possible.lv

Tools for effortless reverse engineering of MikroTik routers

HACKTIVITY

v3

https://github.com/0ki/mikrotik-tools
http ://kirils.org/

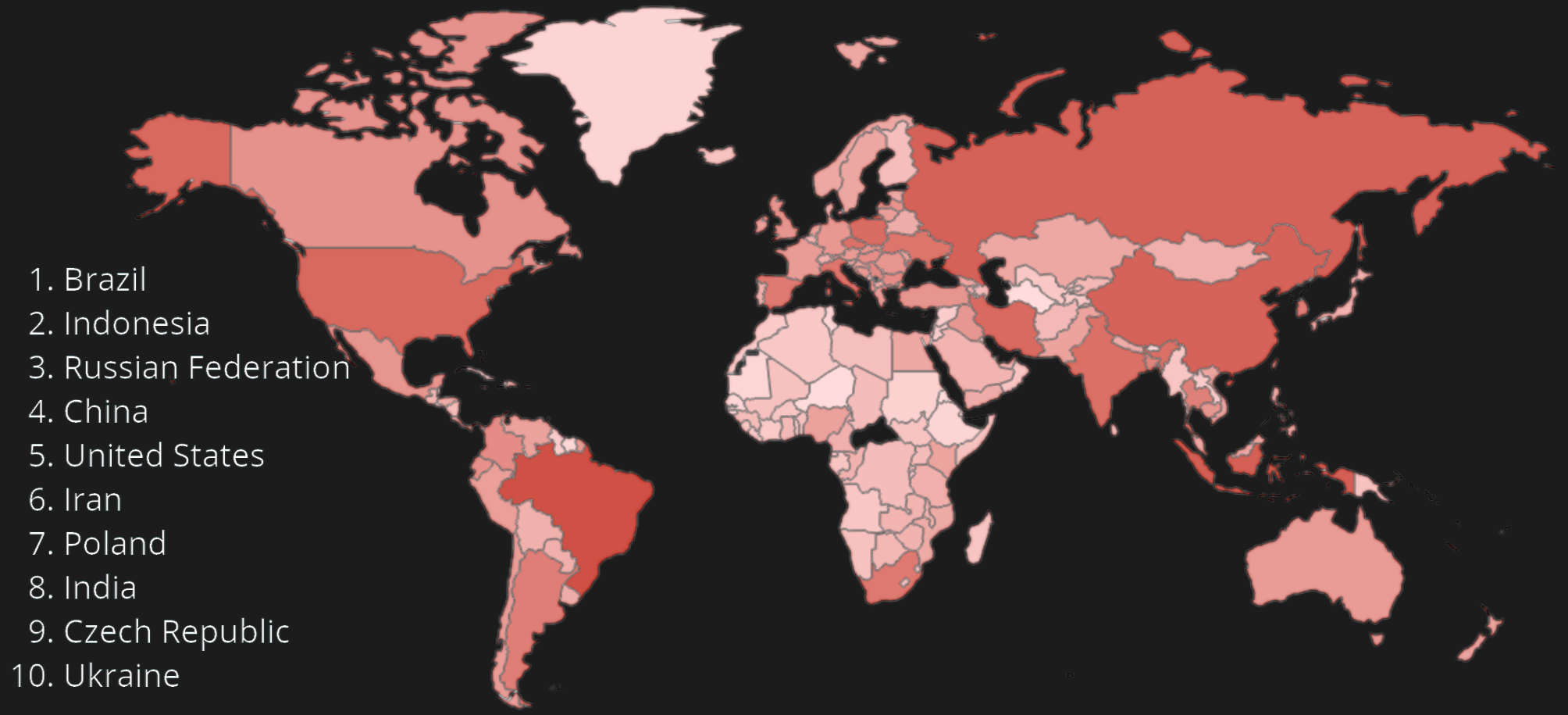
possible.lv

Legal disclaimer

Goal of this presentation is to allow the members of the research community to assess security and achieve the interoperability of computer programs

MikroTik? Anyone even uses it?

possible.lv



1. Brazil
2. Indonesia
3. Russian Federation
4. China
5. United States
6. Iran
7. Poland
8. India
9. Czech Republic
10. Ukraine

possible.lv

RouterOS is ...

- Linux!
  - old Linux!
- Startup scripts
- Nova binaries
- Config

```
# uname -a
Linux MikroTik 3.3.5 #1 Thu Aug 24 10:36:14 UTC 2017 i686 GNU/Linux
```

Secure | https://www.kernel.org/pub/linux/kernel/v3.x/

linux-3.3.4.tar.xz	27-Apr-2012 17:46
linux-3.3.5.tar.bz2	07-May-2012 16:15
linux-3.3.5.tar.gz	07-May-2012 16:15
linux-3.3.5.tar.sign	07-May-2012 16:15
linux-3.3.5.tar.xz	07-May-2012 16:15

And it's also closed source & closed ecosystem

possible.lv

A jailbreak is needed...

possible.lv

possible.lv

A built-in backdoor. How nice.

