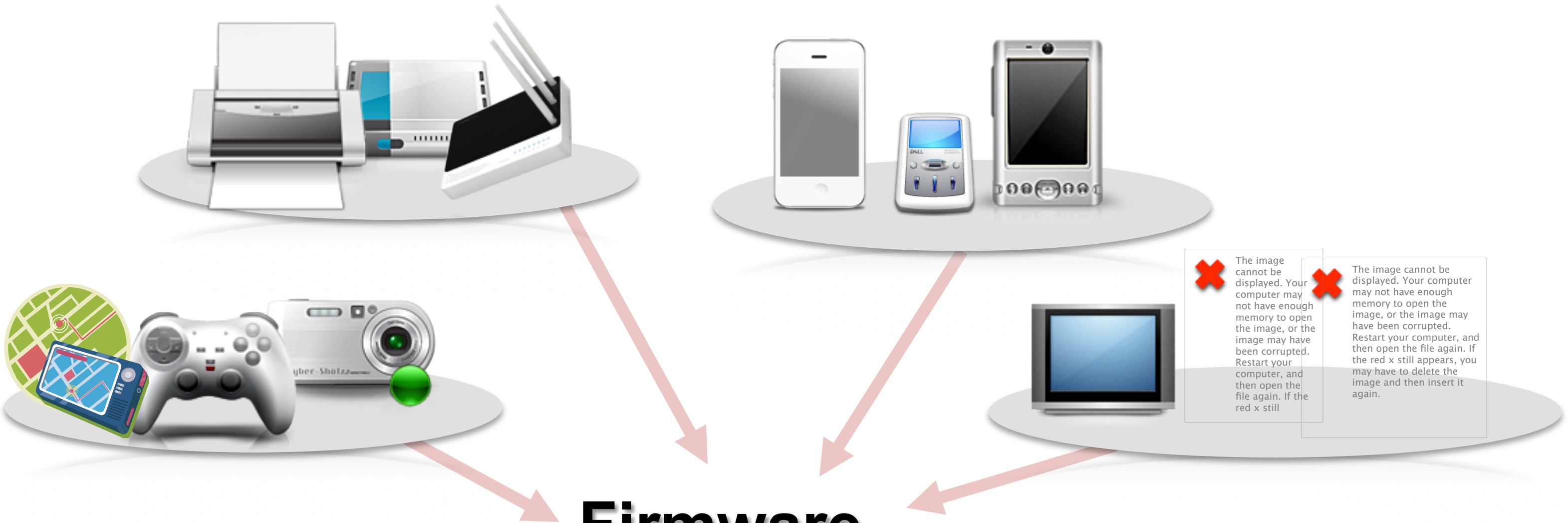
Fu~n of Attacking Firmware

2012 POC

Silverbug

RedHidden

WHAT ..


The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still
The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

Firmware

a small program to control devices

non-volatility memory device

Firmware update : fixing bugs or adding features

…

WHY ..

Provide new features
Unlock hidden functionality
Find vulnerabilities
Use for the malicious purpose

2011.01 SONY “PS Jailbreak”

Geohot – release free jailbreak for v3.55 firmware
Bypass SONY's security check with USB dongle
Execute unsigned code

2011.09 CANON DSLR firmware hacking

Magic Latern – release a custom firmware add-on (modified firmware)
Photo and Video enthusiasts

WHY ..

2009. 10 Samung TV firmware hacking

SamyGo-reverse engineering project for Samsung TV Firmware
Unlocked the ability to use non-Samsung WIFI dongles
Improved playback from USB devices
Implemented NFS and SAMBA for sharing file over the network

2011.11 HP LaserJet Printer Vulnerability

Researchers From Columbia University
Not check digital signatures before installing a firmware update
Accept arbitrarily modified firmware
Erase its existing os and overwrite with a malicious one

WHY ..

2012 SECUINSIDE Wireless Router Hacking

IPTIME G104 - CGI Buffer Overflow Vulnerability
ANYGATE – Execute Command with Non-Authentication

2012 VB2012 ADSL Modem Hacking

Fabio Assolini, Kaspersky
ADSL Router CT-5367 – CSRF, UPNP/SNMP misconfiguration

2012 DEFCON Rooting SOHO Router

Zachary Cutlip, Tactical Network Solutions
Netgear WNDR3700v3 – SQL Injection to MIPS Overflows

Adventures in Router Rootkits

Michael Coppola, VSR
Netgear, Belkin, TRENDnet – Owning the Network

WHY ..

