Keypatch modified this from:
                          ;   BLT loc_1C828
```

```
R3, =(socketList_ptr - 0xFF3B8)
R3, [R4,R3] ; socketList
```

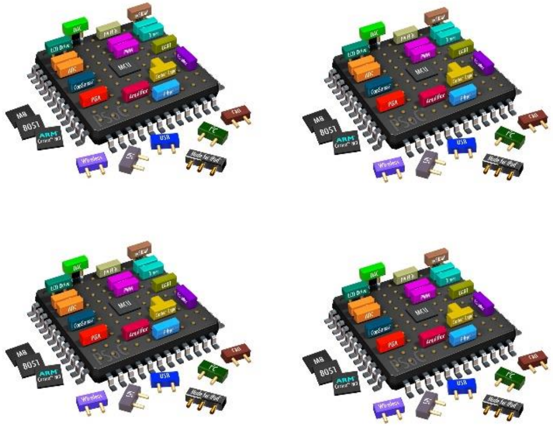
Motivations

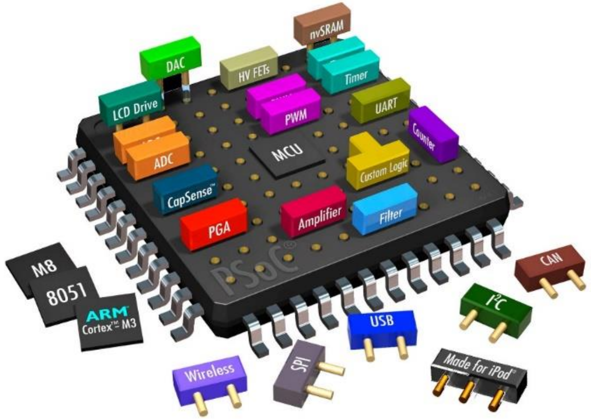
More Resources = More Power

Multicore

MAX RAM

MAX Space





Processor

Normally 1-2 Core

RAM

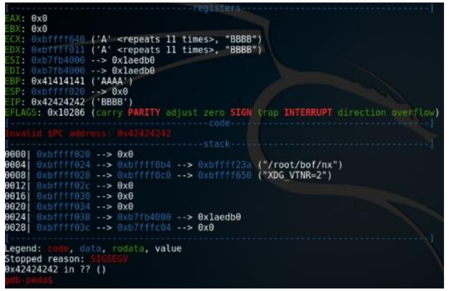
Normally 256MB/512MB

FLASH

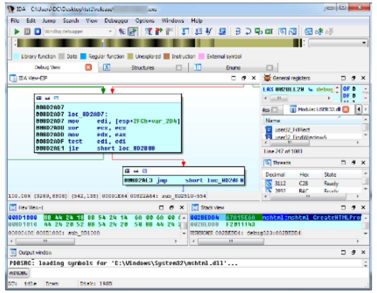
Normally 8MB/16MB/32MB/256MB

Or We Can Just X86 IT

What is Required



```
[----------------------------------registers-----------------------------------]
EAX: 0x0
EBX: 0x0
ECX: 0xbffff640 ('A' <repeats 11 times>, "BBBB")
EDX: 0xbffff011 ('A' <repeats 11 times>, "BBBB")
ESI: 0xb7fb4000 --> 0x1aedb0
EDI: 0xb7fb4000 --> 0x1aedb0
EBP: 0x41414141 ('AAAA')
ESP: 0xbffff020 --> 0x0
EIP: 0x42424242 ('BBBB')
EFLAGS: 0x10286 (carry PARITY adjust zero SIGN trap INTERRUPT direction overflow)
[-------------------------------------code-------------------------------------]
Invalid $PC address: 0x42424242
[------------------------------------stack-------------------------------------]
0000| 0xbffff020 --> 0x0
0004| 0xbffff024 --> 0xbffff0b4 --> 0xbffff23a ("/root/bof/nx")
0008| 0xbffff028 --> 0xbffff0c0 --> 0xbffff650 ("XDG_VTNR=2")
0012| 0xbffff02c --> 0x0
0016| 0xbffff030 --> 0x0
0020| 0xbffff034 --> 0x0
0024| 0xbffff038 --> 0xb7fb4000 --> 0x1aedb0
0028| 0xbffff03c --> 0xb7fffc04 --> 0x0
[------------------------------------------------------------------------------]
Legend: code, data, rodata, value
Stopped reason: SIGSEGV
0x42424242 in ?? ()
gdb-peda$
```

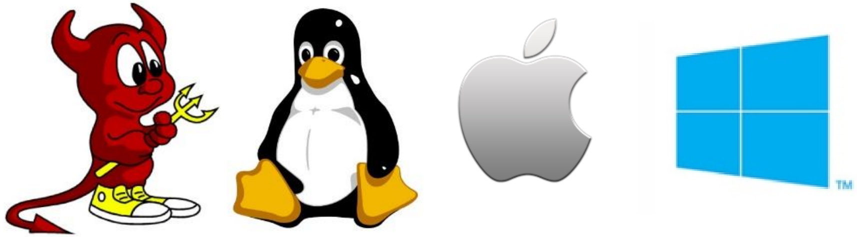


IDA - C:\Users\DC\Desktop\tst1\release\[redacted].exe

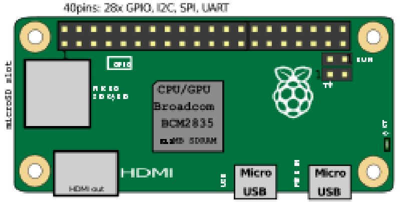
```
008D2AD7
008D2AD7 loc_8D2AD7:
008D2AD7 mov     edi, [esp+2FCh+var_204]
008D2ADB xor     ecx, ecx
008D2ADD mov     edx, eax
008D2ADF test    edi, edi
008D2AE1 jle     short loc_8D2B0B

008D2AE3 jmp     short loc_8D2AE8
```

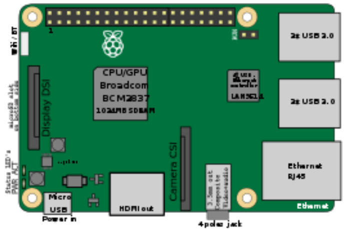
```
PDBSRC: loading symbols for 'C:\Windows\System32\mshtml.dll'...
```

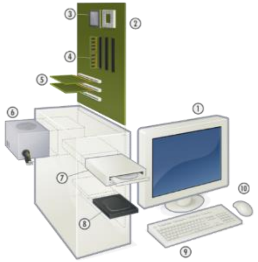
Debugger or Disassembler



*BSD Linux MacOS Windows

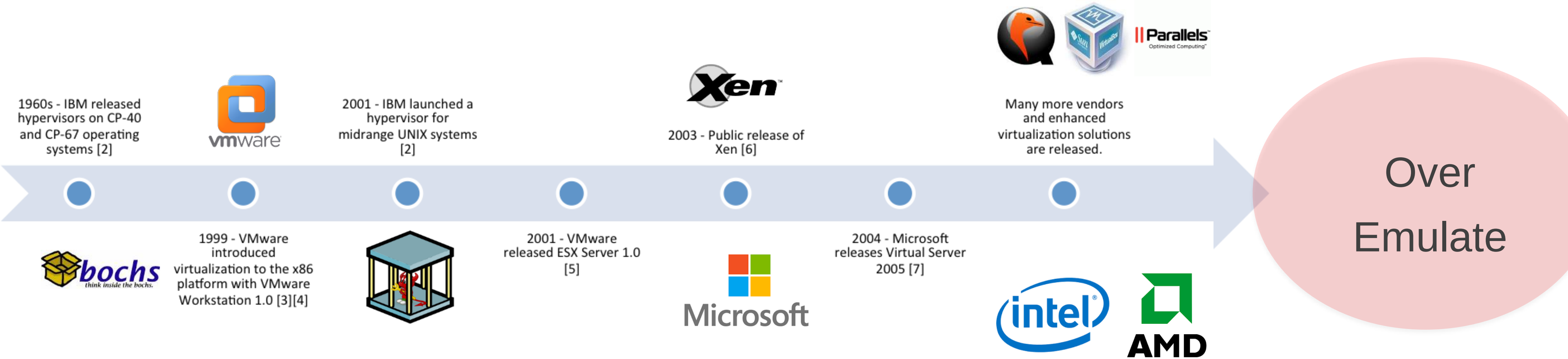






MIPS ARM AARCH64 X86

Why Not Off The Shelf Emulator



1960s - IBM released hypervisors on CP-40 and CP-67 operating systems [2]

1999 - VMware introduced virtualization to the x86 platform with VMware Workstation 1.0 [3][4]

2001 - IBM launched a hypervisor for midrange UNIX systems [2]

2001 - VMware released ESX Server 1.0 [5]

2003 - Public release of Xen [6]

2004 - Microsoft releases Virtual Server 2005 [7]

Many more vendors and enhanced virtualization solutions are released.

Over Emulate

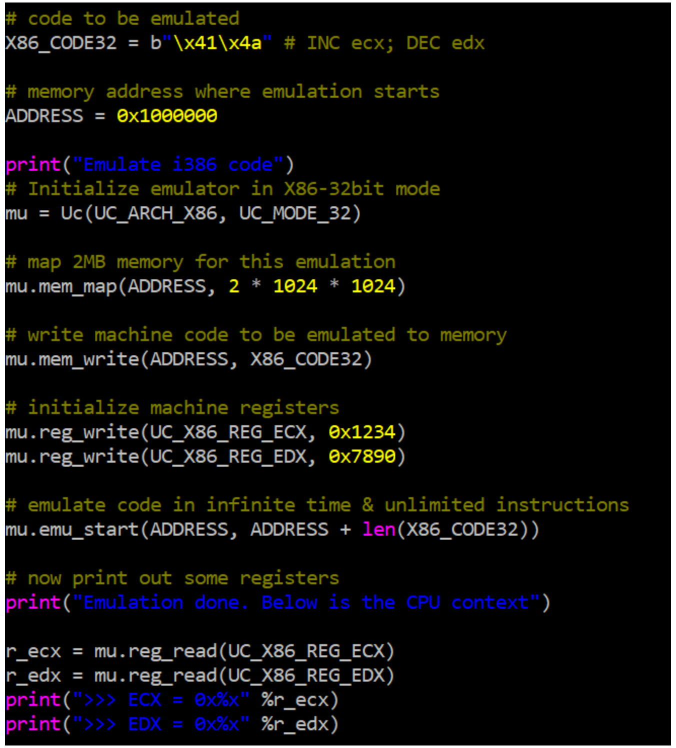
More Emulate = Higher Chances Being Detected

Unicorn Emulator framework

- Multi-architectures: Arm, Arm64, M68K, Mips, Sparc, & X86 (include X86_64)
- Native support for Windows & *nix (with Mac OSX, Linux, *BSD & Solaris confirmed)
- Clean/simple/lightweight/intuitive architecture-neutral API
- Implemented in pure C language, with multiple bindings
- High performance by using Just-In-Time compiler technique
- Support fine-grained instrumentation at various levels

Limitation

- Just emulator for low level instructions + memory access
- No higher level concepts of Operating System
  - File format
  - Library
  - Filesystem
  - Systemcall
  - OS structures



```python
# code to be emulated
X86_CODE32 = b"\x41\x4a" # INC ecx; DEC edx

# memory address where emulation starts
ADDRESS = 0x1000000

print("Emulate i386 code")
# Initialize emulator in X86-32bit mode
mu = Uc(UC_ARCH_X86, UC_MODE_32)

