Rooting the MikroTik routers

A journey into reverse engineering parts of MikroTik system to gain access to hardware features and the shell behind the RouterOS that has no “ls”

possible.lv

SHA +2017

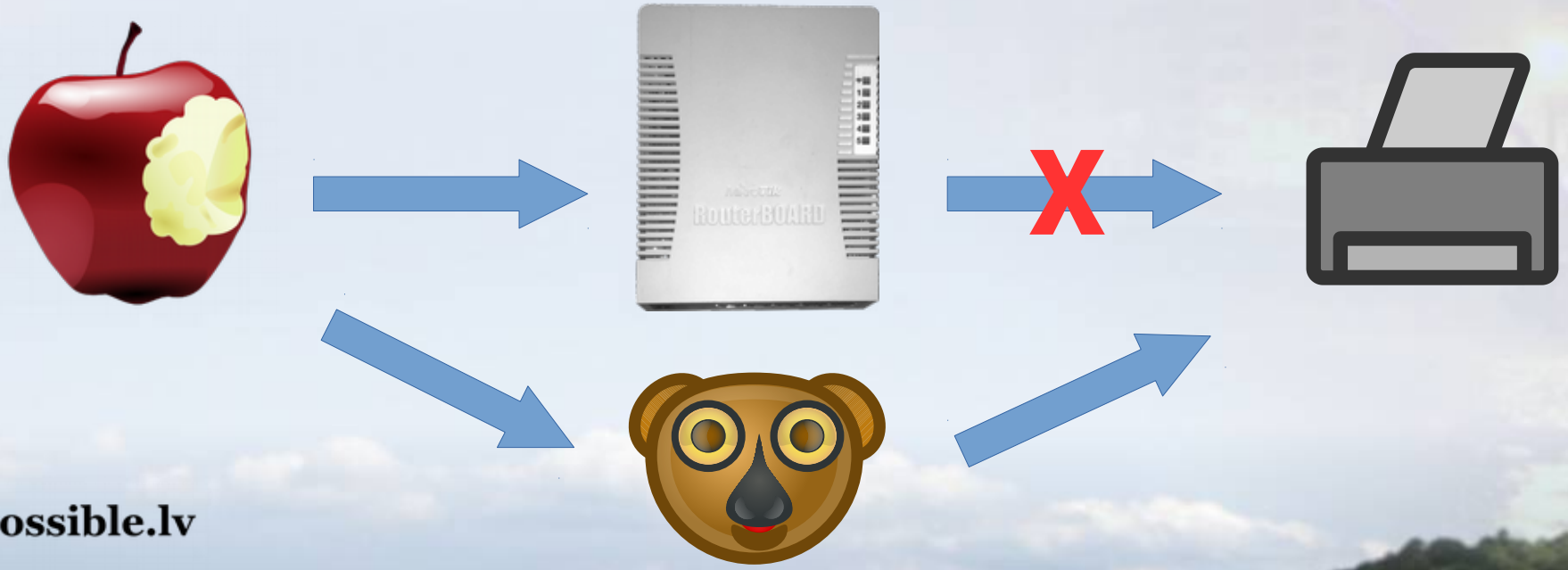
Who? Me?

- **Who am I?**
  - https://twitter.com/KirilsSolovjovs
- **What do I do?**
  - http://kirils.org/

possible.lv

Legal disclaimer

Goal of this research is to achieve the interoperability of computer programs (i.e. software running on MikroTik routers) with other computer programs.



possible.lv

ACK: Prior research

- “antony++” from awmn.net
  - Initial NPK analysis
- “drubicza”
  - NPK file unpacking
- OpenWRT team
  - kernel config files