Wireless Router

OpenWrt
Wireless Freedom

dd-wrt.com

RouterTech

/DEV/TTYS0 Embedded Device Hacking

Firmware Tools

rpef
Router Post-Exploitation Framework

UWfirmforce
Automated firmware reverse-engineering tool

binwalk
Firmware Analysis Tool

firmware-mod-kit
This kit allows for easy deconstruction and reconsutrction of firmware images for various embedded devices

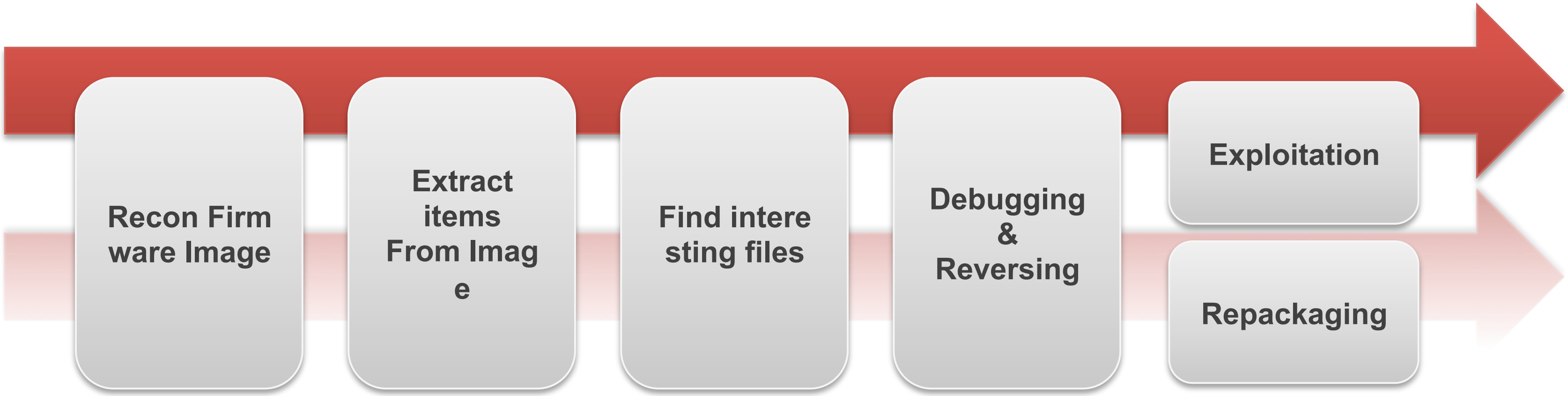
Collection Info.

Router exploitation framework

littleblackbox
Database of private SSL/SSH keys for embedded devices

HOW ..

Firmware Hacking Process



Recon Firm
ware Image

Extract
items
From Imag
e

Find intere
sting files

Debugging
&
Reversing

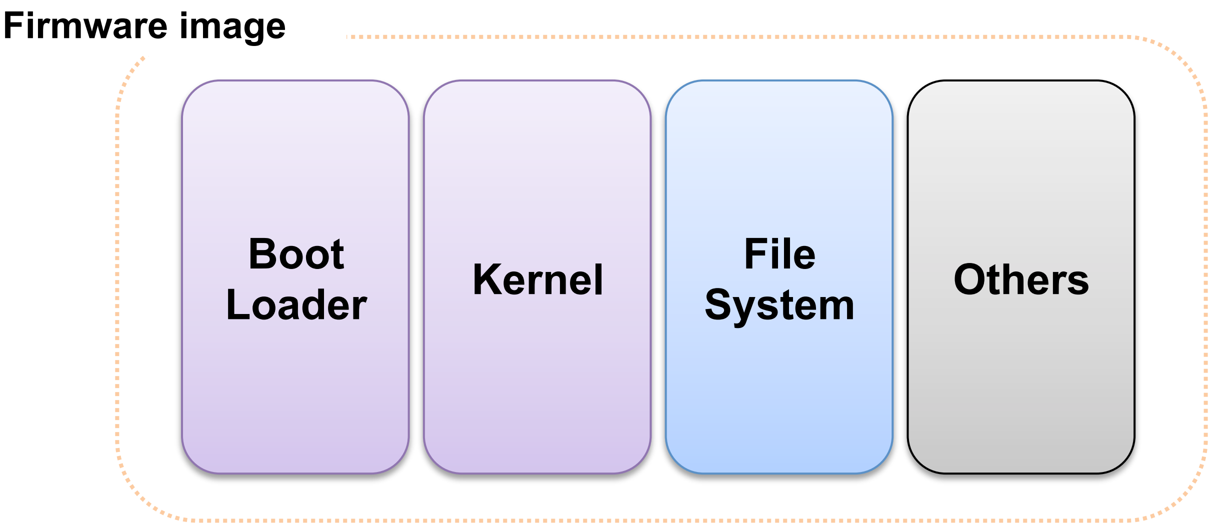
Exploitation

Repackaging

```
myfirm#  binwalk, signsrch, offzip, trid
myfirm#  file, strings, hexdump, objdump
myfirm#  dd
myfirm#  firmware_mod_kit …..
myfirm#  IDA, qemu, gdb …..
myfirm#  extract tools, deflate tools…..
```