# map 2MB memory for this emulation
mu.mem_map(ADDRESS, 2 * 1024 * 1024)

# write machine code to be emulated to memory
mu.mem_write(ADDRESS, X86_CODE32)

# initialize machine registers
mu.reg_write(UC_X86_REG_ECX, 0x1234)
mu.reg_write(UC_X86_REG_EDX, 0x7890)

# emulate code in infinite time & unlimited instructions
mu.emu_start(ADDRESS, ADDRESS + len(X86_CODE32))

# now print out some registers
print("Emulation done. Below is the CPU context")

r_ecx = mu.reg_read(UC_X86_REG_ECX)
r_edx = mu.reg_read(UC_X86_REG_EDX)
print(">>> ECX = 0x%x" %r_ecx)
print(">>> EDX = 0x%x" %r_edx)
```

Qiling Framework

Features

- Cross platform: Windows, MacOS, Linux, BSD
- Cross architecture: X86, X86_64, Arm, Arm64, Mips
- Multiple file formats: PE, MachO, ELF, UEFI(PE)
- Emulate & sandbox machine code in a isolated environment
- Provide high level API to setup & configure the sandbox
- Fine-grain instrumentation: allow hooks at various levels (instruction/basic-block/memory-access/exception/syscall/IO/etc)
- Allow dynamic hotpatch on-the-fly running code, including the loaded library
- True Python framework, making it easy to build customized analysis tools on top
- Full GDB/IDA/r2 Support
- OS profiling support

User Mode Emulation

qemu-usermode

- The TOOL
- Limited OS Support, Very Limited
- No Multi OS Support
- No Instrumentation
- **Syscall Forwarding**

usercorn

- Very good project !
- It’s a Framework !
- Mostly *nix based only
- Limited OS Support (No Windows)
- Go and Lua is not hacker’s friendly
- **Syscall Forwarding**

Binee

- Very good project too
- Only X86 (32 and 64)
- Limited OS Support (No *NIX)
- Just a tool, we don’t need a tool
- Again, is GO

WINE

- Limited ARCH Support
- Limited OS Support, only Windows
- Not Sandbox Designed
- No Instrumentation

WSL/2

Microsoft

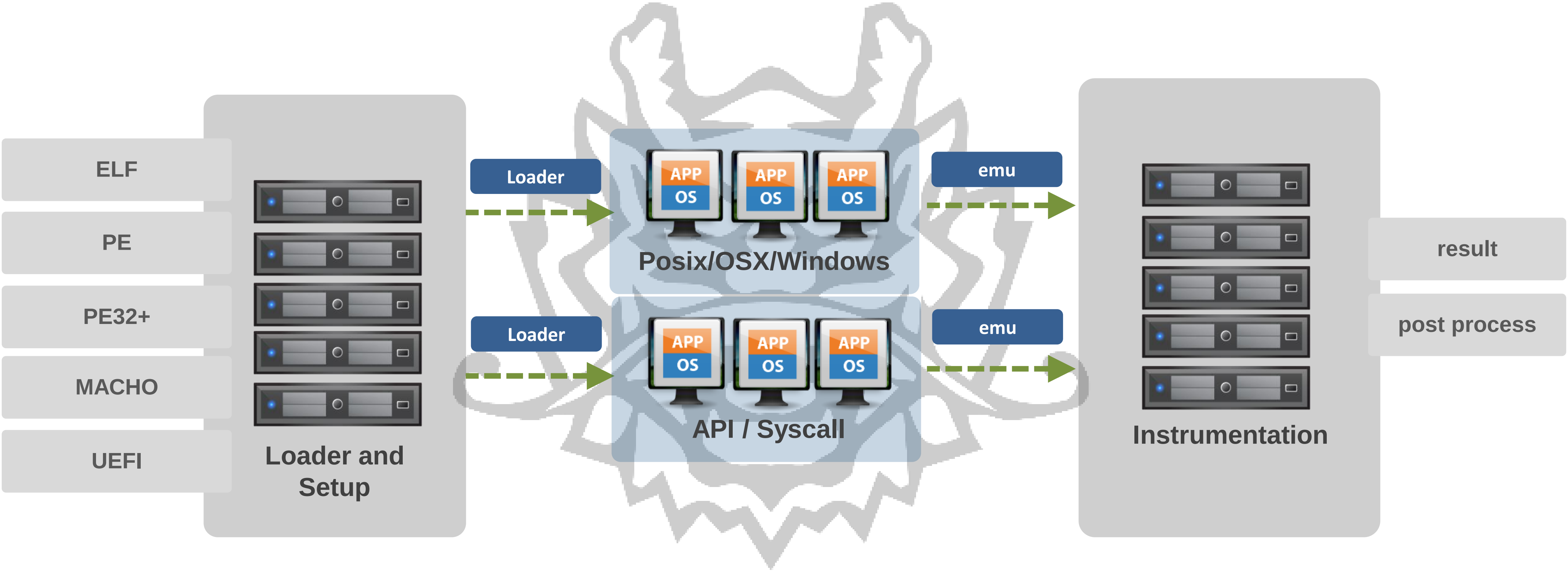
- Limited ARCH Support
- Only Linux and run in Windows
- Not Sandboxed, It linked to /mnt/c
- No Instrumentation (maybe)

Zelos

- Very good project !
- It’s a Framework !
- Linux based only (No Windows)
- Incomplete support for Linux multi arch

How Qiling Works

How Does It Work

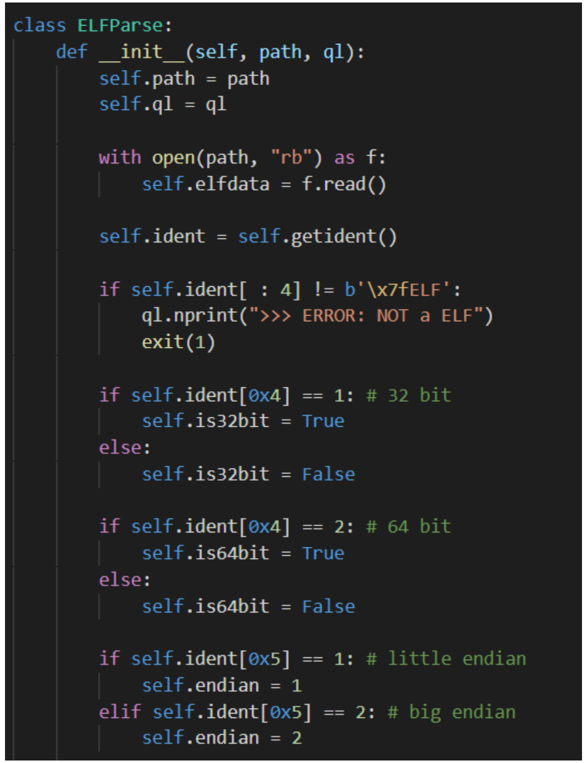


Base OS can be Windows/Linux/BSD or OSX

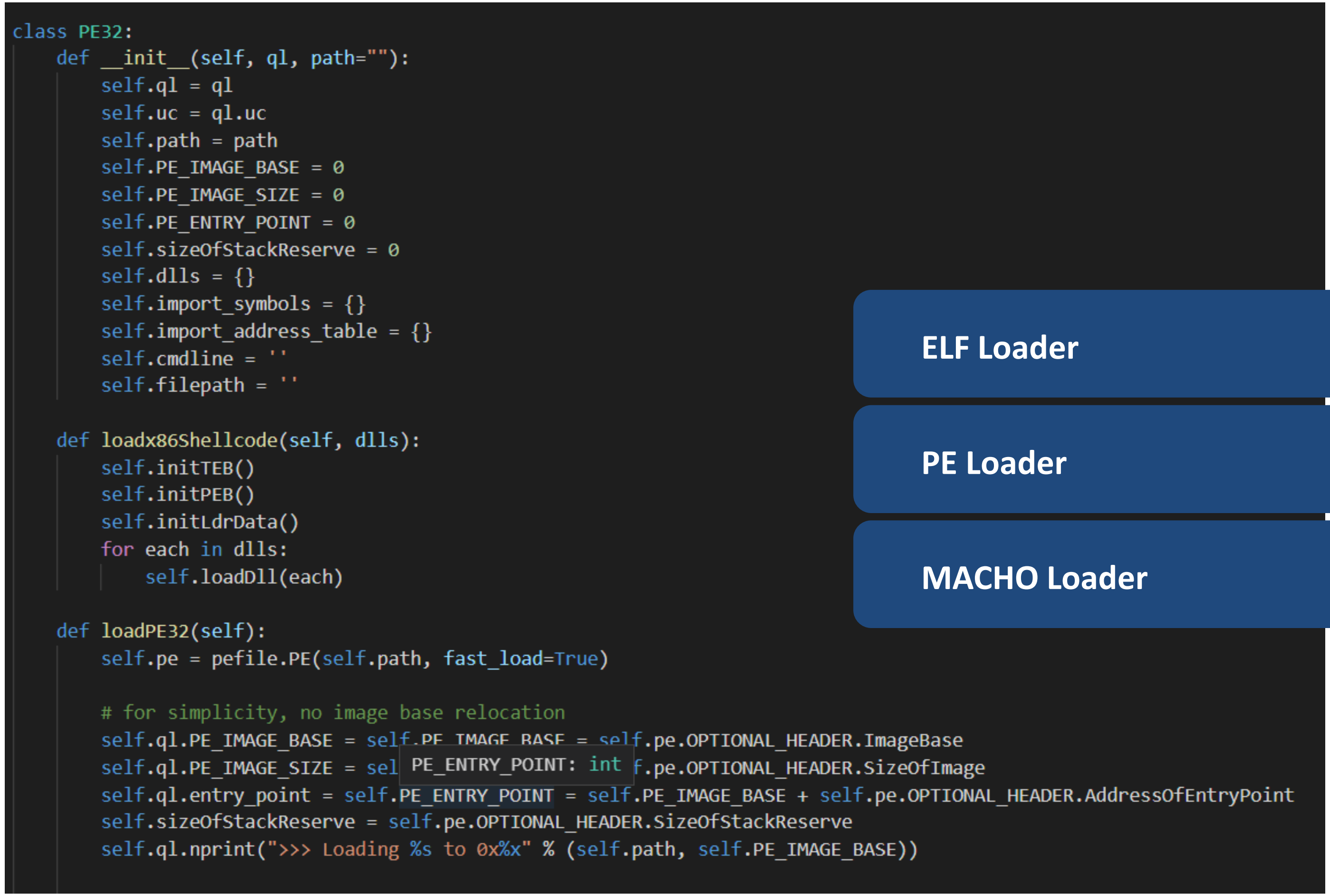
And not limited to ARCH

OS Adventure

Loader



```
class ELFParse:
    def __init__(self, path, ql):
        self.path = path
        self.ql = ql

        with open(path, "rb") as f:
            self.elfdata = f.read()

        self.ident = self.getident()

        if self.ident[ : 4] != b'\x7fELF':
            ql.nprint(">>> ERROR: NOT a ELF")
            exit(1)

        if self.ident[0x4] == 1: # 32 bit
            self.is32bit = True
        else:
            self.is32bit = False

        if self.ident[0x4] == 2: # 64 bit
            self.is64bit = True
        else:
            self.is64bit = False

        if self.ident[0x5] == 1: # little endian
            self.endian = 1
        elif self.ident[0x5] == 2: # big endian
            self.endian = 2
```



```
class PE32:
    def __init__(self, ql, path=""):
        self.ql = ql
        self.uc = ql.uc
        self.path = path
        self.PE_IMAGE_BASE = 0
        self.PE_IMAGE_SIZE = 0
        self.PE_ENTRY_POINT = 0
        self.sizeOfStackReserve = 0
        self.dlls = {}
        self.import_symbols = {}
        self.import_address_table = {}
        self.cmdline = ''
        self.filepath = ''

    def loadx86Shellcode(self, dlls):
        self.initTEB()
        self.initPEB()
        self.initLdrData()
        for each in dlls:
            self.loadDll(each)

    def loadPE32(self):
        self.pe = pefile.PE(self.path, fast_load=True)

        # for simplicity, no image base relocation
        self.ql.PE_IMAGE_BASE = self.PE_IMAGE_BASE = self.pe.OPTIONAL_HEADER.ImageBase
        self.ql.PE_IMAGE_SIZE = sel PE_ENTRY_POINT: int f.pe.OPTIONAL_HEADER.SizeOfImage
        self.ql.entry_point = self.PE_ENTRY_POINT = self.PE_IMAGE_BASE + self.pe.OPTIONAL_HEADER.AddressOfEntryPoint
        self.sizeOfStackReserve = self.pe.OPTIONAL_HEADER.SizeOfStackReserve
        self.ql.nprint(">>> Loading %s to 0x%x" % (self.path, self.PE_IMAGE_BASE))
```

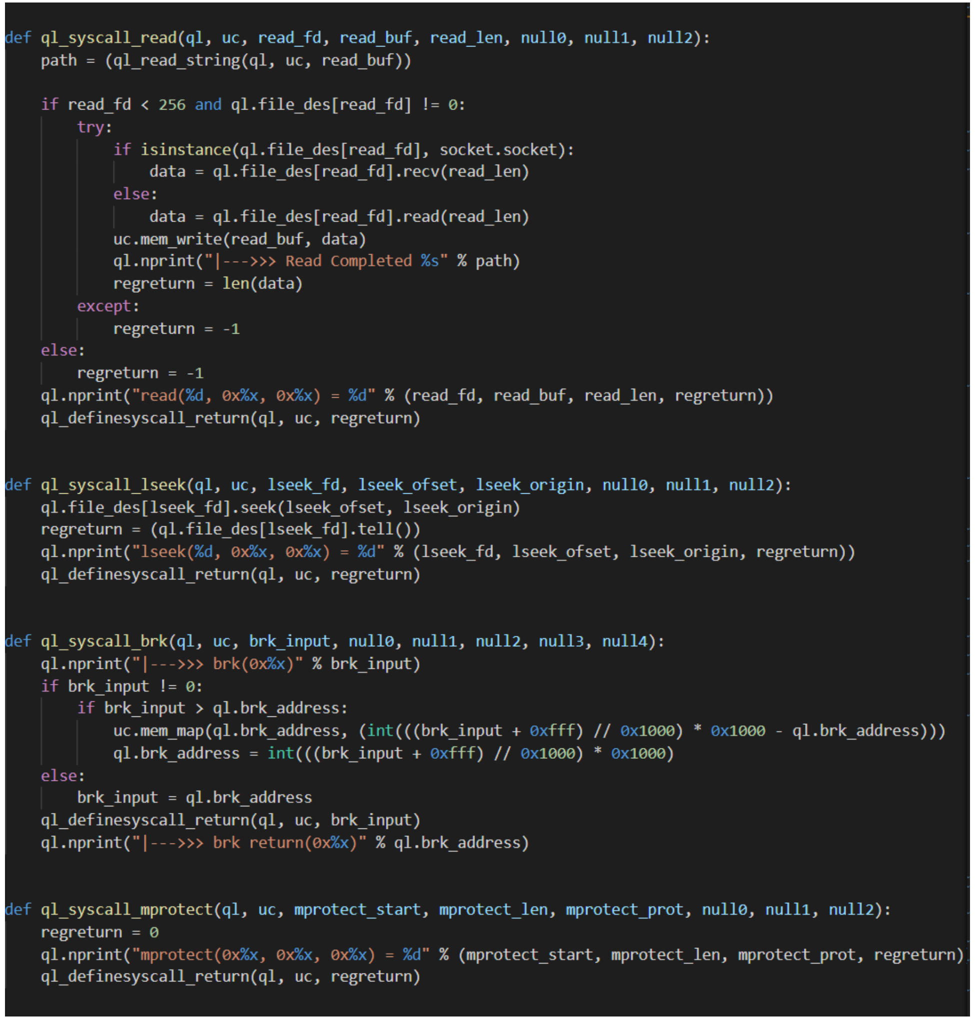
ELF Loader

PE Loader

MACHO Loader

Parse != Loader

Posix Series - Syscall Emulator