ACK: The team

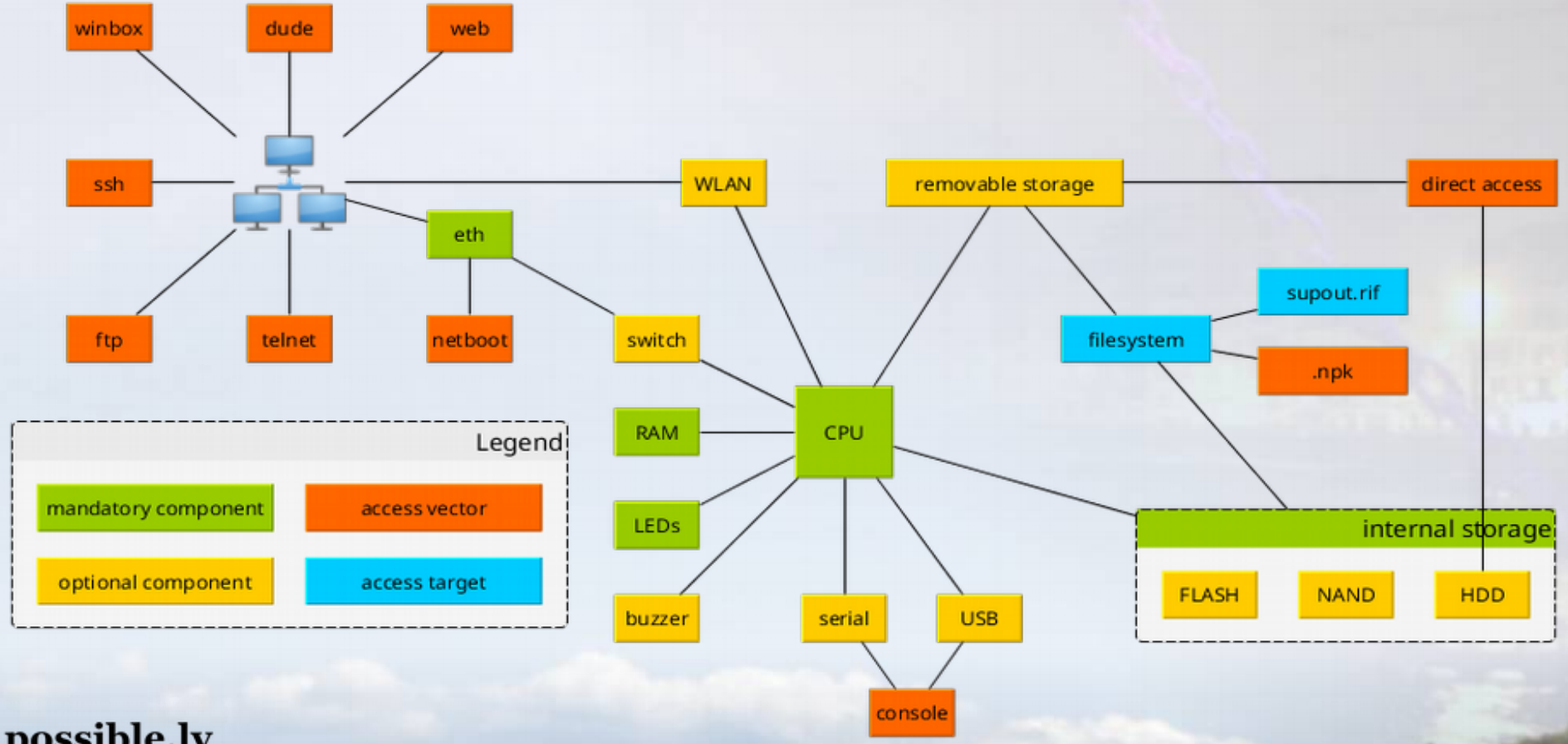
- **Kirils Solovjovs**
  - dynamic binary analysis, jailbreak scripts
- **Jānis Jansons**
  - static binary analysis, bootup sequence
- **Emīls Romanis**
  - music

Content outline

- RouterOS overview
- Reversing supout.rif
- NPK format
- Rooting the router

RouterOS overview

RouterOS ecosystem



possible.lv

NETWORK SYSTEM

TCP/IP protocol suite

DHCP
DHCP server per interface

Routing p
RIP 1 / 2
OSPF v2
protocol st

PPP
ISDN dial-
ISDN dial-
RADIUS a
onboard s
modem pc

Bridging
spanning t
multiple br

Firewall
packet filte
static NAT
masquera

Tunnels
PPTP enc
IPIP tunne

NETWORK INTERFACES

Wireless
Aironet 655/ IC 2000/ IC 2200 ISA
Aironet 4800 / Cisco 340 ISA/PCI/PC
WaveLAN 802.11 bronze / silver / gold ISA / PC
Samsung SWL - 2000 P (v2.1 August 2000) PCI
Compaq WL - 200 (v2.1 August 2000) PCI

Asynchron
Moxa PCI 8
Equinox PC

Synchrono
Moxa C101
MicroGate
LMC 1000
LMC 5245P DS3
LMC 5200P HSSI
LMC 1200P E1/T1 (v 2.1 September 2000)

Ethernet
ISA NE2000 compatible
PCI NE2000 compatible
Four port D-Link 10/100 DFE 570TK
RTL 8139 10/100

HARDWARE REQUIRED

CPU motherboard	486/dx or better
RAM	16 MB or more
hard disk/Flash IDE	32 MB or more
for installation	floppy drive
	keyboard
	monitor

possible.lv


LUCIFER
DRAGONFLY
ES DOM
PS

RouterOS history

- 1999
  - MikroTik™ v2.0 Router Software
    - initial release
    - works on 486
    - upgrades available as packages
- 2000
  - MikroTik™ v2.1 Router Software
    - according to marketing

RouterOS history

- **2001**
  - MikroTik™ v2.2 Router Software
  - MikroTik™ v2.3 Router Software
    - npk first mentioned as method for extending functionality
- **Jan 2002**
  - MikroTik RouterOS™ V2.4
  - RouterOS is finally born!