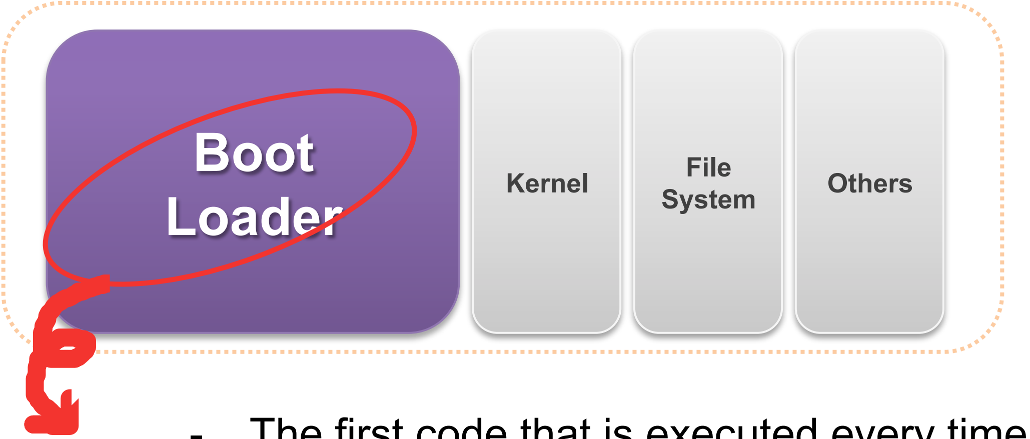
HOW ..

Firmware(Image) Structure



HOW ..

Firmware(Image) - BootLoader

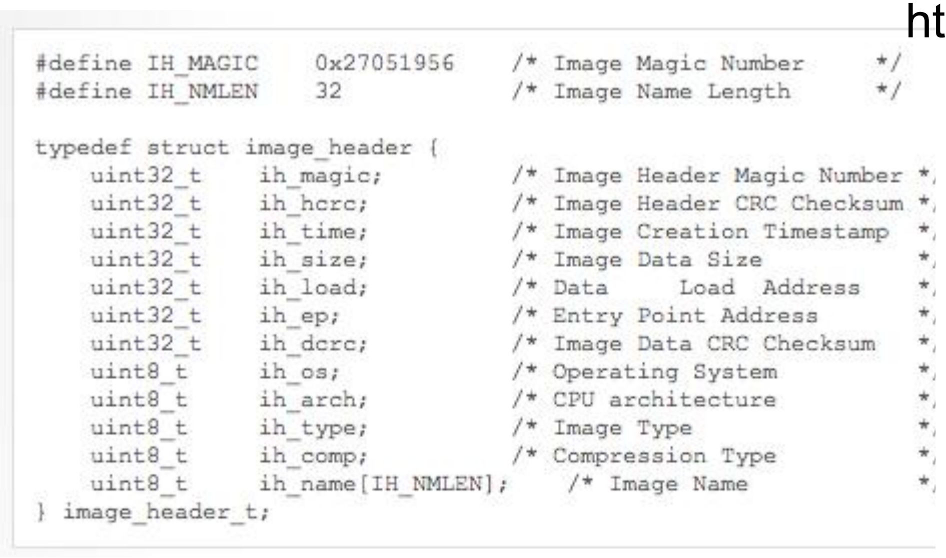


- The first code that is executed every time a system reset
- Initialize hardware and load the correct image from flash
- Execute the Kernel
- Placed in a part of flash or a separate EEPROM
- For Embedded Devices,
  - *Das U-Boot, RedBoot*
  - *CFE, Adam2, PSPBoot*
  - *NetBoot(DWL7100AP)*
  - *VxWorks' own bootloader(Netgear WGT624)*
  - *ThreadX(D-Link)*

HOW ..

BootLoader (ex) U-Boot

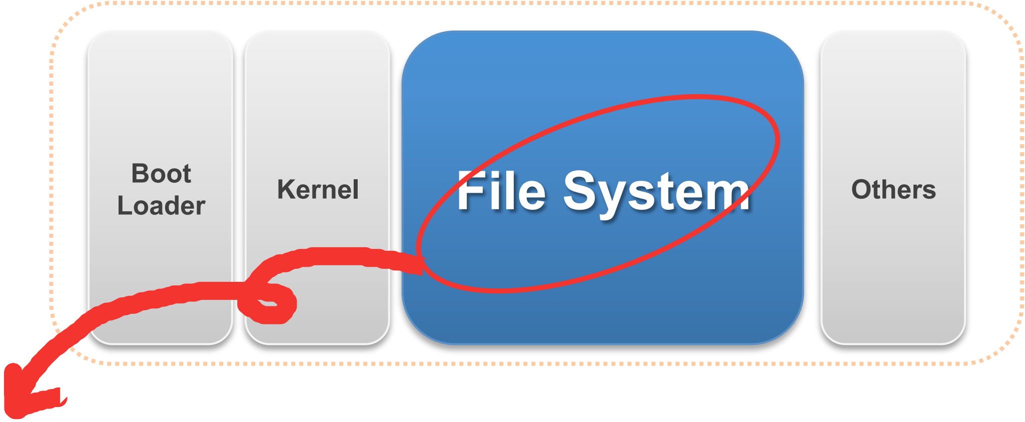
"Das U-Boot" - Universal Bootloader
PowerPC, ARM, MIPS systems
mostly used to load and boot a kernel image
U-Boot Image = <U-Boot Header> <Kernel Image>
http://www.denx.de/wiki/U-Boot
http://sourceforge.net/project/u-boot
Image Create : mkimage



```
#define IH_MAGIC    0x27051956    /* Image Magic Number     */
#define IH_NMLEN    32            /* Image Name Length      */

typedef struct image_header {
    uint32_t    ih_magic;         /* Image Header Magic Number */
    uint32_t    ih_hcrc;          /* Image Header CRC Checksum */
    uint32_t    ih_time;          /* Image Creation Timestamp  */
    uint32_t    ih_size;          /* Image Data Size           */
    uint32_t    ih_load;          /* Data     Load  Address    */
    uint32_t    ih_ep;            /* Entry Point Address       */
    uint32_t    ih_dcrc;          /* Image Data CRC Checksum   */
    uint8_t     ih_os;            /* Operating System          */
    uint8_t     ih_arch;          /* CPU architecture          */
    uint8_t     ih_type;          /* Image Type                */
    uint8_t     ih_comp;          /* Compression Type          */
    uint8_t     ih_name[IH_NMLEN];    /* Image Name            */
} image_header_t;
```