```
def ql_syscall_read(ql, uc, read_fd, read_buf, read_len, null0, null1, null2):
    path = (ql_read_string(ql, uc, read_buf))

    if read_fd < 256 and ql.file_des[read_fd] != 0:
        try:
            if isinstance(ql.file_des[read_fd], socket.socket):
                data = ql.file_des[read_fd].recv(read_len)
            else:
                data = ql.file_des[read_fd].read(read_len)
            uc.mem_write(read_buf, data)
            ql.nprint("|--->>> Read Completed %s" % path)
            regreturn = len(data)
        except:
            regreturn = -1
    else:
        regreturn = -1
    ql.nprint("read(%d, 0x%x, 0x%x) = %d" % (read_fd, read_buf, read_len, regreturn))
    ql_definesyscall_return(ql, uc, regreturn)


def ql_syscall_lseek(ql, uc, lseek_fd, lseek_ofset, lseek_origin, null0, null1, null2):
    ql.file_des[lseek_fd].seek(lseek_ofset, lseek_origin)
    regreturn = (ql.file_des[lseek_fd].tell())
    ql.nprint("lseek(%d, 0x%x, 0x%x) = %d" % (lseek_fd, lseek_ofset, lseek_origin, regreturn))
    ql_definesyscall_return(ql, uc, regreturn)


def ql_syscall_brk(ql, uc, brk_input, null0, null1, null2, null3, null4):
    ql.nprint("|--->>> brk(0x%x)" % brk_input)
    if brk_input != 0:
        if brk_input > ql.brk_address:
            uc.mem_map(ql.brk_address, (int(((brk_input + 0xfff) // 0x1000) * 0x1000 - ql.brk_address)))
            ql.brk_address = int(((brk_input + 0xfff) // 0x1000) * 0x1000)
    else:
        brk_input = ql.brk_address
    ql_definesyscall_return(ql, uc, brk_input)
    ql.nprint("|--->>> brk return(0x%x)" % ql.brk_address)


def ql_syscall_mprotect(ql, uc, mprotect_start, mprotect_len, mprotect_prot, null0, null1, null2):
    regreturn = 0
    ql.nprint("mprotect(0x%x, 0x%x, 0x%x) = %d" % (mprotect_start, mprotect_len, mprotect_prot, regreturn)
    ql_definesyscall_return(ql, uc, regreturn)
```

Syscall almost the same for OSX/Linux/*BSD

Kernel Programming 101

Emulate Syscall

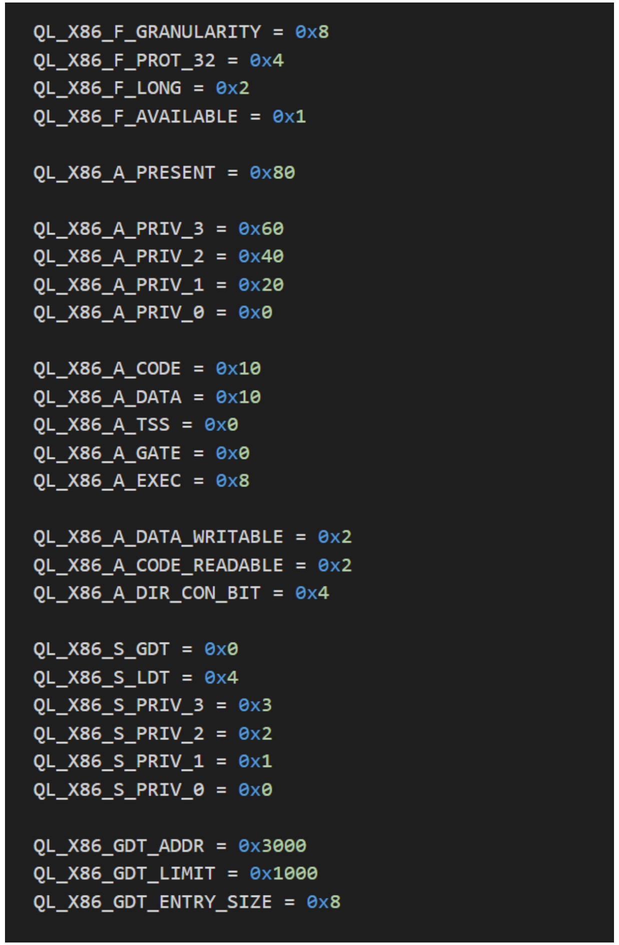
Skip/Forward or Emulate Code

Prepare Execution Report

Syscall Implementation

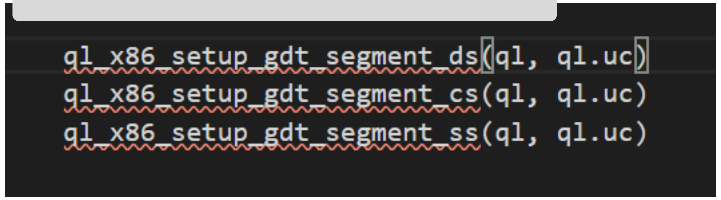
CPU Adventure

X86 32/64 Series



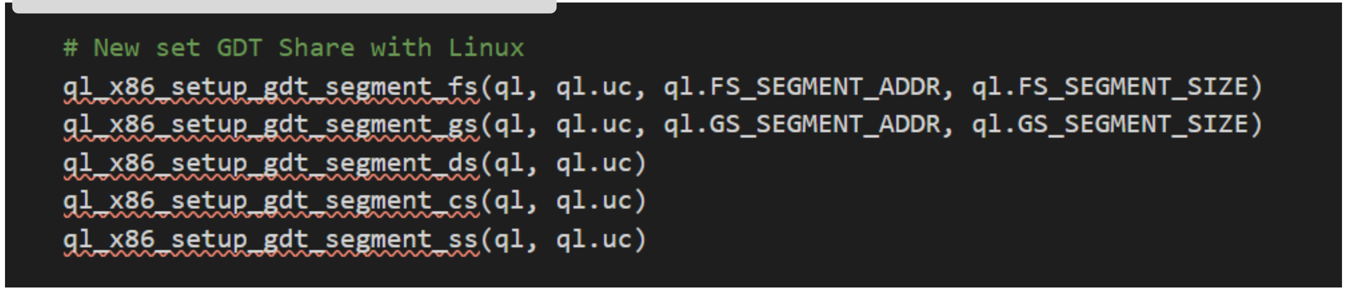
```
QL_X86_F_GRANULARITY = 0x8
QL_X86_F_PROT_32 = 0x4
QL_X86_F_LONG = 0x2
QL_X86_F_AVAILABLE = 0x1

QL_X86_A_PRESENT = 0x80

QL_X86_A_PRIV_3 = 0x60
QL_X86_A_PRIV_2 = 0x40
QL_X86_A_PRIV_1 = 0x20
QL_X86_A_PRIV_0 = 0x0

QL_X86_A_CODE = 0x10
QL_X86_A_DATA = 0x10
QL_X86_A_TSS = 0x0
QL_X86_A_GATE = 0x0
QL_X86_A_EXEC = 0x8

QL_X86_A_DATA_WRITABLE = 0x2
QL_X86_A_CODE_READABLE = 0x2
QL_X86_A_DIR_CON_BIT = 0x4

QL_X86_S_GDT = 0x0
QL_X86_S_LDT = 0x4
QL_X86_S_PRIV_3 = 0x3
QL_X86_S_PRIV_2 = 0x2
QL_X86_S_PRIV_1 = 0x1
QL_X86_S_PRIV_0 = 0x0

QL_X86_GDT_ADDR = 0x3000
QL_X86_GDT_LIMIT = 0x1000
QL_X86_GDT_ENTRY_SIZE = 0x8
```

X86 32/64bit GDT For Linux



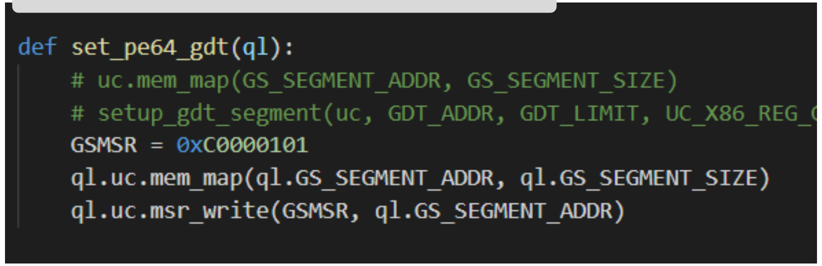
```
ql_x86_setup_gdt_segment_ds(ql, ql.uc)
ql_x86_setup_gdt_segment_cs(ql, ql.uc)
ql_x86_setup_gdt_segment_ss(ql, ql.uc)
```

X86 32bit GDT For Windows



```
# New set GDT Share with Linux
ql_x86_setup_gdt_segment_fs(ql, ql.uc, ql.FS_SEGMENT_ADDR, ql.FS_SEGMENT_SIZE)
ql_x86_setup_gdt_segment_gs(ql, ql.uc, ql.GS_SEGMENT_ADDR, ql.GS_SEGMENT_SIZE)
ql_x86_setup_gdt_segment_ds(ql, ql.uc)
ql_x86_setup_gdt_segment_cs(ql, ql.uc)
ql_x86_setup_gdt_segment_ss(ql, ql.uc)
```

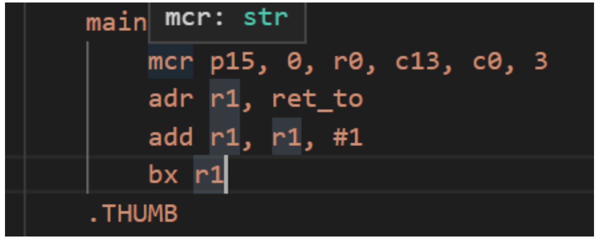
X86 64bit GDT For Windows



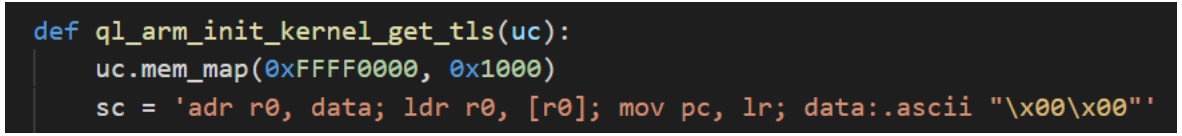
```
def set_pe64_gdt(ql):
    # uc.mem_map(GS_SEGMENT_ADDR, GS_SEGMENT_SIZE)
    # setup_gdt_segment(uc, GDT_ADDR, GDT_LIMIT, UC_X86_REG_
    GSMSR = 0xC0000101
    ql.uc.mem_map(ql.GS_SEGMENT_ADDR, ql.GS_SEGMENT_SIZE)
    ql.uc.msr_write(GSMSR, ql.GS_SEGMENT_ADDR)
```

It took us sometime to fix the GDT and Set Thread Area

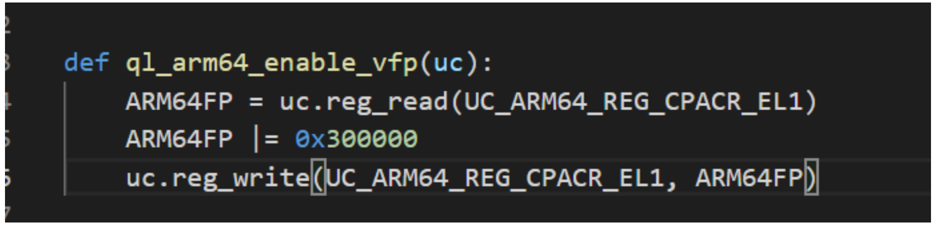
ARM/64 Series



```
    main
        mcr p15, 0, r0, c13, c0, 3
        adr r1, ret_to
        add r1, r1, #1
        bx r1
    .THUMB
```



```
def ql_arm_init_kernel_get_tls(uc):
    uc.mem_map(0xFFFF0000, 0x1000)
    sc = 'adr r0, data; ldr r0, [r0]; mov pc, lr; data:.ascii "\x00\x00"'
```



```
def ql_arm64_enable_vfp(uc):
    ARM64FP = uc.reg_read(UC_ARM64_REG_CPACR_EL1)
    ARM64FP |= 0x300000
    uc.reg_write(UC_ARM64_REG_CPACR_EL1, ARM64FP)
```

ARM/Thumb and ARM64

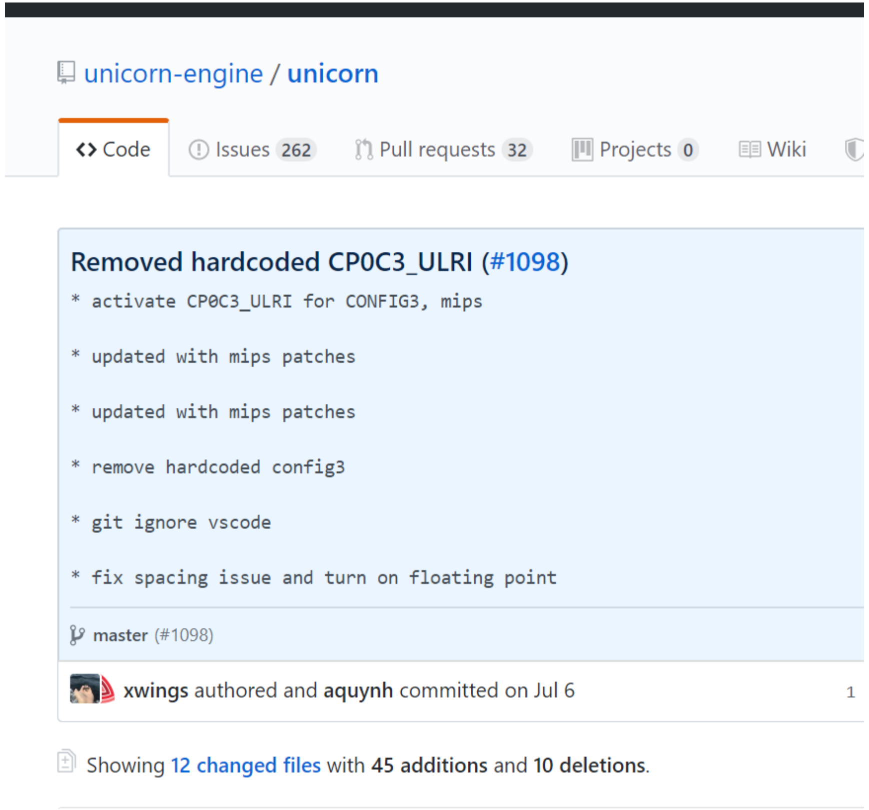
Making Sure Loader is compatible

ARM MCR instruction for Set TLS

ARM Kernel Initialization

ARM and ARM64 Enable VFP

MIPS32EL Series



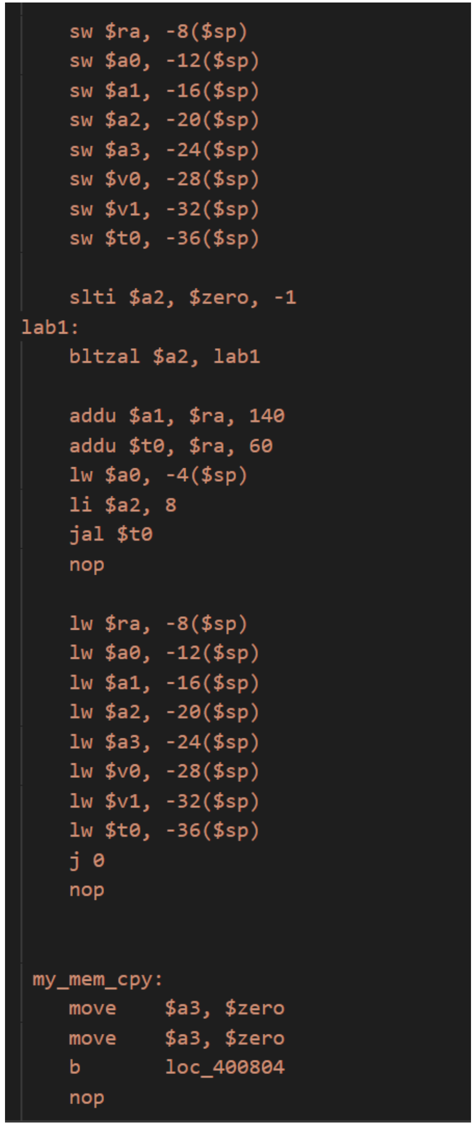
Removed hardcoded CP0C3_ULRI (#1098)

```
* activate CP0C3_ULRI for CONFIG3, mips

* updated with mips patches

* updated with mips patches

* remove hardcoded config3

* git ignore vscode

* fix spacing issue and turn on floating point
```

master (#1098)

xwings authored and aquynh committed on Jul 6 1

Showing 12 changed files with 45 additions and 10 deletions.