RouterOS history

- **Aug 2002**
  - MikroTik RouterOS™ V2.5
- **Dec 2002**
  - MikroTik RouterOS™ V2.6
- **Dec 2003**
  - MikroTik RouterOS™ V2.7

- **You've reached the most boring slide. Apologies.**
- **And congrats – next slides will be less boring.**

RouterOS history

- **12 Feb 2004**
  - MikroTik RouterOS™ V2.8
    - software key system changed
    - key algorithm has not been changed since
- **1 Aug 2005**
  - MikroTik RouterOS™ V2.9
    - new architecture introduced
      - mipsel for RB500

RouterOS history

- **15 Nov 2005**
  - 2.9.8
    - a wild “/nova/etc/devel-login” appears in /nova/bin/login
    - [-f /nova/etc/devel-login && username == devel && password == admin.password] && /bin/sh
- **20 Feb 2008**
  - 2.9.51
    - … is as high as 2.9 branch goes

possible.lv

RouterOS history

- **15 Jan 2008**
  - 3.0
- **mid-2008**
  - around 3.10
    - anotony++ releases createnpk.py and dumpnpk.py on the forums of Athens Wireless Metropolitan Network
    - allows to not only unpack npk, but also create your own

RouterOS history

- **8 Feb 2009**
  - 3.21
    - what's up with this version?
    - why has it vanished from the internet?
- **16 Mar 2009**
  - 3.22
    - npk verification and signing added
    - checksum and signature checked by /nova/bin/installer
    - no more free lunches

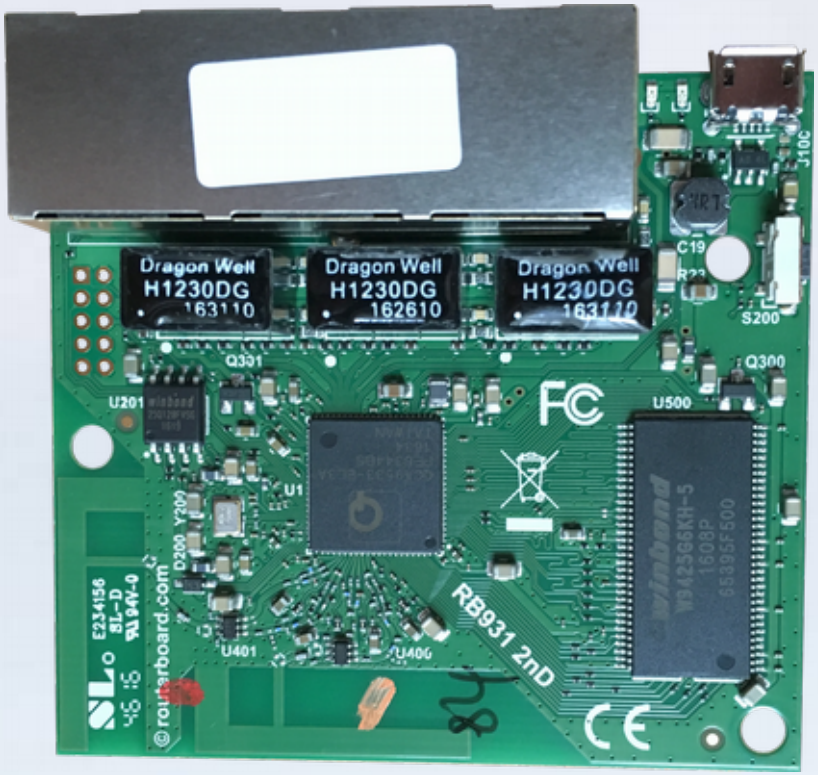
RouterOS history

- **12 Oct 2009**
  - 4.0
- **31 Mar 2011**
  - 5.0
    - release cycle is getting slower...
- **7 May 2013**
  - 6.0 (since beta3)
    - SquashFS employed in npk files
    - zerofill blocks added
      - so that actual SquashFS start is located at addresses divisible by 4096

RouterOS history

- **8 Jul 2015**
  - 6.30
    - sha1 digest block (ascii) added to npk files
    - format suggests it's not being used for verification, probably just for identification
- **6 Nov 2015**
  - 6.33
    - packages now include distribution channel
      - bugfix | current | development | release-candidate

hAP mini (RB931-2nD)