HOW ..

Firmware(Image) - Filesystem



Use flash memory as storage media
Size and bootup time are very important
Used with the enhanced compression, or the ability to execute file directly from flash
For Embedded System,

- *SquashFS, JFFS2*
- *cramFS, ext2*
- *YAFFS2, tmpfFS*
- *PFS*

HOW ..

FileSystem - SquashFS

Linux, read only compressed file system.
Use zlib, lzo, xz (LZMA) compression for files, inodes, directories
max filesystem size : 2^64

packing/unpacking tool :
- squashfs-tools (mksquashfs, unsquashfs)
- Re7zip
- E-Pack Decompressor
- https://github.com/vasi/squash.rb/blob/master/squash.rb
- https://github.com/matteomattei/PySquashfsImage

HOW ..

FileSystem - cramFS

Linux, cram a file system onto a small ROM
Read-only file system
Designed to be simple and small, and to compress things well
Data stored in compress format – Zlib
Meta data is not compressed
Max file system size : 2^16(256MB)
cramFS = <superbloc><directory_structure><data>

Packing/unpacking tool :

- cramfs tools : mkcramfs
- E-Pack Decompressor
- Fsck.cramfs, mkfs.cramfs

HOW ..

File System – JFFS2

Linux, the journaling Flash file system v2, a log-based file system
Read/Write File system
Add compression to JFFS
Compress algorithm : zlib, runbin, rtime
Designed for use on NOR and NAND flash devices

Packing/unpacking tool :

- mkfs.jffs2
- E-Pack Decompressor
- mtd-mods(projects)

HOW ..

Archive Format

Gzip(Zlib)

GNU Zip, primary compression format used by Unix-based system
Compression Algorithm : DEFLATE
Format = <Gzip header ><Deflate compressed Blocks><GZIP Footer>
Header : 10byte – magic number, version, timestamp
Footer : 8byte – CRC Checksum, uncompressed data length
Magic Signature : \x1F\x8B
uncompress : gzip –d <.gz file>

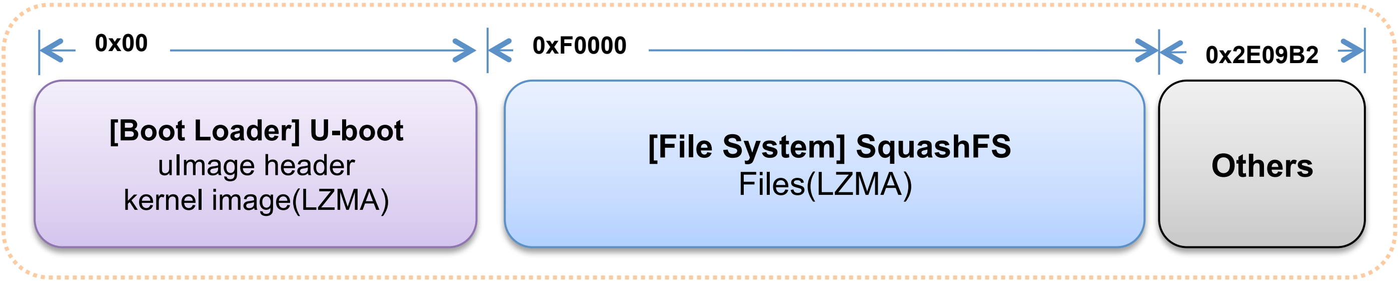
LZMA

Lempel-Ziv-Markov chain algorithm
Compression Algorithm : dictionary compression scheme(LZ77 variant)
Magic Signature : \x5D\x00\x00\x80
Uncompress : lzma –d <.lzma file>

HOW ..