```
    sw $ra, -8($sp)
    sw $a0, -12($sp)
    sw $a1, -16($sp)
    sw $a2, -20($sp)
    sw $a3, -24($sp)
    sw $v0, -28($sp)
    sw $v1, -32($sp)
    sw $t0, -36($sp)

    slti $a2, $zero, -1
lab1:
    bltzal $a2, lab1

    addu $a1, $ra, 140
    addu $t0, $ra, 60
    lw $a0, -4($sp)
    li $a2, 8
    jal $t0
    nop

    lw $ra, -8($sp)
    lw $a0, -12($sp)
    lw $a1, -16($sp)
    lw $a2, -20($sp)
    lw $a3, -24($sp)
    lw $v0, -28($sp)
    lw $v1, -32($sp)
    lw $t0, -36($sp)
    j 0
    nop


 my_mem_cpy:
    move    $a3, $zero
    move    $a3, $zero
    b       loc_400804
    nop
```

MIPS Comes with CO Processor

Configuration needed for CO Processor

Unicorn does not support Floating Point

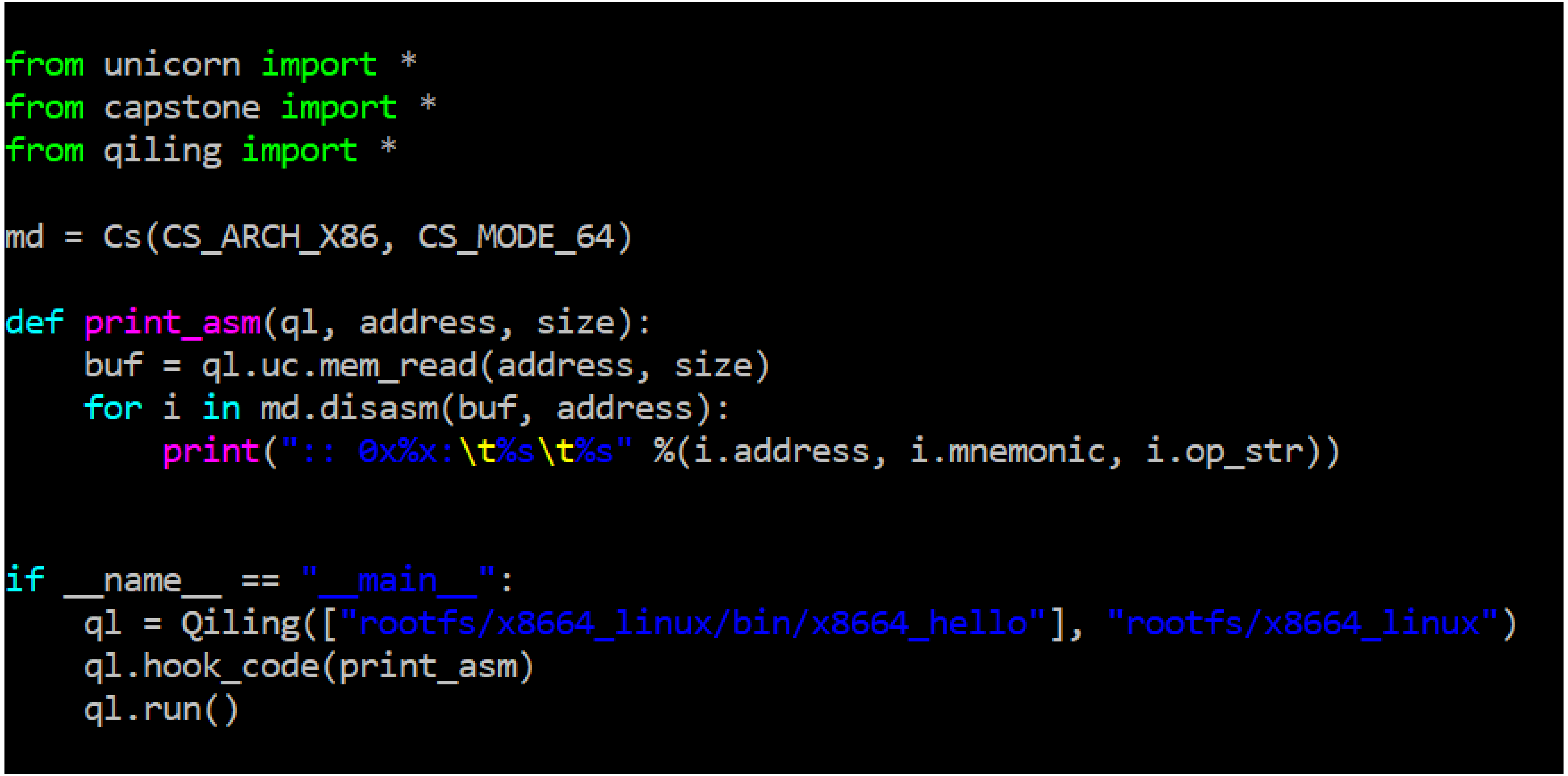
Patch Unicorn to Support CO Processors

Custom Binary Injected for Set Thread Area

Applications of Qiling

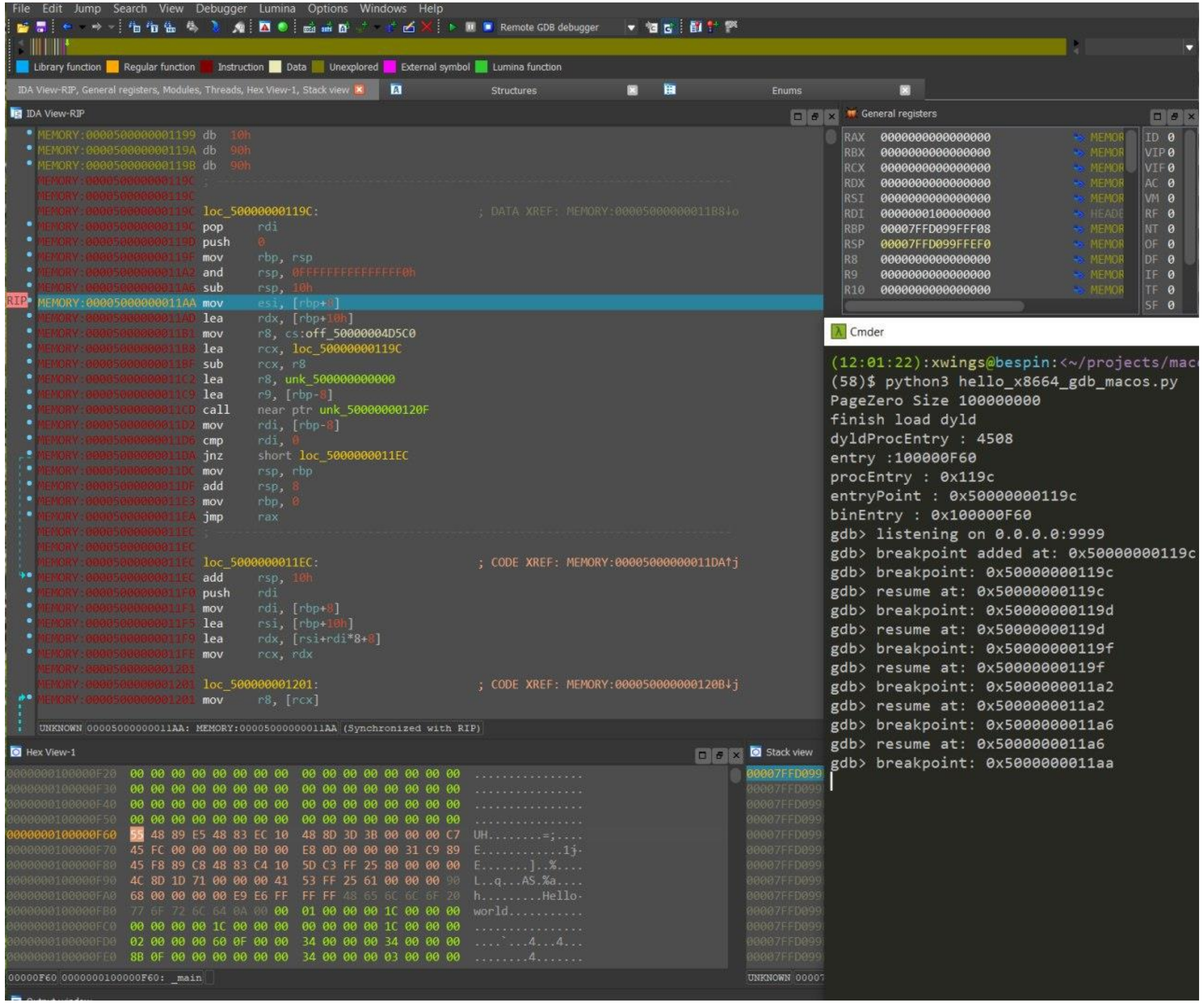
Build dynamic analysis tools – Basic ++

- Let Qiling loads the binary (loading + dynamic linking)
- Syscall & system API logging available, provided by default
- Program callbacks with Qiling hook capabilities: hook memory access, hook address range
- Repeat in a loop: run() → analysis → resume()



```
from unicorn import *
from capstone import *
from qiling import *

md = Cs(CS_ARCH_X86, CS_MODE_64)

def print_asm(ql, address, size):
    buf = ql.uc.mem_read(address, size)
    for i in md.disasm(buf, address):
        print(":: 0x%x:\t%s\t%s" %(i.address, i.mnemonic, i.op_str))


if __name__ == "__main__":
    ql = Qiling(["rootfs/x8664_linux/bin/x8664_hello"], "rootfs/x8664_linux")
    ql.hook_code(print_asm)
    ql.run()
```

Debugger – GDB / IDAPro/ r2



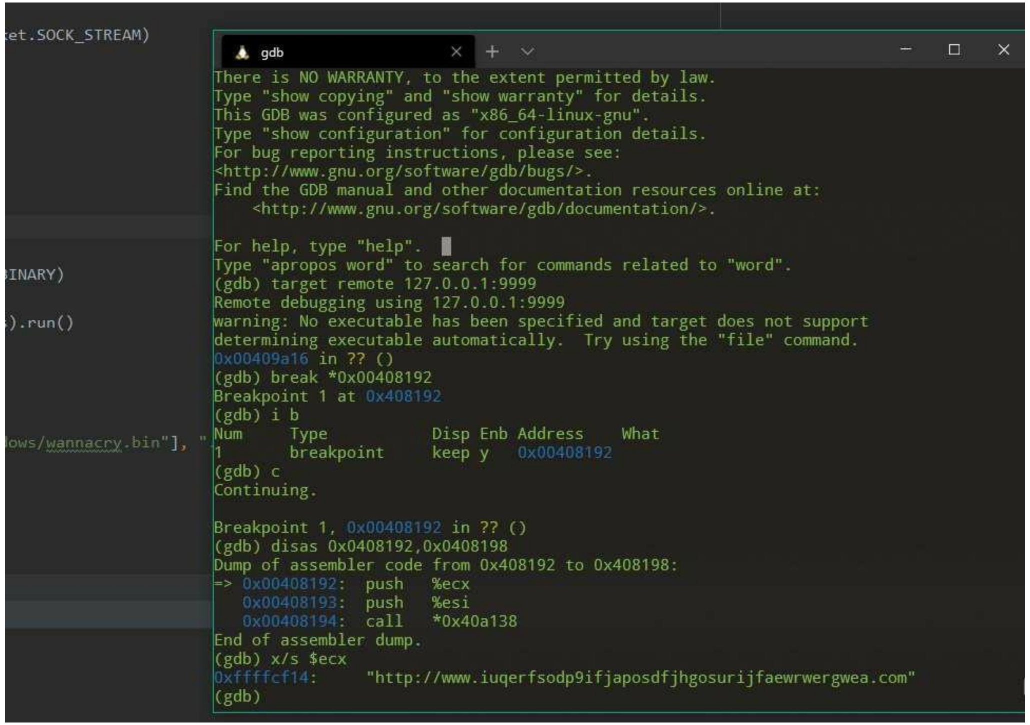
```
MEMORY:0000500000001199 db  10h
MEMORY:000050000000119A db  90h
MEMORY:000050000000119B db  90h
MEMORY:000050000000119C ; ---------------------------------------------------------------
MEMORY:000050000000119C
MEMORY:000050000000119C loc_50000000119C:                      ; DATA XREF: MEMORY:00005000000011B8↓o
MEMORY:000050000000119C pop     rdi
MEMORY:000050000000119D push    0
MEMORY:000050000000119F mov     rbp, rsp
MEMORY:00005000000011A2 and     rsp, 0FFFFFFFFFFFFFFF0h
MEMORY:00005000000011A6 sub     rsp, 10h
RIP MEMORY:00005000000011AA mov     esi, [rbp+8]
MEMORY:00005000000011AD lea     rdx, [rbp+10h]
MEMORY:00005000000011B1 mov     r8, cs:off_50000004D5C0
MEMORY:00005000000011B8 lea     rcx, loc_50000000119C
MEMORY:00005000000011BF sub     rcx, r8
MEMORY:00005000000011C2 lea     r8, unk_500000000000
MEMORY:00005000000011C9 lea     r9, [rbp-8]
MEMORY:00005000000011CD call    near ptr unk_50000000120F
MEMORY:00005000000011D2 mov     rdi, [rbp-8]
MEMORY:00005000000011D6 cmp     rdi, 0
MEMORY:00005000000011DA jnz     short loc_5000000011EC
MEMORY:00005000000011DC mov     rsp, rbp
MEMORY:00005000000011DF add     rsp, 8
MEMORY:00005000000011E3 mov     rbp, 0
MEMORY:00005000000011EA jmp     rax
MEMORY:00005000000011EC ; ---------------------------------------------------------------
MEMORY:00005000000011EC
MEMORY:00005000000011EC loc_5000000011EC:                      ; CODE XREF: MEMORY:00005000000011DA↑j
MEMORY:00005000000011EC add     rsp, 10h
MEMORY:00005000000011F0 push    rdi
MEMORY:00005000000011F1 mov     rdi, [rbp+8]
MEMORY:00005000000011F5 lea     rsi, [rbp+10h]
MEMORY:00005000000011F9 lea     rdx, [rsi+rdi*8+8]
MEMORY:00005000000011FE mov     rcx, rdx
MEMORY:0000500000001201
MEMORY:0000500000001201 loc_500000001201:                      ; CODE XREF: MEMORY:000050000000120B↓j
MEMORY:0000500000001201 mov     r8, [rcx]
UNKNOWN 00005000000011AA: MEMORY:00005000000011AA (Synchronized with RIP)
```