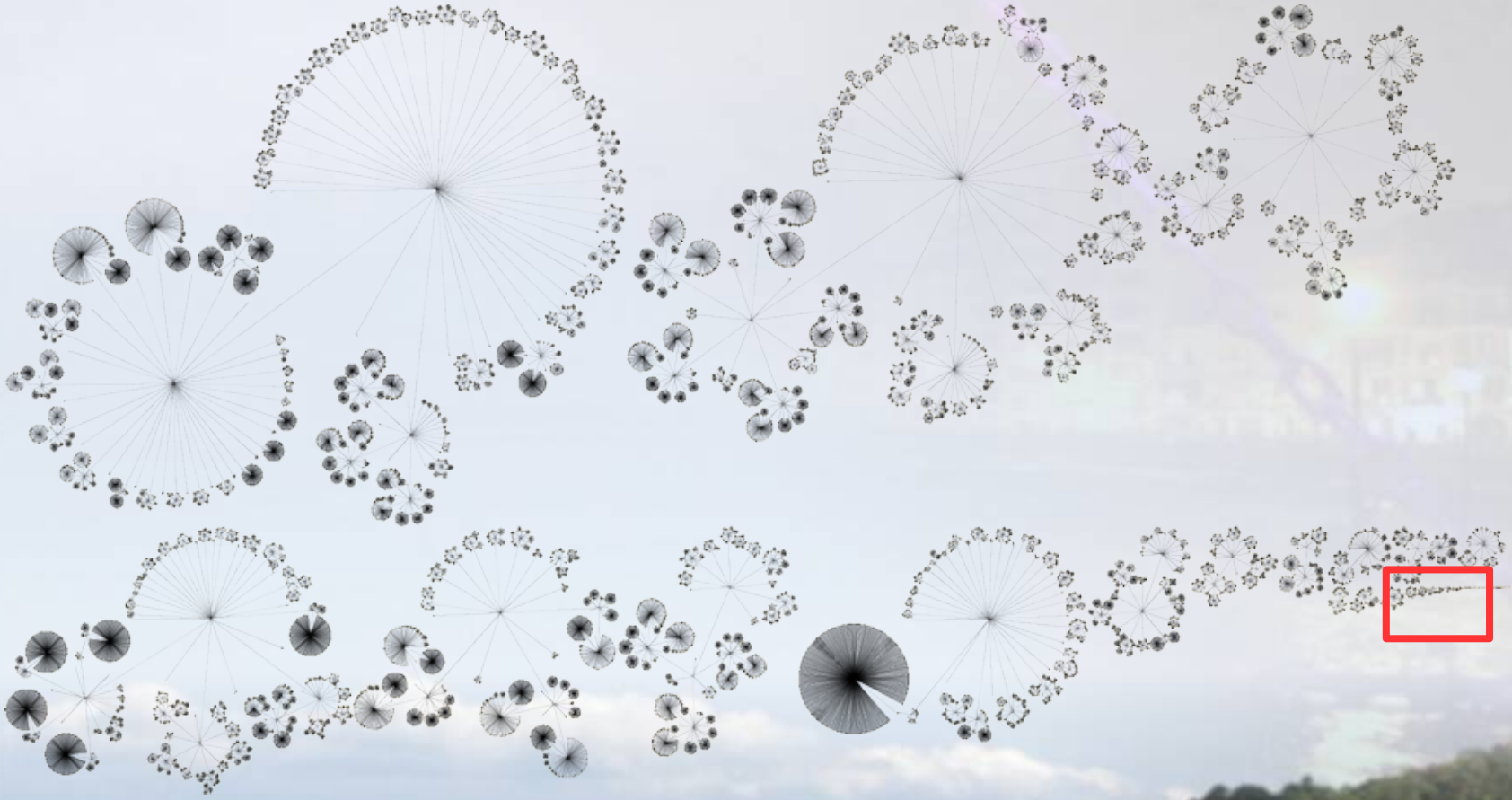
possible.lv


LUCIFER
DRAGONFLY
ESCOM
PS

RouterOS command tree

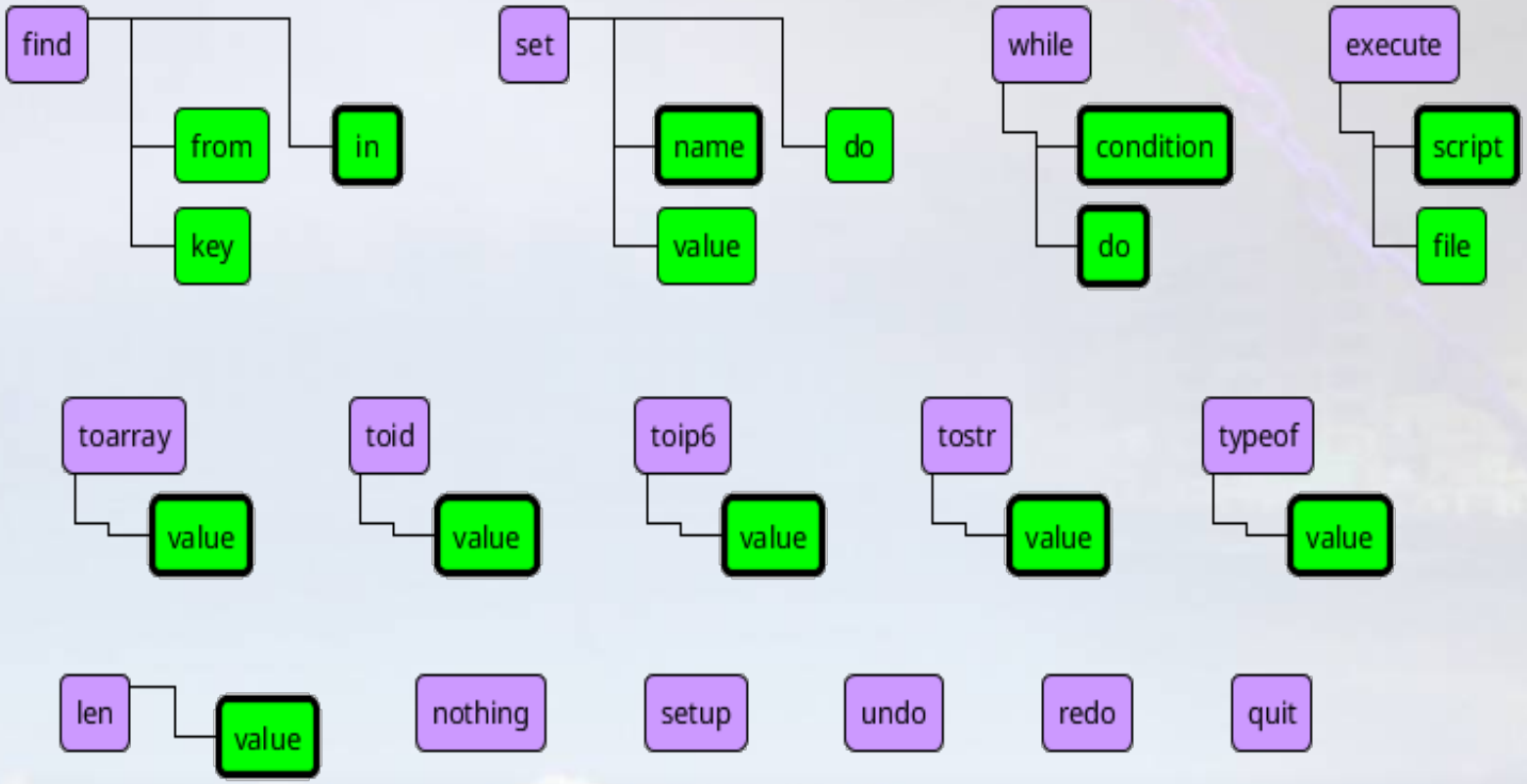
xviewer memory requirements (.png)

/ip	3.7 GiB
/interface	3.5 GiB
/routing	2.1 GiB
/tool	1.9 GiB
/system	1.2 GiB
/caps-man	1.1 GiB
/ipv6	0.9 GiB



possible.lv

RouterOS command tree [15/62]



Reversing supout.rif

SUPOUT.RIF FROM OUTSIDE

```
--BEGIN ROUTEROS SUPOUT SECTION
oVWYsRHaAgHnjXuAAAgJAgB=
--END ROUTEROS SUPOUT SECTION
--BEGIN ROUTEROS SUPOUT SECTION
w9WZt8Wd0BAecukSMFFSO/czNx8SRh8SM3UVog8TVBNyJz8SVBjBKROlmbeKYkxayFAAccOD1D==
--END ROUTEROS SUPOUT SECTION
--BEGIN ROUTEROS SUPOUT SECTION
sNGZ09WdjhGA4x58xZXUwdX9z1gc0HFC21QCxT/cPYe5KpETRhkzP3cTMvUUIvEzNVFyJ5UUQjcy
MvUVwEgSkTp5mnCmoJXAAsy1S0E=
--END ROUTEROS SUPOUT SECTION
```

