

Forum:[Hack Forums](#)

Thread: [\[Tutorial\] Reversing firmware with Binwalk and FMK \(Linux\)](#)

KingMidas - October 19, 2014, 5:16 pm

Hey everybody!

Welcome to my very first tutorial on Reversing firmware with Binwalk and Firmware Mod Kit.

0. About this tutorial

I've written this tutorial so people can have a small taste and understanding of the possibilities of analysing firmware.

There's a whole new world out there for you if you learn how to use and master this very specific skillset.

Even though the security is a laugh, you should know that this is a .bin of the latest firmware (today) of a commonly sold device. I have renamed the files and directories in this tut because I do not want to cause harm to any vendor or product. I just want to teach you about the (in)securities in some systems such as IP-camera's, cell-phones, TV's, routers,... and why you should be careful when you're using them.

1. Know Your Tools

Don't try to invent the wheel again.

There's plenty of wheels out there that are available for you to use.

Tools used

0. Your favorite search engine (for firmware)

1. [Binwalk](#)
2. [Firmware Mod Kit](#)
3. A bit of common sense

Spoiler ([Click to View](#)) Note: I'm assuming you know how to apt-get on Linux and how to extract a tarball, if you are having difficulty, feel free to ask below or to PM me about it.

2. Let's get started!

So, let's start by having a look at binwalk!

Just read through the --help or man page so you have an idea of what the possibilities are, I will not cover all the flags anyways but most of them should be quite obvious.

Spoiler ([Click to View](#))

I'll just elaborate on 3 of these for now because I'll be using them later on: -v, --dd and -e.

-v

I think this is pretty obvious, verbose mode just spits out more information.

It won't always be useful, but the more information you can gather, the merrier.

--dd

This is probably my favorite flag.

It allows you to specify a certain filetype, and extracts it out of the binary in a dd like manner. So no

more counting bytes and finding right offsets!

The way I'm going to use it here is

Code `--dd=<filetype>:<extension>` Which will basically extract any <filetype> to a file with a specified <extension>.

-e

-e is the "please do everything for me"-equivalent of --dd.

This will extract any filetype to the extension it "feels" it should match.

(I personally prefer --dd over -e, especially after 1 once accidentally used -e on a bin which contained about 400 .gifs next to the actual filesystem.)

When you run `binwalk -v filename` you should see something like this (if the binary you have is as simple as mine, otherwise it'll probably contain a lot more "binwalk-entries").

Spoiler [\(Click to View\)](#)

Usually, the filesystem is most interesting for what we're gonna do in this tutorial (finding configuration mistakes), but using `binwalk --dd`, I'll extract both of them, just in case I want to dig deeper later.

Spoiler [\(Click to View\)](#)

A Bin is like a box of chocolates, you never know what you're gonna get.

After you managed to pull out the filesystem, you should have a look at what Firmware Mod Kit can do for you...

Spoiler [\(Click to View\)](#)

Spoiler [\(Click to View\)](#)...and start hating me because that kinda automates everything I have done with binwalk up till now. ?

Firmware Mod Kit is a set of scripts that make things a lot easier when you're doing this. It contains tools for most filesystems you'll run into and it will do what it has to most of the time.

This is also where the common sense comes in handy.

You've extracted a squashfs file system so the most logical thing to do would be to run the `unsquashfs_all.sh` script right?

I can teach you the tricks and the tools, but I can't teach you how to think, please take enough time to do that yourself ?

That being said:

Spoiler [\(Click to View\)](#)

The scripts require root privileges because they'll need to perform actions such as creating and reading root directories, mounting images, etc.

If you want to have a better view of what exactly it is they do, you can open them with your favorite text editor.

And before you know it, you have access to the file system and all the files on it.

Spoiler [\(Click to View\)](#)

From there on, as you can imagine the possibilities are endless.

(Especially if there's built-in web interface for remote configuration on the device whose cgi scripts I'm gonna analyse thoroughly tomorrow ?)

Thank you for taking the time to read this. I hope I have informed you enough, for you to take matters into your own hands when it comes to securing these devices.

If you have any remarks or questions, feel free to PM me or post below!

Peace!?

Midas

PS: Big thank you to [this man](#) who inspired me to do this and for writing both Binwalk and FMK.

dotcppfile3 - October 20, 2014, 12:32 am

Nice tut mate, keep these threads coming.

KingMidas - October 20, 2014, 2:43 am

(Oct 20, 2014 12:32 AM)dotcppfile3 Wrote: [\[CITE\]](#) Nice tut mate, keep these threads coming.

Thank you! Will do some editing later (spotted a few typo's) and I'll look for another subject to write about.

Maybe about finding vuln's in the .cgi scripts but that's a first for me so I'll have to look into it!

Edit: think I've got most typo's out by now!

?

HeX_Replays - October 20, 2014, 4:54 pm

thinks for the tut i will be using this

ht275 - October 20, 2014, 6:37 pm

Thanks It was helpfull But i still struggled using it a little

KingMidas - October 21, 2014, 12:52 am

(Oct 20, 2014 06:37 PM)ht275 Wrote: [\[CITE\]](#) Thanks It was helpfull But i still struggled using it a little
Hi!

Anything in specific that gave you problems?

tatosoppelsa - August 13, 2016, 3:06 pm

Hi Midas!

First of all, thanks for your post, was really helpfull. I'm trying to extract firmwares to mount the Web user interface in a local http server. Any ideas?At this time I cannot extract .cgi files with "binwalk -e"
. Hope you still there :|

Tato

lakyet1984 - August 21, 2016, 8:13 am

nice tutorial ... thanks for sharing