```
RAX  0000000000000000   MEMOR   ID  0
RBX  0000000000000000   MEMOR   VIP 0
RCX  0000000000000000   MEMOR   VIF 0
RDX  0000000000000000   MEMOR   AC  0
RSI  0000000000000000   MEMOR   VM  0
RDI  0000000100000000   HEADE   RF  0
RBP  00007FFD099FFF08   MEMOR   NT  0
RSP  00007FFD099FFEF0   MEMOR   OF  0
R8   0000000000000000   MEMOR   DF  0
R9   0000000000000000   MEMOR   IF  0
R10  0000000000000000   MEMOR   TF  0
                                SF  0
```

```
0000000100000F20  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000000100000F30  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000000100000F40  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000000100000F50  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000000100000F60  55 48 89 E5 48 83 EC 10  48 8D 3D 3B 00 00 00 C7  UH.......=;.....
0000000100000F70  45 FC 00 00 00 00 B0 00  E8 0D 00 00 00 31 C9 89  E............1j·
0000000100000F80  45 F8 89 C8 48 83 C4 10  5D C3 FF 25 80 00 00 00  E.......]..%....
0000000100000F90  4C 8D 1D 71 00 00 00 41  53 FF 25 61 00 00 00 90  L..q...AS.%a....
0000000100000FA0  68 00 00 00 00 E9 E6 FF  FF FF 48 65 6C 6C 6F 20  h.........Hello·
0000000100000FB0  77 6F 72 6C 64 0A 00 00  01 00 00 00 1C 00 00 00  world...........
0000000100000FC0  00 00 00 00 1C 00 00 00  00 00 00 00 1C 00 00 00  ................
0000000100000FD0  02 00 00 00 60 0F 00 00  34 00 00 00 34 00 00 00  ....`...4...4...
0000000100000FE0  8B 0F 00 00 00 00 00 00  34 00 00 00 03 00 00 00  ........4.......
00000F60 0000000100000F60: _main
```

```
00007FFD099
00007FFD099
00007FFD099
00007FFD099
00007FFD099
00007FFD099
00007FFD099
00007FFD099
00007FFD099
00007FFD099
00007FFD099
00007FFD099
00007FFD099
UNKNOWN 00007
```

```
(12:01:22):xwings@bespin:<~/projects/mac
(58)$ python3 hello_x8664_gdb_macos.py
PageZero Size 100000000
finish load dyld
dyldProcEntry : 4508
entry :100000F60
procEntry : 0x119c
entryPoint : 0x50000000119c
binEntry : 0x100000F60
gdb> listening on 0.0.0.0:9999
gdb> breakpoint added at: 0x50000000119c
gdb> breakpoint: 0x50000000119c
gdb> resume at: 0x50000000119c
gdb> breakpoint: 0x50000000119d
gdb> resume at: 0x50000000119d
gdb> breakpoint: 0x50000000119f
gdb> resume at: 0x50000000119f
gdb> breakpoint: 0x5000000011a2
gdb> resume at: 0x5000000011a2
gdb> breakpoint: 0x5000000011a6
gdb> resume at: 0x5000000011a6
gdb> breakpoint: 0x5000000011aa
```



```
et.SOCK_STREAM)
INARY)
).run()
ows/wannacry.bin"], "
```

```
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word".
(gdb) target remote 127.0.0.1:9999
Remote debugging using 127.0.0.1:9999
warning: No executable has been specified and target does not support
determining executable automatically.  Try using the "file" command.
0x00409a16 in ?? ()
(gdb) break *0x00408192
Breakpoint 1 at 0x408192
(gdb) i b
Num     Type           Disp Enb Address    What
1       breakpoint     keep y   0x00408192
(gdb) c
Continuing.

Breakpoint 1, 0x00408192 in ?? ()
(gdb) disas 0x0408192,0x0408198
Dump of assembler code from 0x408192 to 0x408198:
=> 0x00408192:  push   %ecx
   0x00408193:  push   %esi
   0x00408194:  call   *0x40a138
End of assembler dump.
(gdb) x/s $ecx
0xffffcf14:     "http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com"
(gdb)
```

Guided fuzzer – cross platform/architecture

- Cross platform/architecture: Windows, MacOS, Linux, BSD on X86, Arm, Arm64, Mips
- https://github.com/qilingframework/qiling/tree/dev/examples/fuzzing

←

Dominik
9,790 Tweets

ooo Follow

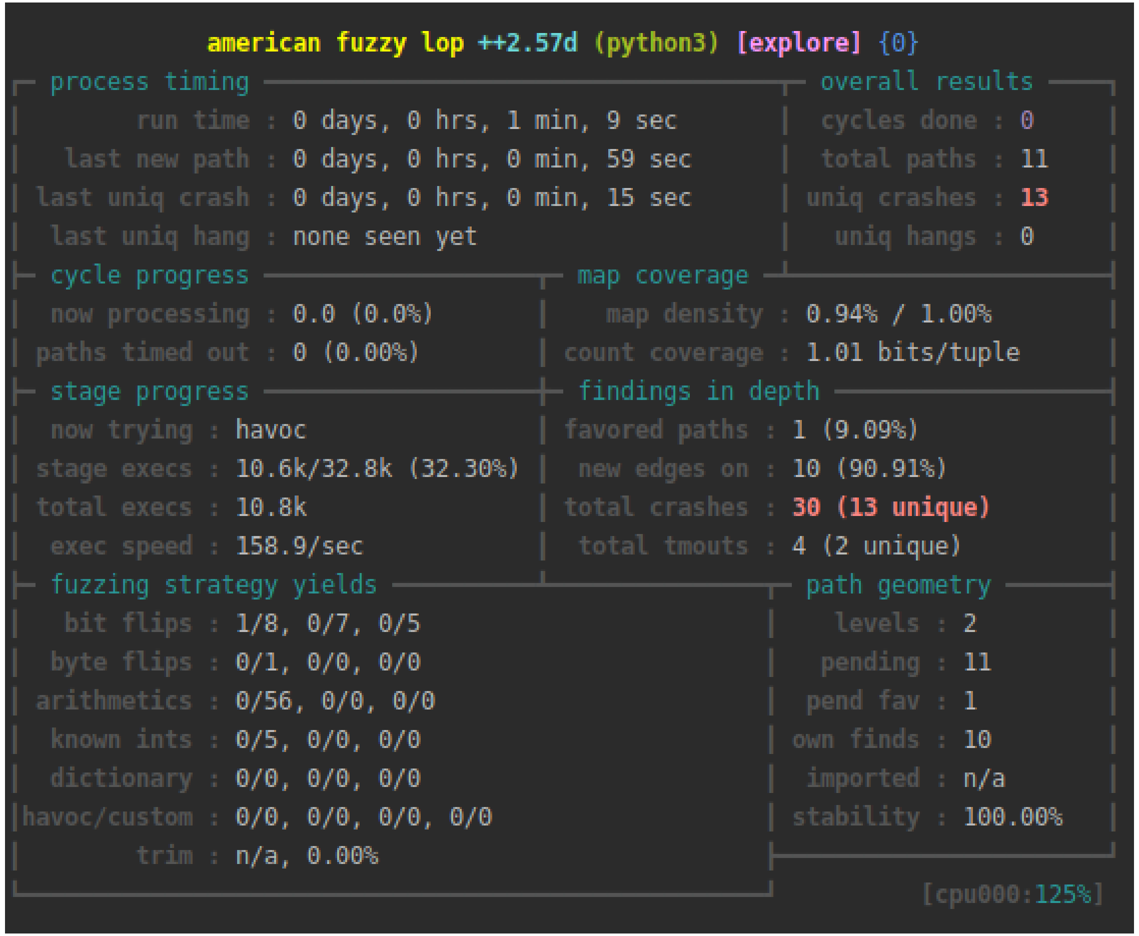
Dominik
@domenuk

【DΞCOMPILΞ NΣVΞR】
Infosec guy, researcher, securing CPE
How do I know you're procrastinating right now?
aflplus.plus @enoflag

Deutschland Joined April 2009

962 Following **1,540** Followers

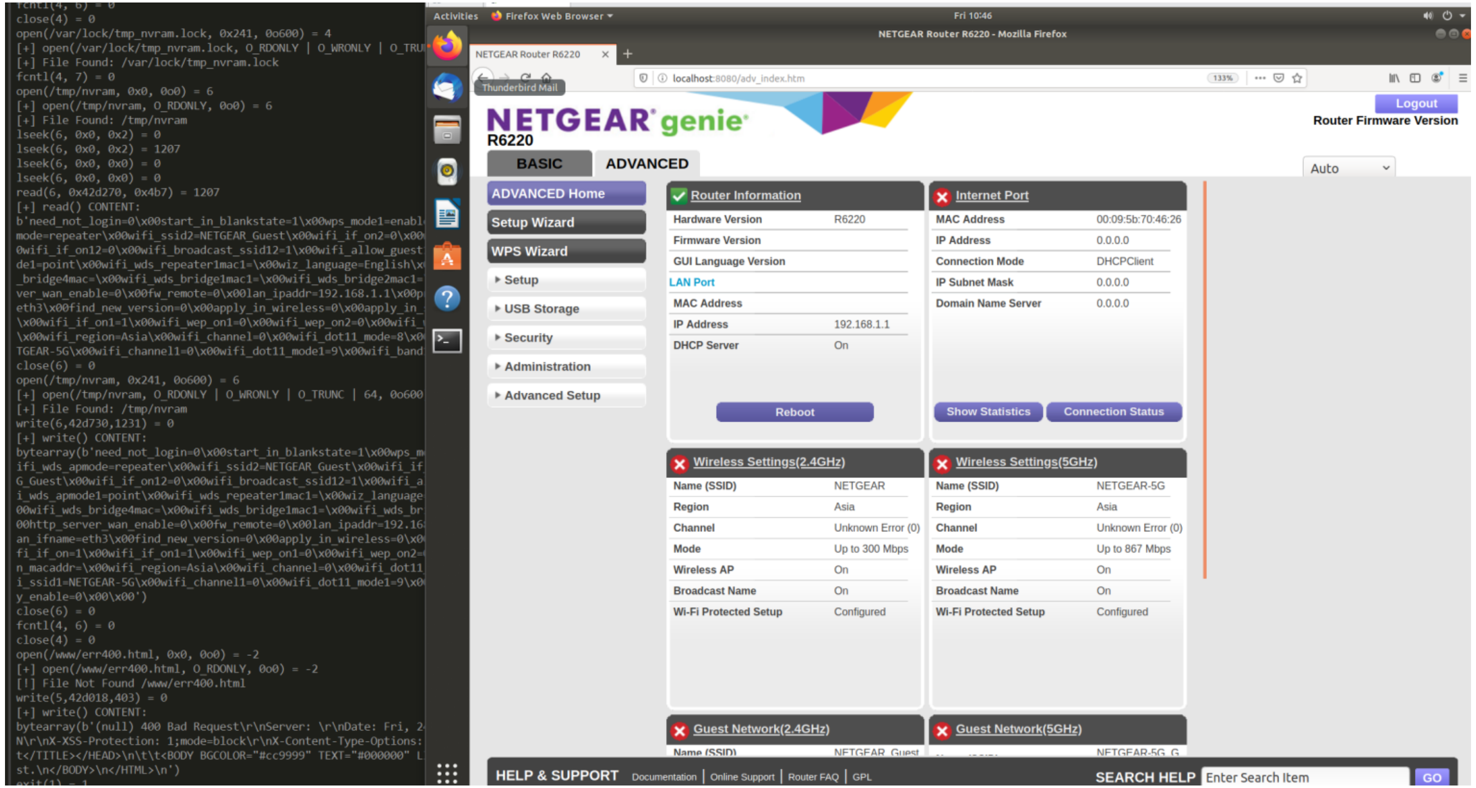
Not followed by anyone you're following



```
          american fuzzy lop ++2.57d (python3) [explore] {0}
┌─ process timing ────────────────────────────────────┬─ overall results ────┐
│        run time : 0 days, 0 hrs, 1 min, 9 sec       │  cycles done : 0     │
│   last new path : 0 days, 0 hrs, 0 min, 59 sec      │  total paths : 11    │
│ last uniq crash : 0 days, 0 hrs, 0 min, 15 sec      │ uniq crashes : 13    │
│  last uniq hang : none seen yet                     │   uniq hangs : 0     │
├─ cycle progress ───────────────────┬─ map coverage ─┴──────────────────────┤
│  now processing : 0.0 (0.0%)       │    map density : 0.94% / 1.00%        │
│ paths timed out : 0 (0.00%)        │ count coverage : 1.01 bits/tuple      │
├─ stage progress ───────────────────┼─ findings in depth ───────────────────┤
│  now trying : havoc                │ favored paths : 1 (9.09%)             │
│ stage execs : 10.6k/32.8k (32.30%) │  new edges on : 10 (90.91%)           │
│ total execs : 10.8k                │ total crashes : 30 (13 unique)        │
│  exec speed : 158.9/sec            │  total tmouts : 4 (2 unique)          │
├─ fuzzing strategy yields ──────────┴───────────┬─ path geometry ───────────┤
│   bit flips : 1/8, 0/7, 0/5                    │    levels : 2             │
│  byte flips : 0/1, 0/0, 0/0                    │   pending : 11            │
│ arithmetics : 0/56, 0/0, 0/0                   │  pend fav : 1             │
│  known ints : 0/5, 0/0, 0/0                    │ own finds : 10            │
│  dictionary : 0/0, 0/0, 0/0                    │  imported : n/a           │
│havoc/custom : 0/0, 0/0, 0/0, 0/0               │ stability : 100.00%       │
│        trim : n/a, 0.00%                       ├───────────────────────────┘
└────────────────────────────────────────────────┘          [cpu000:125%]
```

Firmware analysis

- Emulation offers a chance to move analysis to a much more powerful platform
- Emulate a single binary is better than whole firmware
  - Hardware emulation is tough without hardware specs
  - Series of different firmware can share the same target binary
- Challenges
  - Dump firmware, or extract firmware from binary blob
  - Extract the target binary
  - NVRAM emulation
  - Dependency libraries
  - Presence of other devices: wireless interface