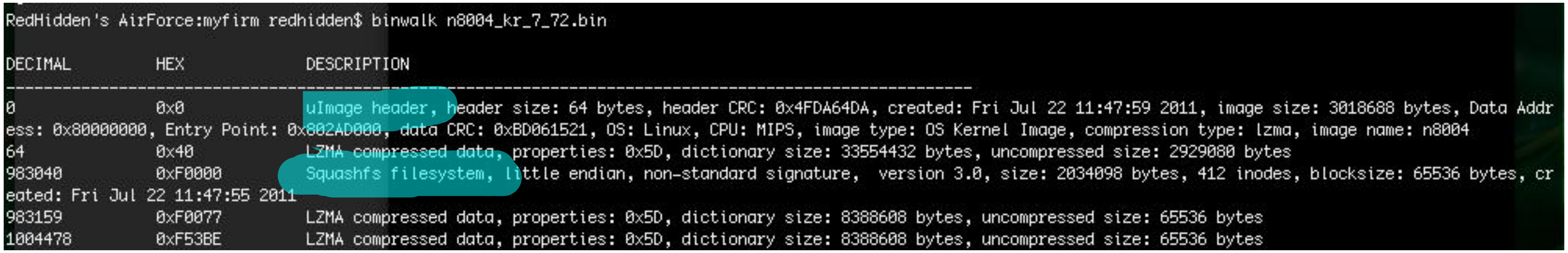
Firmware Hacking (Demo)

# Target : IPTIME N8004
# Firmware Version : 7.72 (n8004_kr_7.72.bin)
# Vulnerability : Get Administrator Password (p
artial apply the patch)

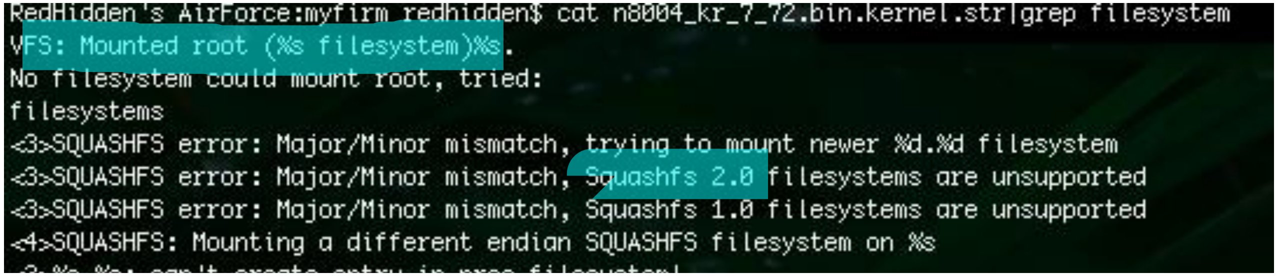


HOW ..

Recon the firmware image



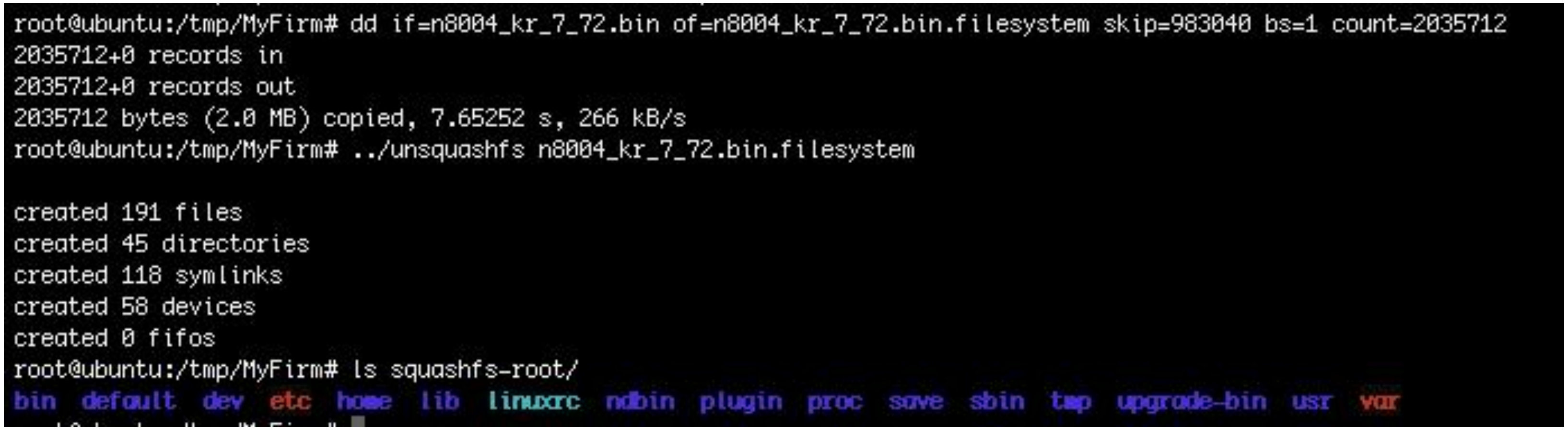
```
RedHidden's AirForce:myfirm redhidden$ binwalk n8004_kr_7_72.bin

DECIMAL         HEX             DESCRIPTION
-------------------------------------------------------------------------------------------------------
0               0x0             uImage header, header size: 64 bytes, header CRC: 0x4FDA64DA, created: Fri Jul 22 11:47:59 2011, image size: 3018688 bytes, Data Addr
ess: 0x80000000, Entry Point: 0x802AD000, data CRC: 0xBD061521, OS: Linux, CPU: MIPS, image type: OS Kernel Image, compression type: lzma, image name: n8004
64              0x40            LZMA compressed data, properties: 0x5D, dictionary size: 33554432 bytes, uncompressed size: 2929080 bytes
983040          0xF0000         Squashfs filesystem, little endian, non-standard signature,  version 3.0, size: 2034098 bytes, 412 inodes, blocksize: 65536 bytes, cr
eated: Fri Jul 22 11:47:55 2011
983159          0xF0077         LZMA compressed data, properties: 0x5D, dictionary size: 8388608 bytes, uncompressed size: 65536 bytes
1004478         0xF53BE         LZMA compressed data, properties: 0x5D, dictionary size: 8388608 bytes, uncompressed size: 65536 bytes
```



```
RedHidden's AirForce:myfirm redhidden$ cat n8004_kr_7_72.bin.kernel.str|grep filesystem
VFS: Mounted root (%s filesystem)%s.
No filesystem could mount root, tried:
filesystems
<3>SQUASHFS error: Major/Minor mismatch, trying to mount newer %d.%d filesystem
<3>SQUASHFS error: Major/Minor mismatch, Squashfs 2.0 filesystems are unsupported
<3>SQUASHFS error: Major/Minor mismatch, Squashfs 1.0 filesystems are unsupported
<4>SQUASHFS: Mounting a different endian SQUASHFS filesystem on %s
<3>%s %s: can't create entry in proc filesystem!
```