- **each section decodes to:**
  - **name + ’\0’ + zlib_compressed_content**

SUPOUT.RIF FROM INSIDE

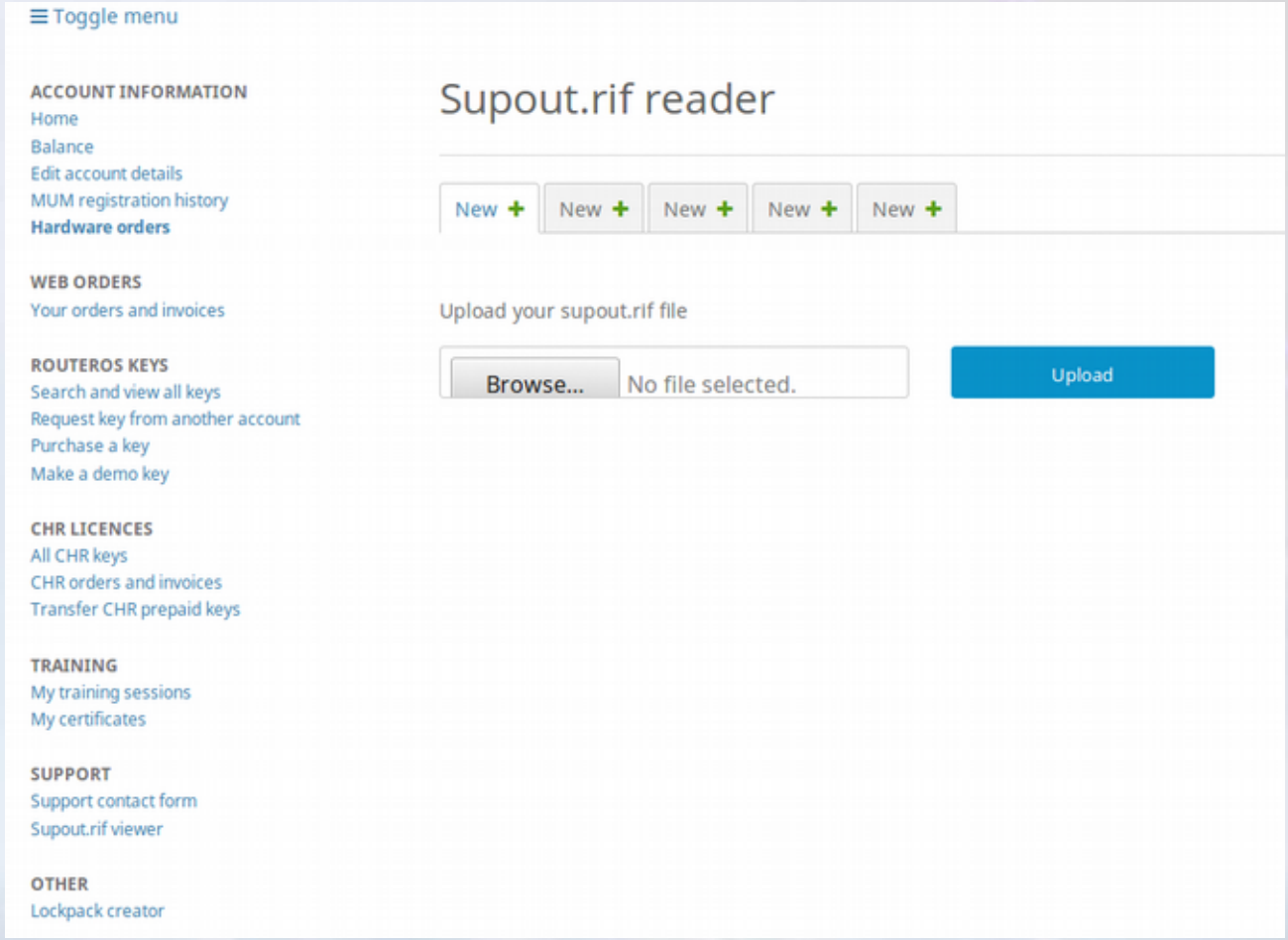
- **What does it contain?**
  - your whole configuration
  - /proc/ folder
  - memory addresses
  - your log
  - and more

```
$ ls supout.rif_contents/
01_.debug       16_arp              31_profile        46_wirelesselog
02_.profile     17_ip               32_dhcp           47_bfd
03_.proc        18_nexthop          33_neighbor       48_bgp
04_.startup     19_route            34_dhcp6          49_mme
05_.livetrace   20_user             35_license        50_mpls
06_resource     21_firewall         36_package        51_ntp-client
07_pci          22_firewall-stats   37_instchk        52_ospf
08_usb          23_bridge           38_oops           53_ppp
09_log          24_mesh             39_backtrace      54_ipsec
10_export       25_queue            40_store          55_health
11_interface    26_queue-packets    41_hotspot        56_poe-out
12_ethernet     27_queue-bytes      42_routerboard    57_lcdtouch
13_switch       28_queue-stats      43_webproxy
14_address      29_ippool           44_wireless
15_port         30_certificate      45_wirelessdump
$
```