```
fcntl(4, 6) = 0
close(4) = 0
open(/var/lock/tmp_nvram.lock, 0x241, 0o600) = 4
[+] open(/var/lock/tmp_nvram.lock, O_RDONLY | O_WRONLY | O_TRU
[+] File Found: /var/lock/tmp_nvram.lock
fcntl(4, 7) = 0
open(/tmp/nvram, 0x0, 0o0) = 6
[+] open(/tmp/nvram, O_RDONLY, 0o0) = 6
[+] File Found: /tmp/nvram
lseek(6, 0x0, 0x2) = 0
lseek(6, 0x0, 0x2) = 1207
lseek(6, 0x0, 0x0) = 0
lseek(6, 0x0, 0x0) = 0
read(6, 0x42d270, 0x4b7) = 1207
[+] read() CONTENT:
b'need_not_login=0\x00start_in_blankstate=1\x00wps_mode1=enabl
mode=repeater\x00wifi_ssid2=NETGEAR_Guest\x00wifi_if_on2=0\x00
0wifi_if_on12=0\x00wifi_broadcast_ssid12=1\x00wifi_allow_guest
de1=point\x00wifi_wds_repeater1mac1=\x00wiz_language=English\x
_bridge4mac=\x00wifi_wds_bridge1mac1=\x00wifi_wds_bridge2mac1=
ver_wan_enable=0\x00fw_remote=0\x00lan_ipaddr=192.168.1.1\x00p
eth3\x00find_new_version=0\x00apply_in_wireless=0\x00apply_in_
\x00wifi_if_on1=1\x00wifi_wep_on1=0\x00wifi_wep_on2=0\x00wifi_
\x00wifi_region=Asia\x00wifi_channel=0\x00wifi_dot11_mode=8\x0
TGEAR-5G\x00wifi_channel1=0\x00wifi_dot11_mode1=9\x00wifi_band
close(6) = 0
open(/tmp/nvram, 0x241, 0o600) = 6
[+] open(/tmp/nvram, O_RDONLY | O_WRONLY | O_TRUNC | 64, 0o600
[+] File Found: /tmp/nvram
write(6,42d730,1231) = 0
[+] write() CONTENT:
bytearray(b'need_not_login=0\x00start_in_blankstate=1\x00wps_m
ifi_wds_apmode=repeater\x00wifi_ssid2=NETGEAR_Guest\x00wifi_if
G_Guest\x00wifi_if_on12=0\x00wifi_broadcast_ssid12=1\x00wifi_a
i_wds_apmode1=point\x00wifi_wds_repeater1mac1=\x00wiz_language
00wifi_wds_bridge4mac=\x00wifi_wds_bridge1mac1=\x00wifi_wds_br
00http_server_wan_enable=0\x00fw_remote=0\x00lan_ipaddr=192.16
an_ifname=eth3\x00find_new_version=0\x00apply_in_wireless=0\x0
fi_if_on=1\x00wifi_if_on1=1\x00wifi_wep_on1=0\x00wifi_wep_on2=
n_macaddr=\x00wifi_region=Asia\x00wifi_channel=0\x00wifi_dot11
i_ssid1=NETGEAR-5G\x00wifi_channel1=0\x00wifi_dot11_mode1=9\x0
y_enable=0\x00\x00')
close(6) = 0
fcntl(4, 6) = 0
close(4) = 0
open(/www/err400.html, 0x0, 0o0) = -2
[+] open(/www/err400.html, O_RDONLY, 0o0) = -2
[!] File Not Found /www/err400.html
write(5,42d018,403) = 0
[+] write() CONTENT:
bytearray(b'(null) 400 Bad Request\r\nServer: \r\nDate: Fri, 2
N\r\nX-XSS-Protection: 1;mode=block\r\nX-Content-Type-Options:
t</TITLE></HEAD>\n\t\t<BODY BGCOLOR="#cc9999" TEXT="#000000" L
st.\n</BODY>\n</HTML>\n')
exit(1) = 1
```

NETGEAR Router R6220 - Mozilla Firefox

localhost:8080/adv_index.htm

NETGEAR genie R6220

Router Firmware Version

Router Information

Hardware Version R6220
Firmware Version
GUI Language Version
LAN Port
MAC Address
IP Address 192.168.1.1
DHCP Server On

Internet Port

MAC Address 00:09:5b:70:46:26
IP Address 0.0.0.0
Connection Mode DHCPClient
IP Subnet Mask 0.0.0.0
Domain Name Server 0.0.0.0

Wireless Settings(2.4GHz)

Name (SSID) NETGEAR
Region Asia
Channel Unknown Error (0)
Mode Up to 300 Mbps
Wireless AP On
Broadcast Name On
Wi-Fi Protected Setup Configured

Wireless Settings(5GHz)

Name (SSID) NETGEAR-5G
Region Asia
Channel Unknown Error (0)
Mode Up to 867 Mbps
Wireless AP On
Broadcast Name On
Wi-Fi Protected Setup Configured

Guest Network(2.4GHz)

Name (SSID) NETGEAR_Guest

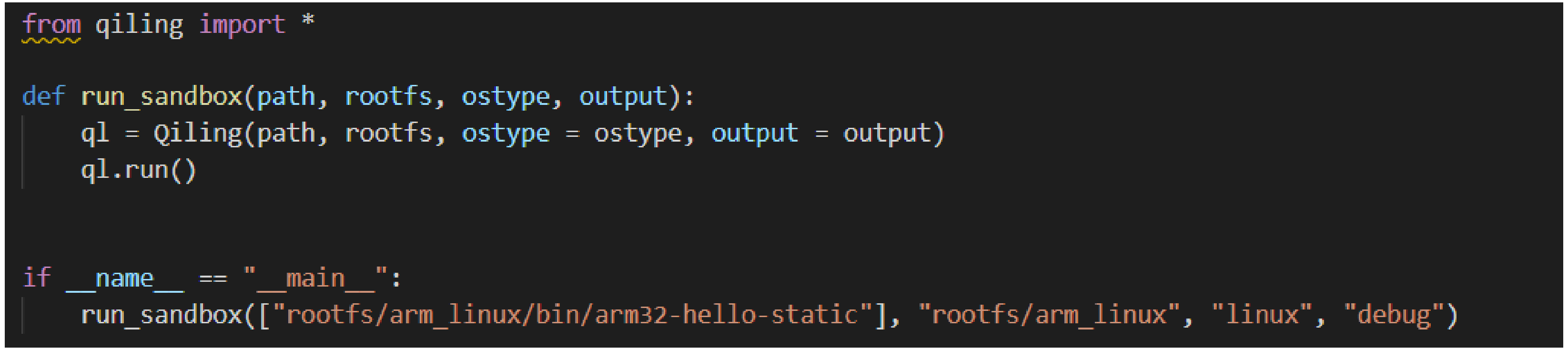
Guest Network(5GHz)

NETGEAR-5G_G

Demo Setup

VirtualBox or VMware

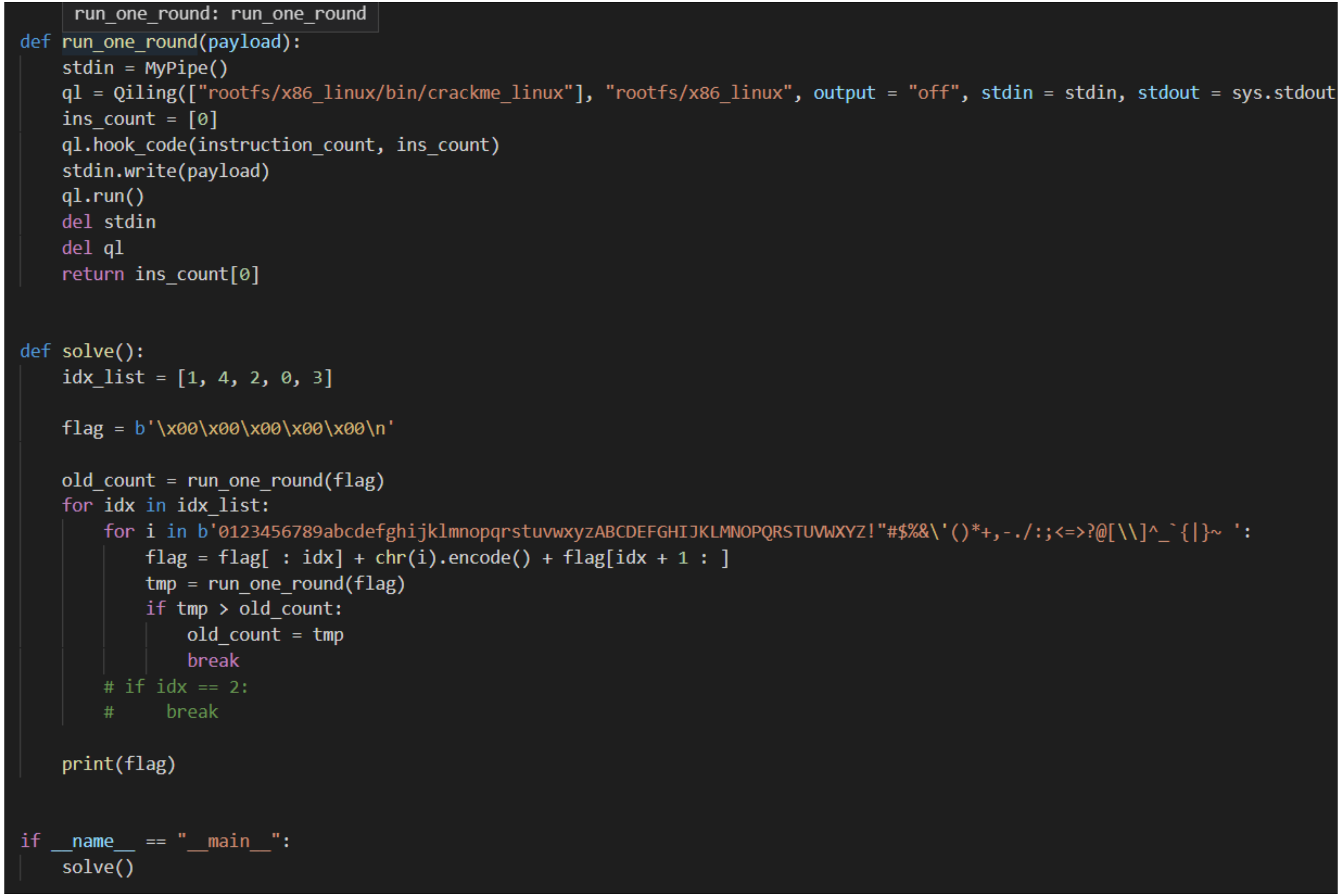
ARM HelloWorld



```
from qiling import *

def run_sandbox(path, rootfs, ostype, output):
    ql = Qiling(path, rootfs, ostype = ostype, output = output)
    ql.run()


if __name__ == "__main__":
    run_sandbox(["rootfs/arm_linux/bin/arm32-hello-static"], "rootfs/arm_linux", "linux", "debug")
```

Debug Mode

Simple Crackme Challenge

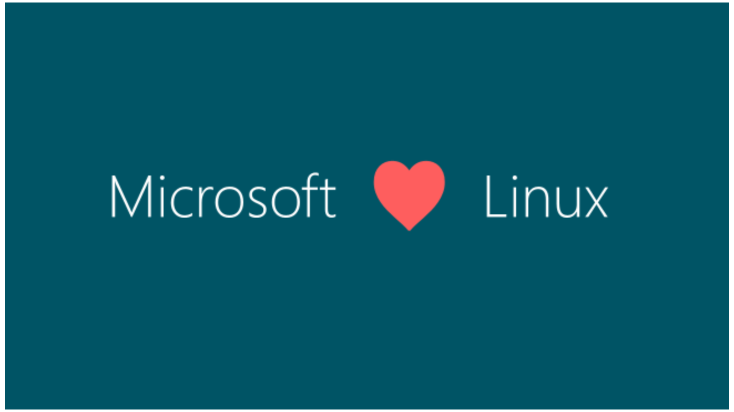