HOW ..

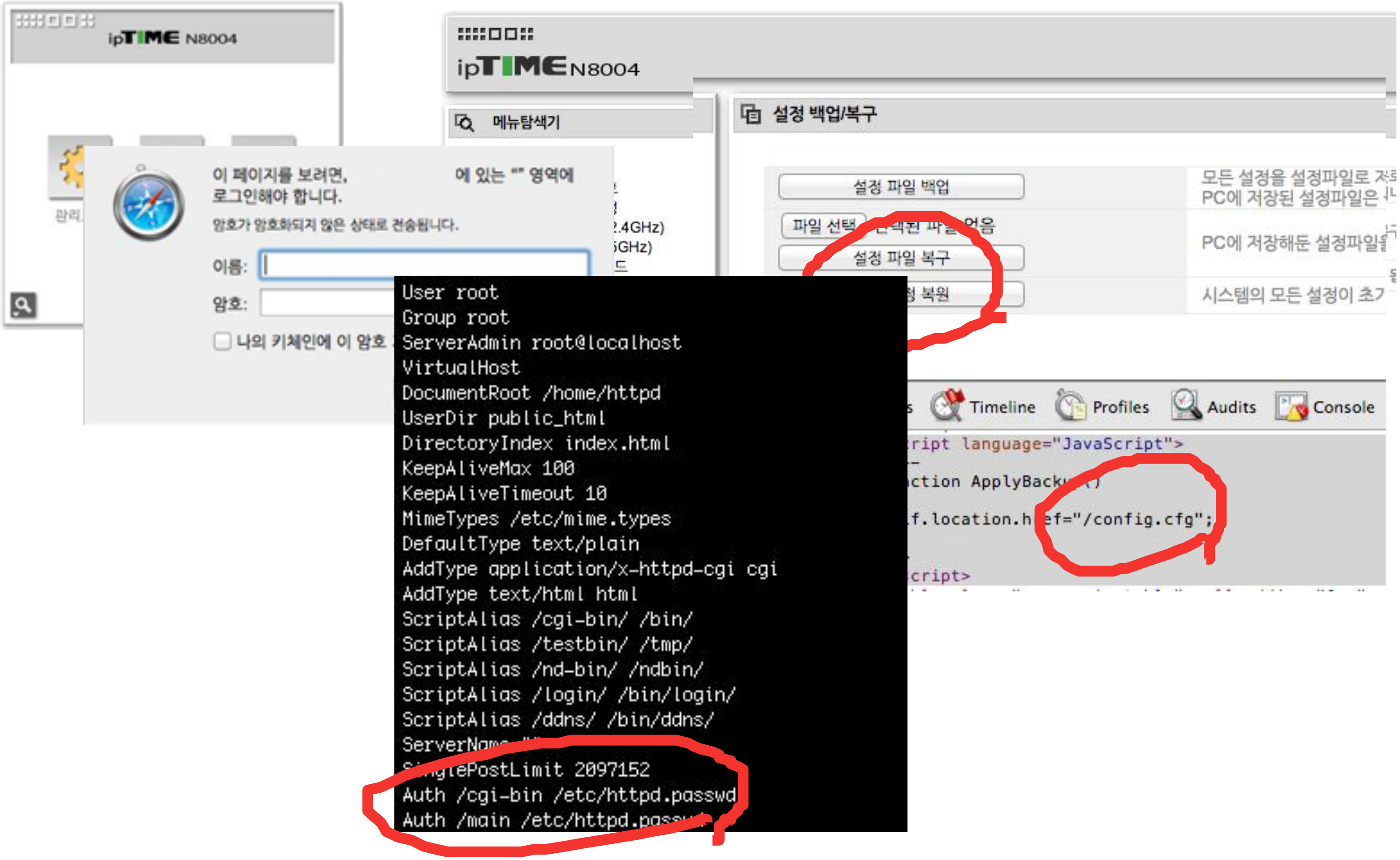
Split the firmware image apart and then unpack



```
root@ubuntu:/tmp/MyFirm# dd if=n8004_kr_7_72.bin of=n8004_kr_7_72.bin.filesystem skip=983040 bs=1 count=2035712
2035712+0 records in
2035712+0 records out
2035712 bytes (2.0 MB) copied, 7.65252 s, 266 kB/s
root@ubuntu:/tmp/MyFirm# ../unsquashfs n8004_kr_7_72.bin.filesystem

created 191 files
created 45 directories
created 118 symlinks
created 58 devices
created 0 fifos
root@ubuntu:/tmp/MyFirm# ls squashfs-root/
bin  default  dev  etc  home  lib  linuxrc  ndbin  plugin  proc  save  sbin  tmp  upgrade-bin  usr  var
```

HOW ..