DEMO

Demo: decode_supout.py

MIKROTIK.COM ALSO HAS A READER ...



possible.lv

... but it won't show you everything

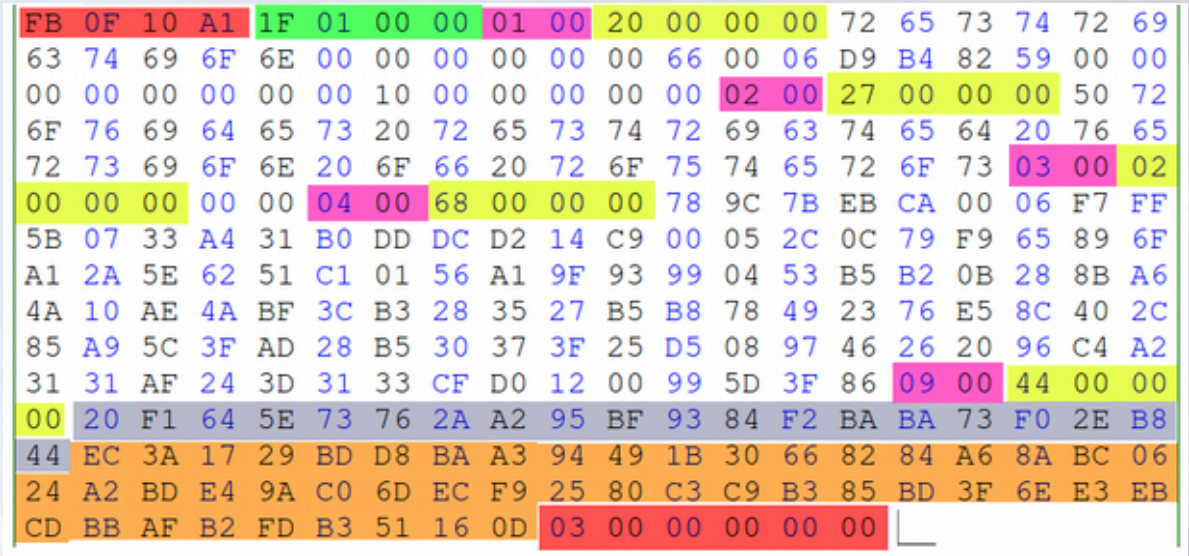
Demo: supout_m.rif

Demo: supout_show.php

NPK FORMAT

NPK FORMAT

- Numeric values are unsigned little endian
- File consists of header, file size, parts and footer.
- File size is 8B less
- Each part consist of:
  - part type (short)
  - payload size (long)
  - payload



possible.lv

NPK FORMAT

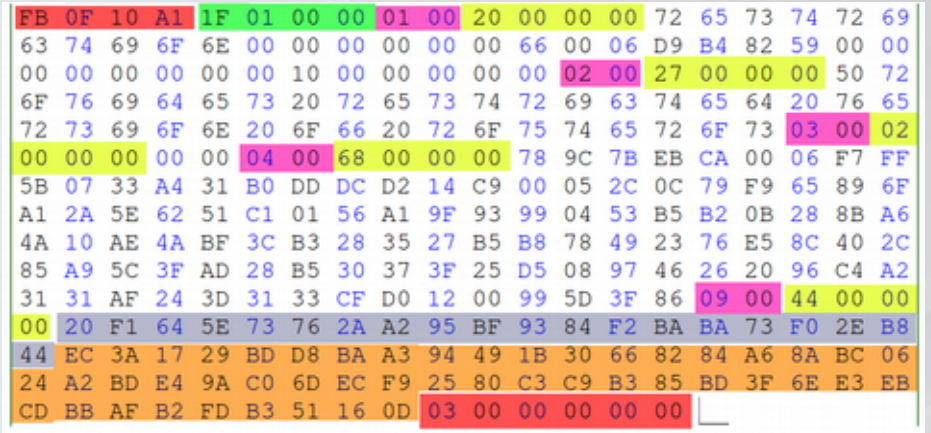
- **At least two types of current NPKs:**
  - package
    - 0..3 header 1E F1 D0 BA
    - footer 10 00 01 00 00 00 49
      - footer since 3.22
  - restriction (invisible package)
    - 0..3 header FB 0F 10 A1
    - footer 03 00 00 00 00 00