```
    run_one_round: run_one_round
def run_one_round(payload):
    stdin = MyPipe()
    ql = Qiling(["rootfs/x86_linux/bin/crackme_linux"], "rootfs/x86_linux", output = "off", stdin = stdin, stdout = sys.stdout
    ins_count = [0]
    ql.hook_code(instruction_count, ins_count)
    stdin.write(payload)
    ql.run()
    del stdin
    del ql
    return ins_count[0]


def solve():
    idx_list = [1, 4, 2, 0, 3]

    flag = b'\x00\x00\x00\x00\x00\n'

    old_count = run_one_round(flag)
    for idx in idx_list:
        for i in b'0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~ ':
            flag = flag[ : idx] + chr(i).encode() + flag[idx + 1 : ]
            tmp = run_one_round(flag)
            if tmp > old_count:
                old_count = tmp
                break
        # if idx == 2:
        #     break

    print(flag)


if __name__ == "__main__":
    solve()
```

Brute Forcer

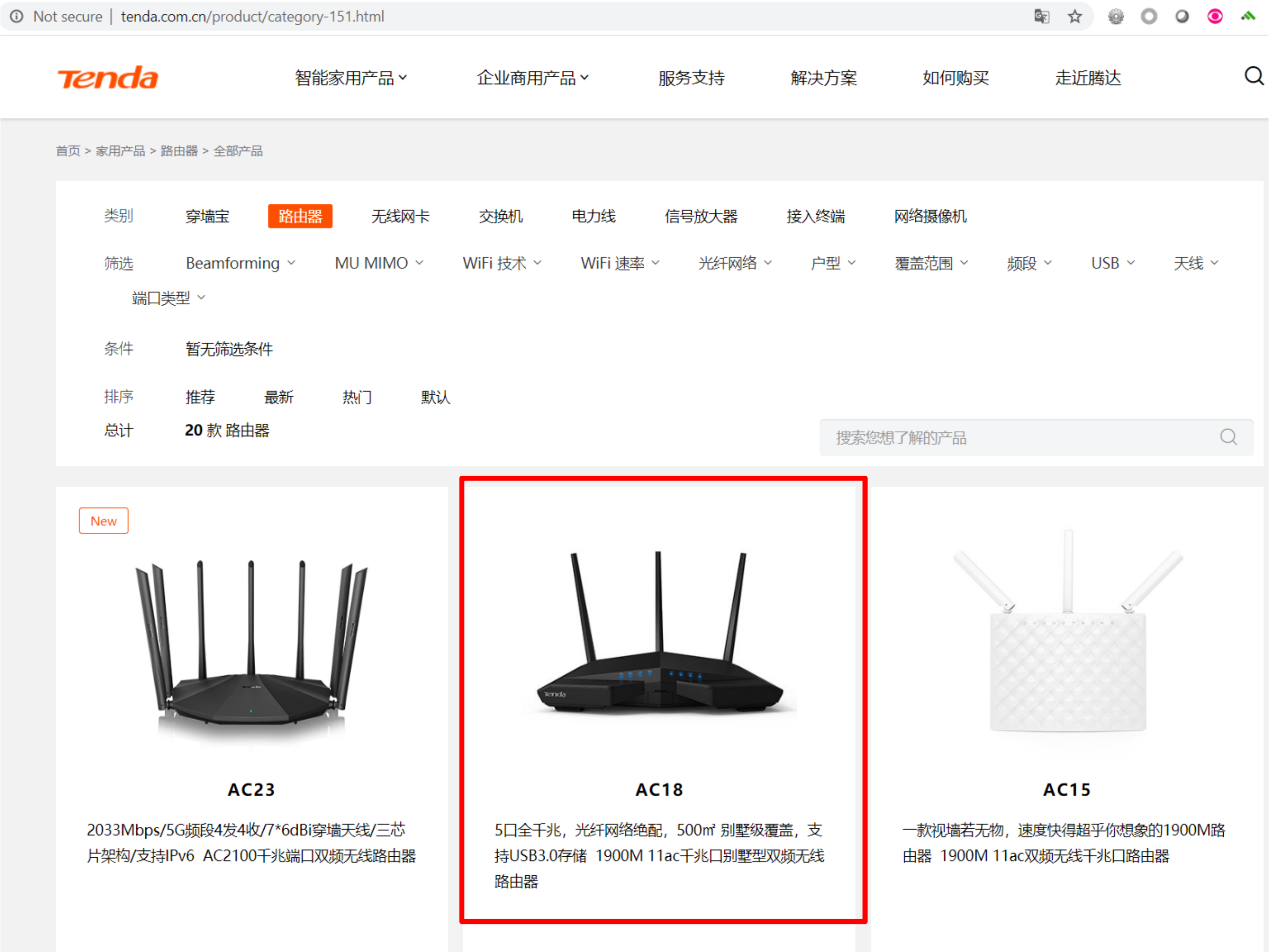
Qiling: Hands On Time

Training Setup



- Required OS
  - Ubuntu 18.04 / 20.04
  - WSL2
- Installation
  - sudo apt-get update
  - sudo apt-get upgrade
  - sudo apt install python3-pip git cmake build-essential libtool-bin python3-dev automake flex bison libglib2.0-dev libpixman-1-dev clang python3-setuptools llvm
  - pip3 install qiling **OR** git clone git@github.com:qilingframework/qiling.git
- Install AFL++
  - git clone https://github.com/AFLplusplus/AFLplusplus.git
  - cd AFLplusplus
  - make
  - cd unicorn_mode
  - ./build_unicorn_support.sh

Emulate a Router



Not secure | tenda.com.cn/product/category-151.html

Tenda 智能家用产品 企业商用产品 服务支持 解决方案 如何购买 走近腾达

首页 > 家用产品 > 路由器 > 全部产品

类别 穿墙宝 路由器 无线网卡 交换机 电力线 信号放大器 接入终端 网络摄像机

筛选 Beamforming MU MIMO WiFi 技术 WiFi 速率 光纤网络 户型 覆盖范围 频段 USB 天线 端口类型

条件 暂无筛选条件

排序 推荐 最新 热门 默认

总计 **20** 款 路由器

搜索您想了解的产品

New

AC23

2033Mbps/5G频段4发4收/7*6dBi穿墙天线/三芯片架构/支持IPv6 AC2100千兆端口双频无线路由器

AC18

5口全千兆，光纤网络绝配，500㎡ 别墅级覆盖，支持USB3.0存储 1900M 11ac千兆口别墅型双频无线路由器

AC15

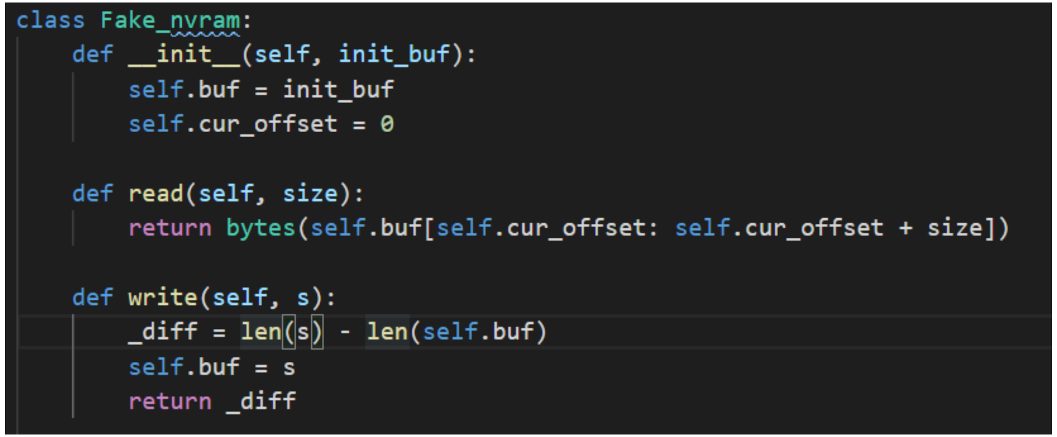
一款视墙若无物，速度快得超乎你想象的1900M路由器 1900M 11ac双频无线千兆口路由器

https://github.com/zsjevilhex/iot/tree/master/route/tenda

Device Emulation

Devices

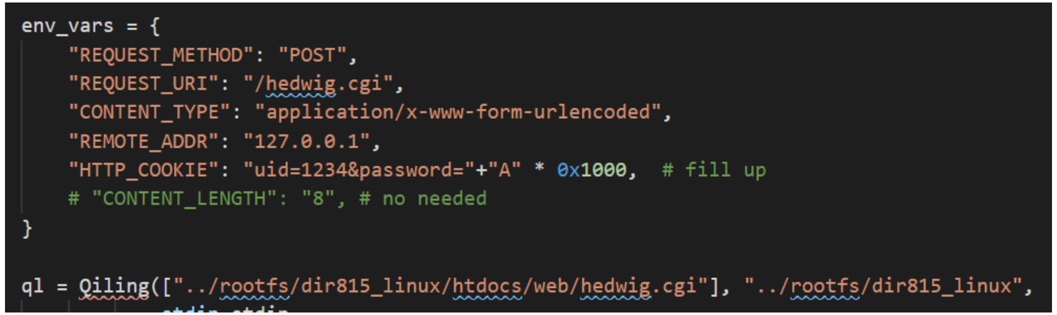
- Read and write emulation for /dev/<devices>
- Able to input custom feedback towards Qiling



```
class Fake_nvram:
    def __init__(self, init_buf):
        self.buf = init_buf
        self.cur_offset = 0

    def read(self, size):
        return bytes(self.buf[self.cur_offset: self.cur_offset + size])

    def write(self, s):
        _diff = len(s) - len(self.buf)
        self.buf = s
        return _diff
```

Third Party NVRAM

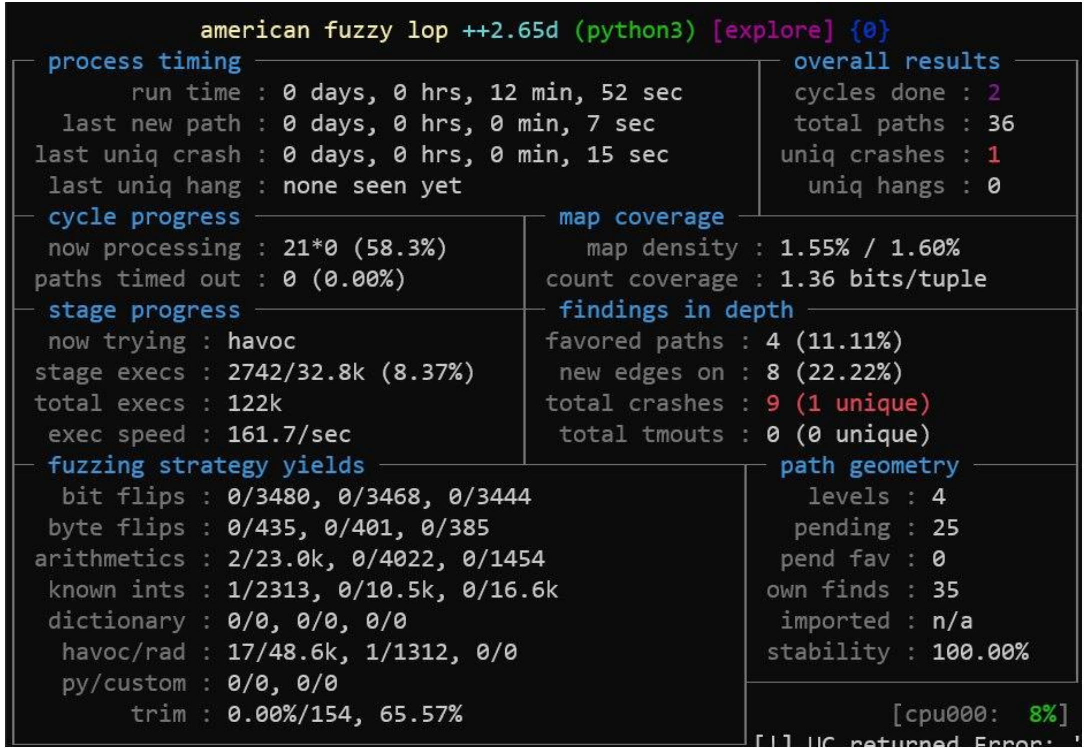
- Emulate Unix Domain Socket Connections
- Emulate ENV Input



```
env_vars = {
    "REQUEST_METHOD": "POST",
    "REQUEST_URI": "/hedwig.cgi",
    "CONTENT_TYPE": "application/x-www-form-urlencoded",
    "REMOTE_ADDR": "127.0.0.1",
    "HTTP_COOKIE": "uid=1234&password="+"A" * 0x1000,  # fill up
    # "CONTENT_LENGTH": "8", # no needed
}

ql = Qiling(["../rootfs/dir815_linux/htdocs/web/hedwig.cgi"], "../rootfs/dir815_linux",
```

Firmware Fuzzing

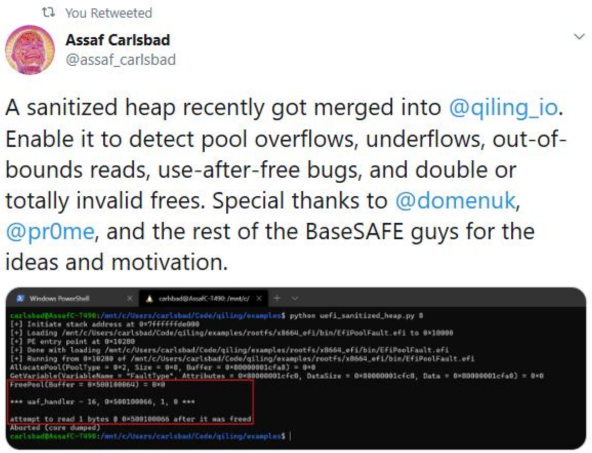
- Fuzzing DIR-815
- https://www.exploit-db.com/exploits/33863
- https://drive.google.com/file/d/10f3cqObsyZ_GHFy0DM-9d1VdsKCVhYjS/view?usp=sharing



```
               american fuzzy lop ++2.65d (python3) [explore] {0}
 process timing                                        overall results
        run time : 0 days, 0 hrs, 12 min, 52 sec       cycles done : 2
   last new path : 0 days, 0 hrs, 0 min, 7 sec         total paths : 36
 last uniq crash : 0 days, 0 hrs, 0 min, 15 sec       uniq crashes : 1
  last uniq hang : none seen yet                        uniq hangs : 0
 cycle progress                       map coverage
  now processing : 21*0 (58.3%)          map density : 1.55% / 1.60%
 paths timed out : 0 (0.00%)          count coverage : 1.36 bits/tuple
 stage progress                       findings in depth
  now trying : havoc                  favored paths : 4 (11.11%)
 stage execs : 2742/32.8k (8.37%)      new edges on : 8 (22.22%)
 total execs : 122k                   total crashes : 9 (1 unique)
  exec speed : 161.7/sec               total tmouts : 0 (0 unique)
 fuzzing strategy yields                                path geometry
   bit flips : 0/3480, 0/3468, 0/3444                     levels : 4
  byte flips : 0/435, 0/401, 0/385                       pending : 25
 arithmetics : 2/23.0k, 0/4022, 0/1454                  pend fav : 0
  known ints : 1/2313, 0/10.5k, 0/16.6k                own finds : 35
  dictionary : 0/0, 0/0, 0/0                            imported : n/a
   havoc/rad : 17/48.6k, 1/1312, 0/0                   stability : 100.00%
   py/custom : 0/0, 0/0
        trim : 0.00%/154, 65.57%                              [cpu000:  8%]
                                                    [!] UC returned Error: '
```

What Else

More Features

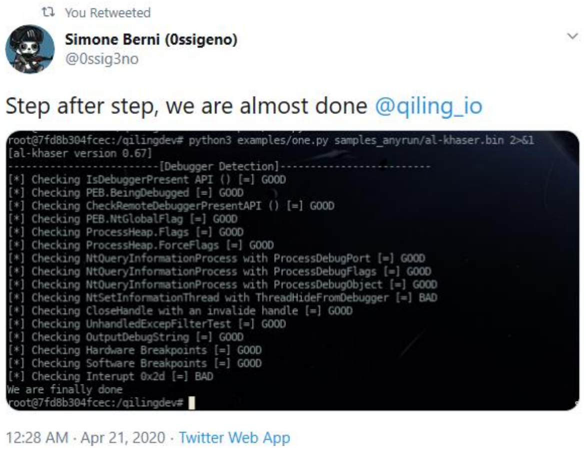


You Retweeted

Assaf Carlsbad
@assaf_carlsbad

A sanitized heap recently got merged into @qiling_io. Enable it to detect pool overflows, underflows, out-of-bounds reads, use-after-free bugs, and double or totally invalid frees. Special thanks to @domenuk, @pr0me, and the rest of the BaseSAFE guys for the ideas and motivation.