Find a bugs and vulnerability



이 페이지를 보려면, 에 있는 "" 영역에
로그인해야 합니다.

암호가 암호화되지 않은 상태로 전송됩니다.

이름:

암호:

□ 나의 키체인에 이 암호

설정 백업/복구

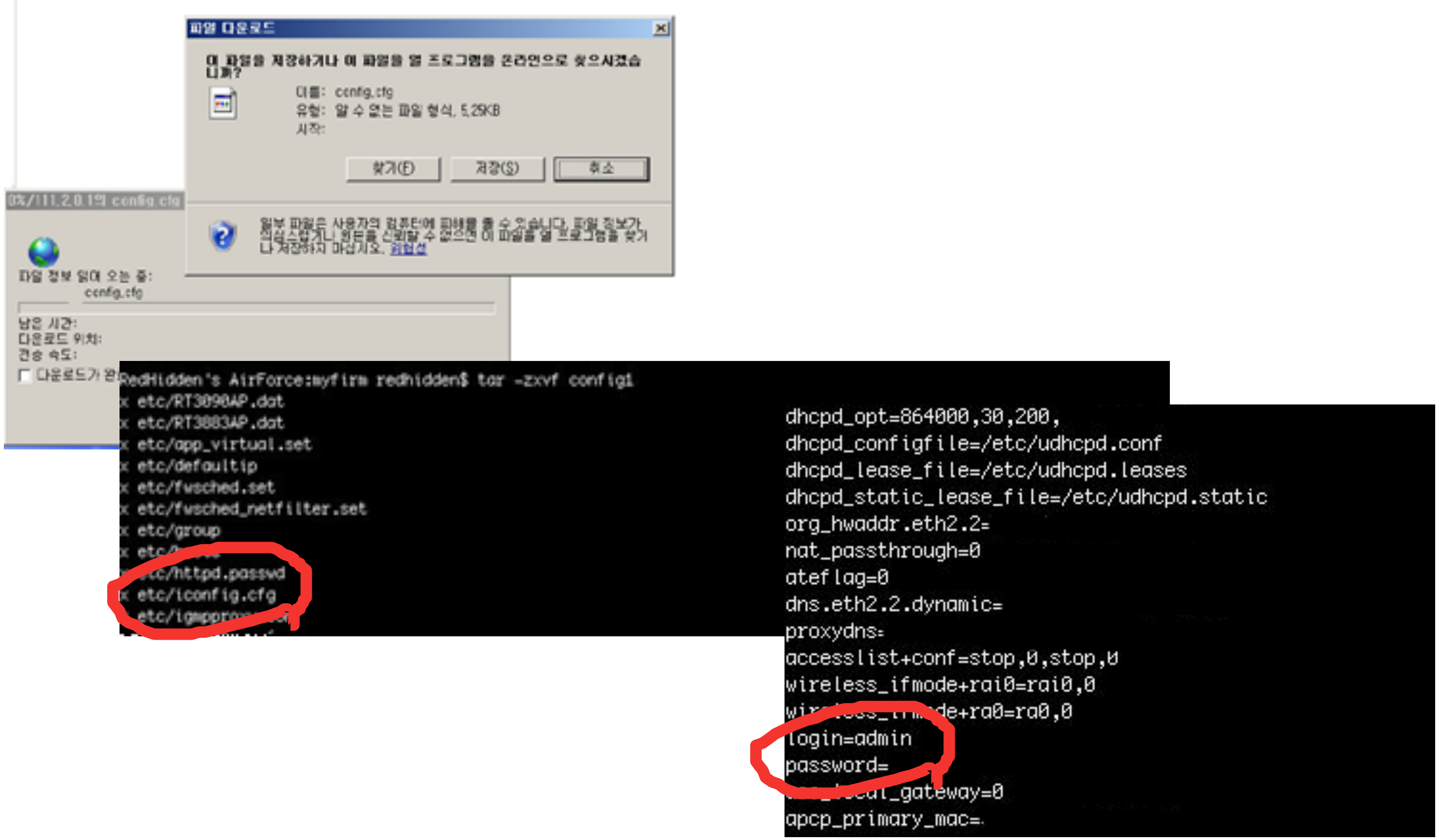
설정 파일 백업	모든 설정을 설정파일로 저 PC에 저장된 설정파일은
파일 선택 선택된 파일 없음 설정 파일 복구	PC에 저장해둔 설정파일을
설정 복원	시스템의 모든 설정이 초기

```
ript language="JavaScript">
-
ction ApplyBackup()

f.location.href="/config.cfg";

cript>
```

```
User root
Group root
ServerAdmin root@localhost
VirtualHost
DocumentRoot /home/httpd
UserDir public_html
DirectoryIndex index.html
KeepAliveMax 100
KeepAliveTimeout 10
MimeTypes /etc/mime.types
DefaultType text/plain
AddType application/x-httpd-cgi cgi
AddType text/html html
ScriptAlias /cgi-bin/ /bin/
ScriptAlias /testbin/ /tmp/
ScriptAlias /nd-bin/ /ndbin/
ScriptAlias /login/ /bin/login/
ScriptAlias /ddns/ /bin/ddns/
ServerName ""
SinglePostLimit 2097152
Auth /cgi-bin /etc/httpd.passwd
Auth /main /etc/httpd.passwd
```