Part types

N	Type	Meaning	First seen	Last seen	Mandatory
1	01 00	Part info	forever	now	yes
2	02 00	Part description	forever	now	yes
3	03 00	Dependencies	forever	now	yes
4	04 00	File container	forever	now	no
5	05 00	Install script (libinstall)	forever	2.7.xx	no
6	06 00	? Uninstall script (libinstall)	never	never	no
7	07 00	Install script (bash)	forever	now	no
8	08 00	Uninstall script (bash)	forever	now	no
9	09 00	Signature	3.22	now	yes
10	0a 00	unused	never	never	no
11	0b 00	unused	never	never	no
12	0c 00	unused	never	never	no
13	0d 00	unused	never	never	no
14	0e 00	unused	never	never	no
15	0f 00	unused	never	never	no
16	10 00	Architecture	2.9	now	yes
17	11 00	Package conflicts	3.14	3.22	no
18	12 00	Package info	2.9	now	no
19	13 00	Part features	2.9	now	no
20	14 00	Package features	2.9	now	no
21	15 00	SquashFS block	6.0beta3	now	package only
22	16 00	Zero padding	6.0beta3	now	no
23	17 00	Digest	6.30	now	package only
24	18 00	Channel	6.33	now	package only

Part 0x09 – signature

- **Packages are signed (since 3.22)**
  - broken packages will not be installed
- **Part type 09 00 – signature**
- **Size – always 0x44**
- **First 20 bytes**
  - sha1sum of everything from the previous part 01 00 (including part type & size) up to 09 00 44 00 00 00
- **Remaining 48 bytes - unknown signature**
  - Last byte always less than 0x10
  - Verified based on public key or seed C2 75 D7 23 57 66 AE C8 66 D4 C5 95 73 C8 E1 88 A5 13 39 93 6E 94 D2 CC F1 1F 9F F5 BA ED 71 37



possible.lv

PART 0x17 - DIGEST

- Size – 0x28 (40 bytes)
- ascii representation of a SHA1 hash
- most likely used here as UUID

Rooting the router

Getting shell

1) Create /nova/etc/devel-login
2) telnet to 192.168.88.1 as devel
   - yaay! :)
3) ls
   - fail :(

```
$ telnet 192.168.88.1
Trying 192.168.88.1...
Connected to 192.168.88.1.
Escape character is '^]'.

MikroTik v6.39.2 (stable)
Login: devel
Password:


BusyBox v1.00 (2017.05.31-11:35+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

# ls
bash: ls: not found
#
```

[TAB] to the rescue

```
# cat
bin/    boot/   etc/    home/   nova/   proc/   rw/     sys/    usr/
bndl/   dev/    flash/  lib/    pckg/   ram/    sbin/   tmp/    var/
# ls
bin     boot    dude    flash   lib     pckg    ram     sbin    tmp     var
bndl    dev     etc     home    nova    proc    rw      sys     usr
#
```

- **No ls? No problem!**
  - cat, space, tab, tab
- **Or, you know, do it properly, and upload busybox**
  - statically linked, for the right architecture
    - uname -m
  - this might be of interest:
    - https://busybox.net/downloads/binaries/1.21.1/

Can we speed this up?

- **Of course.**
- **A VirtualBox appliance!**
  - does ~~all~~ most of the work for you
- **This should work out nicely***
  - If your CPU is AR9344 and device has at least two ethernet ports
    - RB951G-2HnD, RB951Ui-2HnD <== tested
    - CRS109-8G-1S-2HnD-IN, CRS125-24G-1S-IN, CRS125-24G-1S-2HnD-IN
    - RB2011L, RB2011LS, RB2011iLS-IN, RB2011iL-IN, RB2011UiAS-IN RB2011UiAS-RM, RB2011UiAS-2HnD-IN
    - OmniTIK 5, OmniTIK 5 PoE

MT_JB_0.81_fin.
ova

How to use the appliance

Demo: MT_JB_0.81_fin.ova

1) **Import the appliance**
2) **Make sure bridged network card is set to ethernet**
3) **Disconnect all wires from the router, power it up**
4) **Start the virtual machine and follow instructions**
5) **Be ready to swiftly re-plug the cable when prompted**

Yes, yes, that's nice, but ...

- **Can my RouterBOARD play Für Elise?**
- **Let's see and listen!**


Für Elise
Demo: elise.sh

FIN

- **Tools (will be) available**

  https://github.com/0ki/

- **Didn't manage to ask your question? Wanna hang out?**
  - call 4488
  - tweet @KirilsSolovjovs
  - mail sha2017 at kirils org
  - meet SpeakerDesk

possible.lv