Binary File and Image Notes

1 Introduction

Goal is to understand how image file(s) are generated and how they can be deconstructed so that a web file set and other information can be recovered.

2 Misc.

- Use the file command to obtain info about a file.
- Compressed Image header: 0x

2.1 File Signatures

GZIP	1F	8B	08*	
TAR	1F	9D	90	
ZIP	50	4B	03	04
CRAMFS	45	3D	CD	28

Table 1File Type Signatures

2.1.1 GZIP

First two bytes of the signature are the GZIP ID, the third byte is the compression method used, which is often '08' but is not guaranteed. If the 0x1F8B08 signature can't be found look for 0x1F8B.

2.1.2 TAR

2.1.3 ZIP

2.1.4 CRAMFS

3 DAPDK Images

The system image is generated in the uClinux directory and contains the kernel, applications, libraries, and some web interface functionality.

3.1 uClinux

Order in which the images are created is somewhat confusing. It looks like ".romfs.img" file is created first in the uClinux directory and is then copied to the kernel directory as "romfs.img". Then we go to the kernel directory and create zImage. After zImage is created (zImage.bin) then it is copied to the uClinux directory as the final image, uclinux_system.img.

Just guessing but it looks like we take piggy, compress piggy, then use piggy.bz2 to make piggy.o (?). piggy.o, vmlinux.lds, head-isil.o, misc-bzip2.o, and bootrec.o are used to create vmlinux. Vmlinux is then turned into zImage

3.2 apfw

3.3 webfileset

3.4 apfw_package

4 NSLU2

The Linksys NSLU2 Network Attached Storage (NAS) device is a Linux based platform popular with hackers. It is possible to reverse engineer part of the image Linksys distributes for this device, as outlined below.

5 File System Types

There are a variety of small and simple filesystems used in embedded devices such as a cramfs and romfs. Typically these images based on these file systems can be mounted by via loopback.

Here is the typical command for mounting a cramfs, romfs, or ramfs files system:

```
mount -o loop image_name mnt_point
```

the loopback (loop) interface allows one to use a file as if it were a device.

5.1 cramfs

A cramfs is a small, compressed Linux file system. Used for embedded systems and small devices.

5.2 romfs

A romfs is a small, efficient, *read-only* Linux file system. Stores only the minimum file system, excluding common information such as modification times and permissions.

romfs is used with uClinux, SnapGear, Kiwi, and other embedded projects.

5.3 ramfs

A ramfs keeps all files in RAM allowing read and write access to files.

5.4 jffs