HOW ..

Find a bugs and vulnerability



파일 다운로드

이 파일을 저장하거나 이 파일을 열 프로그램을 온라인으로 찾으시겠습니까?

이름: config.cfg
유형: 알 수 없는 파일 형식, 5.25KB
시작:

찾기(F) 저장(S) 취소

일부 파일은 사용자의 컴퓨터에 피해를 줄 수 있습니다. 파일 정보가 의심스럽거나 원본을 신뢰할 수 없으면 이 파일을 열 프로그램을 찾거나 저장하지 마십시오. 위험성

0%/111.2.0.1의 config.cfg

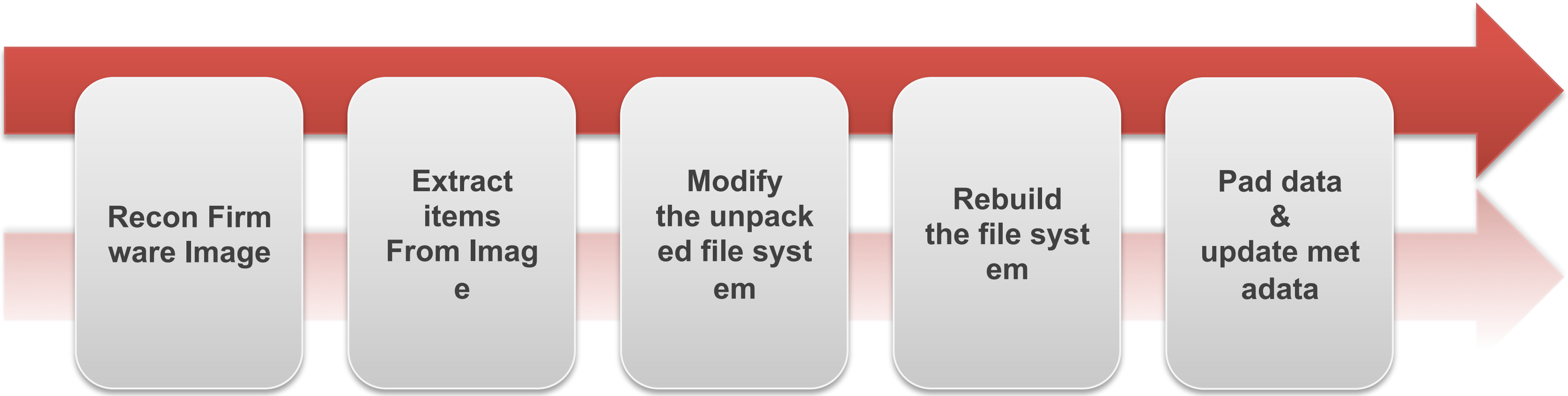
파일 정보 읽어 오는 중:
config.cfg
남은 시간:
다운로드 위치:
전송 속도:
□ 다운로드가 완

```
RedHidden's AirForce:myfirm redhidden$ tar -zxvf config1
x etc/RT3090AP.dat
x etc/RT3883AP.dat
x etc/app_virtual.set
x etc/defaultip
x etc/fwsched.set
x etc/fwsched_netfilter.set
x etc/group
x etc/
x etc/httpd.passwd
x etc/iconfig.cfg
x etc/igmpprox
```

```
dhcpd_opt=864000,30,200,
dhcpd_configfile=/etc/udhcpd.conf
dhcpd_lease_file=/etc/udhcpd.leases
dhcpd_static_lease_file=/etc/udhcpd.static
org_hwaddr.eth2.2=
nat_passthrough=0
ateflag=0
dns.eth2.2.dynamic=
proxydns=
accesslist+conf=stop,0,stop,0
wireless_ifmode+rai0=rai0,0
wir        de+ra0=ra0,0
login=admin
password=
     ocal_gateway=0
apcp_primary_mac=
```

HOW ..

Firmware Repackaging Process



Recon Firm ware Image

Extract items From Imag e

Modify the unpack ed file syst em

Rebuild the file syst em

Pad data & update met adata

Conclusion

As the use of smart and portable devices increase, it’s very easy for us to meet various firmware.

Devices are smart, but not secure.

By firmware hacking research, you can do the following things :

Even though there is no known information,
you can get the"DIY devices" that correspond to the purpose what you want.
You can find the potential security threat in the firmware.

let's start challenging from the firmware located around you.