```
carlsbad@AssafC-T490:/mnt/c/Users/carlsbad/Code/qiling/examples$ python uefi_sanitized_heap.py 8
[+] Initiate stack address at 0x7ffffffde000
[+] Loading /mnt/c/Users/carlsbad/Code/qiling/examples/rootfs/x8664_efi/bin/EfiPoolFault.efi to 0x10000
[+] PE entry point at 0x10280
[+] Done with loading /mnt/c/Users/carlsbad/Code/qiling/examples/rootfs/x8664_efi/bin/EfiPoolFault.efi
[+] Running from 0x10280 of /mnt/c/Users/carlsbad/Code/qiling/examples/rootfs/x8664_efi/bin/EfiPoolFault.efi
AllocatePool(PoolType = 0x2, Size = 0x8, Buffer = 0x80000001cfa8) = 0x0
GetVariable(VariableName = "FaultType", Attributes = 0x80000001cfc0, DataSize = 0x80000001cfc8, Data = 0x80000001cfa0) = 0x0
FreePool(Buffer = 0x500100064) = 0x0

*** uaf_handler - 16, 0x500100066, 1, 0 ***

attempt to read 1 bytes @ 0x500100066 after it was freed
Aborted (core dumped)
carlsbad@AssafC-T490:/mnt/c/Users/carlsbad/Code/qiling/examples$ |
```



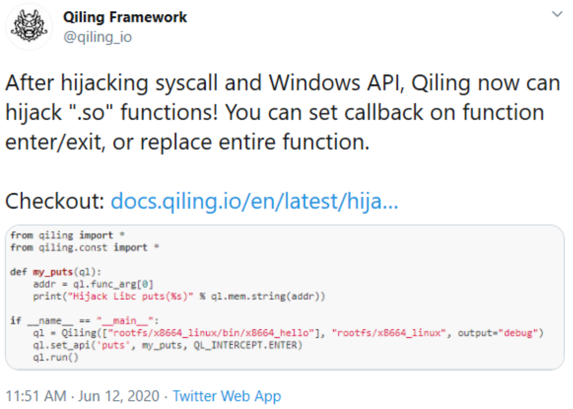
You Retweeted

Simone Berni (0ssigeno)
@0ssig3no

Step after step, we are almost done @qiling_io

```
root@7fd8b304fcec:/qilingdev# python3 examples/one.py samples_anyrun/al-khaser.bin 2>&1
[al-khaser version 0.67]
-------------------------[Debugger Detection]-------------------------
[*] Checking IsDebuggerPresent API () [=] GOOD
[*] Checking PEB.BeingDebugged [=] GOOD
[*] Checking CheckRemoteDebuggerPresentAPI () [=] GOOD
[*] Checking PEB.NtGlobalFlag [=] GOOD
[*] Checking ProcessHeap.Flags [=] GOOD
[*] Checking ProcessHeap.ForceFlags [=] GOOD
[*] Checking NtQueryInformationProcess with ProcessDebugPort [=] GOOD
[*] Checking NtQueryInformationProcess with ProcessDebugFlags [=] GOOD
[*] Checking NtQueryInformationProcess with ProcessDebugObject [=] GOOD
[*] Checking NtSetInformationThread with ThreadHideFromDebugger [=] BAD
[*] Checking CloseHandle with an invalide handle [=] GOOD
[*] Checking UnhandledExcepFilterTest [=] GOOD
[*] Checking OutputDebugString [=] GOOD
[*] Checking Hardware Breakpoints [=] GOOD
[*] Checking Software Breakpoints [=] GOOD
[*] Checking Interupt 0x2d [=] BAD
We are finally done
root@7fd8b304fcec:/qilingdev# █
```

12:28 AM · Apr 21, 2020 · Twitter Web App



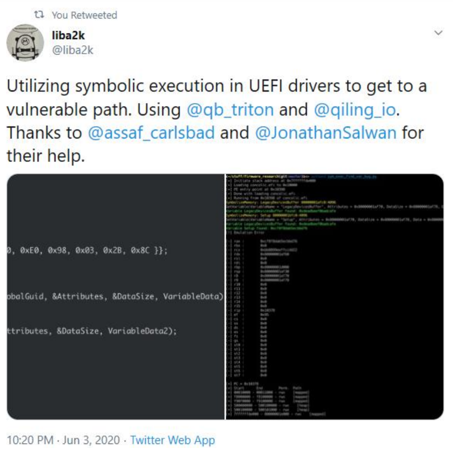
Qiling Framework
@qiling_io

After hijacking syscall and Windows API, Qiling now can hijack ".so" functions! You can set callback on function enter/exit, or replace entire function.

Checkout: docs.qiling.io/en/latest/hija...

```python
from qiling import *
from qiling.const import *

def my_puts(ql):
    addr = ql.func_arg[0]
    print("Hijack Libc puts(%s)" % ql.mem.string(addr))

if __name__ == "__main__":
    ql = Qiling(["rootfs/x8664_linux/bin/x8664_hello"], "rootfs/x8664_linux", output="debug")
    ql.set_api('puts', my_puts, QL_INTERCEPT.ENTER)
    ql.run()
```

11:51 AM · Jun 12, 2020 · Twitter Web App



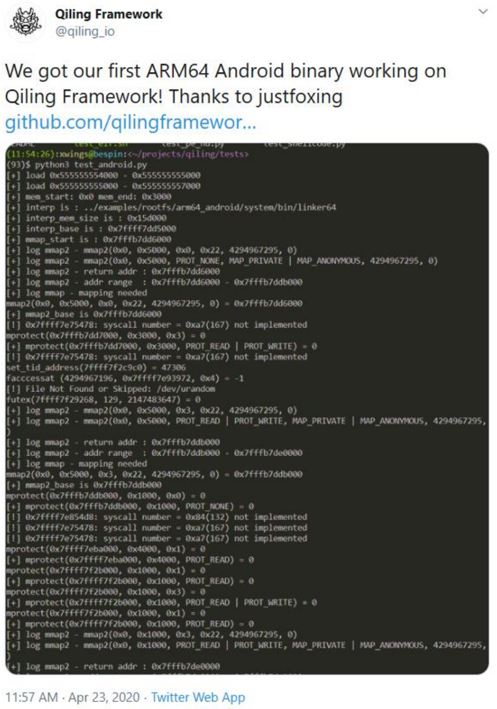
You Retweeted

liba2k
@liba2k

Utilizing symbolic execution in UEFI drivers to get to a vulnerable path. Using @qb_triton and @qiling_io. Thanks to @assaf_carlsbad and @JonathanSalwan for their help.

```
0, 0xE0, 0x98, 0x03, 0x2B, 0x8C }};

obalGuid, &Attributes, &DataSize, VariableData)

ttributes, &DataSize, VariableData2);
```

10:20 PM · Jun 3, 2020 · Twitter Web App

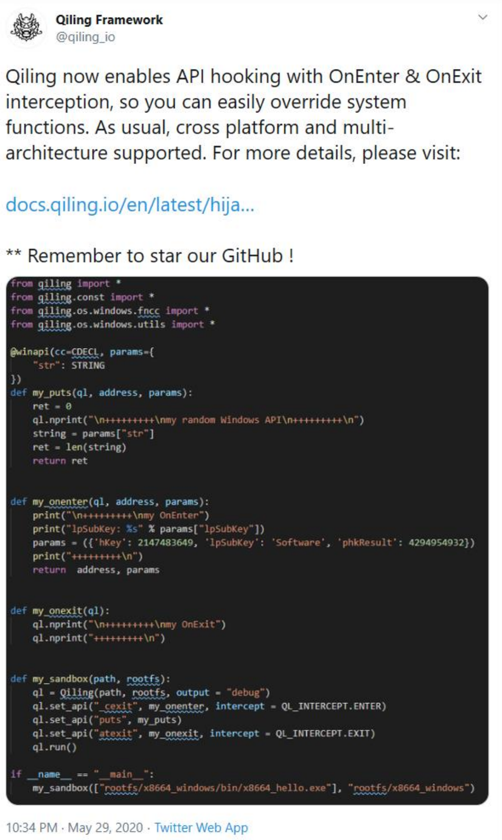


Qiling Framework
@qiling_io

We got our first ARM64 Android binary working on Qiling Framework! Thanks to justfoxing github.com/qilingframewor...

```
(11:54:26):xwings@bespin:<~/projects/qiling/tests>
(93)$ python3 test_android.py
[+] load 0x555555554000 - 0x555555555000
[+] load 0x555555555000 - 0x555555557000
[+] mem_start: 0x0 mem_end: 0x3000
[+] interp is : ../examples/rootfs/arm64_android/system/bin/linker64
[+] interp_mem_size is : 0x15d000
[+] interp_base is : 0x7ffff7dd5000
[+] mmap_start is : 0x7fffb7dd6000
[+] log mmap2 - mmap2(0x0, 0x5000, 0x0, 0x22, 4294967295, 0)
[+] log mmap2 - mmap2(0x0, 0x5000, PROT_NONE, MAP_PRIVATE | MAP_ANONYMOUS, 4294967295, 0)
[+] log mmap2 - return addr : 0x7fffb7dd6000
[+] log mmap2 - addr range  : 0x7fffb7dd6000 - 0x7fffb7ddb000
[+] log mmap - mapping needed
mmap2(0x0, 0x5000, 0x0, 0x22, 4294967295, 0) = 0x7fffb7dd6000
[+] mmap2_base is 0x7fffb7dd6000
[!] 0x7ffff7e75478: syscall number = 0xa7(167) not implemented
mprotect(0x7fffb7dd7000, 0x3000, 0x3) = 0
[+] mprotect(0x7fffb7dd7000, 0x3000, PROT_READ | PROT_WRITE) = 0
[!] 0x7ffff7e75478: syscall number = 0xa7(167) not implemented
set_tid_address(7ffff7f2c9c0) = 47306
facccessat (4294967196, 0x7ffff7e93972, 0x4) = -1
[!] File Not Found or Skipped: /dev/urandom
futex(7ffff7f29268, 129, 2147483647) = 0
[+] log mmap2 - mmap2(0x0, 0x5000, 0x3, 0x22, 4294967295, 0)
[+] log mmap2 - mmap2(0x0, 0x5000, PROT_READ | PROT_WRITE, MAP_PRIVATE | MAP_ANONYMOUS, 4294967295,
)
[+] log mmap2 - return addr : 0x7fffb7ddb000
[+] log mmap2 - addr range  : 0x7fffb7ddb000 - 0x7fffb7de0000
[+] log mmap - mapping needed
mmap2(0x0, 0x5000, 0x3, 0x22, 4294967295, 0) = 0x7fffb7ddb000
[+] mmap2_base is 0x7fffb7ddb000
mprotect(0x7fffb7ddb000, 0x1000, 0x0) = 0
[+] mprotect(0x7fffb7ddb000, 0x1000, PROT_NONE) = 0
[!] 0x7ffff7e854d8: syscall number = 0x84(132) not implemented
[!] 0x7ffff7e75478: syscall number = 0xa7(167) not implemented
[!] 0x7ffff7e75478: syscall number = 0xa7(167) not implemented
mprotect(0x7ffff7eba000, 0x4000, 0x1) = 0
[+] mprotect(0x7ffff7eba000, 0x4000, PROT_READ) = 0
mprotect(0x7ffff7f2b000, 0x1000, 0x1) = 0
[+] mprotect(0x7ffff7f2b000, 0x1000, PROT_READ) = 0
mprotect(0x7ffff7f2b000, 0x1000, 0x3) = 0
[+] mprotect(0x7ffff7f2b000, 0x1000, PROT_READ | PROT_WRITE) = 0
mprotect(0x7ffff7f2b000, 0x1000, 0x1) = 0
[+] mprotect(0x7ffff7f2b000, 0x1000, PROT_READ) = 0
[+] log mmap2 - mmap2(0x0, 0x1000, 0x3, 0x22, 4294967295, 0)
[+] log mmap2 - mmap2(0x0, 0x1000, PROT_READ | PROT_WRITE, MAP_PRIVATE | MAP_ANONYMOUS, 4294967295,
)
[+] log mmap2 - return addr : 0x7fffb7de0000
```

11:57 AM · Apr 23, 2020 · Twitter Web App



Qiling Framework
@qiling_io

Qiling now enables API hooking with OnEnter & OnExit interception, so you can easily override system functions. As usual, cross platform and multi-architecture supported. For more details, please visit:

docs.qiling.io/en/latest/hija...

** Remember to star our GitHub !

```python
from qiling import *
from qiling.const import *
from qiling.os.windows.fncc import *
from qiling.os.windows.utils import *

@winapi(cc=CDECL, params={
    "str": STRING
})
def my_puts(ql, address, params):
    ret = 0
    ql.nprint("\n+++++++++\nmy random Windows API\n+++++++++\n")
    string = params["str"]
    ret = len(string)
    return ret


def my_onenter(ql, address, params):
    print("\n+++++++++\nmy OnEnter")
    print("lpSubKey: %s" % params["lpSubKey"])
    params = ({'hKey': 2147483649, 'lpSubKey': 'Software', 'phkResult': 4294954932})
    print("+++++++++\n")
    return  address, params


def my_onexit(ql):
    ql.nprint("\n+++++++++\nmy OnExit")
    ql.nprint("+++++++++\n")


def my_sandbox(path, rootfs):
    ql = Qiling(path, rootfs, output = "debug")
    ql.set_api("_cexit", my_onenter, intercept = QL_INTERCEPT.ENTER)
    ql.set_api("puts", my_puts)
    ql.set_api("atexit", my_onexit, intercept = QL_INTERCEPT.EXIT)
    ql.run()

if __name__ == "__main__":
    my_sandbox(["rootfs/x8664_windows/bin/x8664_hello.exe"], "rootfs/x8664_windows")
```

10:34 PM · May 29, 2020 · Twitter Web App

https://docs.qiling.io

One Last Thing

Call for sponsor for development of Unicorn 2

- Current Unicorn is based on Qemu 2.1.2, from 2015
- Planning for **Unicorn 2**, based on new Qemu (5+)
- Some new exciting APIs in planning
- https://github.com/unicorn-engine/unicorn/issues/1217

NGUYEN Anh Quynh, aquynh -at- gmail.com, @unicorn_engine