- RouterOS 2.9.8 delivered on 15 Nov 2005
  - a wild “/nova/etc/devel-login” appears in /nova/bin/login
  - [-f /nova/etc/devel-login && username == devel && password == admin.password] && /bin/ash

possible.lv

All we gotta do is ...

All we gotta do is ...

1) Create /nova/etc/devel-login
2) telnet to 192.168.88.1 as devel

```
$ 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

possible.lv

```
# 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/

But how... ?

possible.lv

The old way

possible.lv

- A VirtualBox appliance!
- DEMO

The old way

possible.lv

- A VirtualBox appliance!
- Works only if
  - 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

possible.lv

The new way

- A bash/python script
- Works regardless of architecture
- Very fast
- Can do remote jailbreaks
- Will not help you recover lost passwords
- Will probably get patched soon after this presentation
- DEMO

Now. The tools.

possible.lv

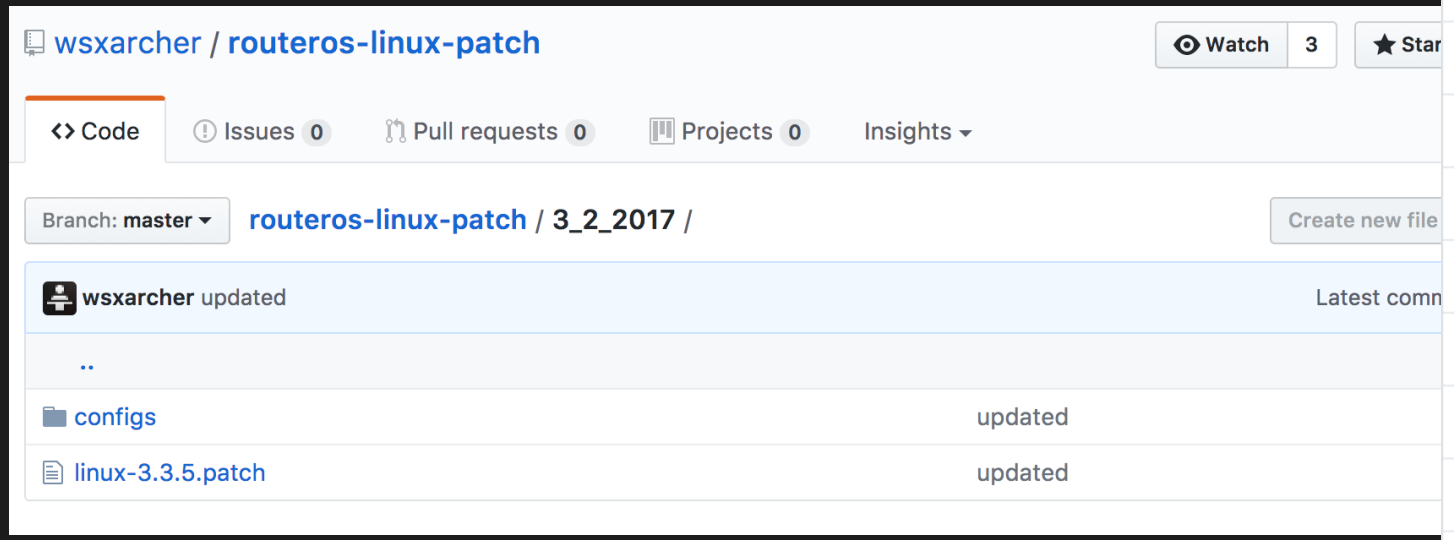
NPK file sourcing

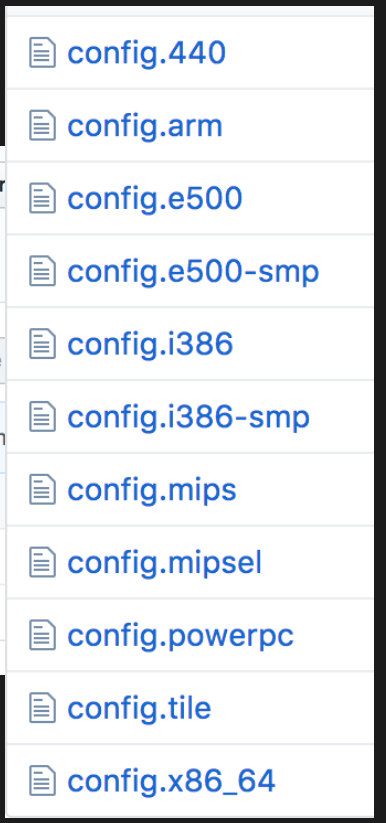
possible.lv

- getnpk.sh
  - deps: wget
- reversenpk.sh
  - deps: unsquashfs, unnpk
    - https://github.com/rsa9000/npk-tools

Kernel patches

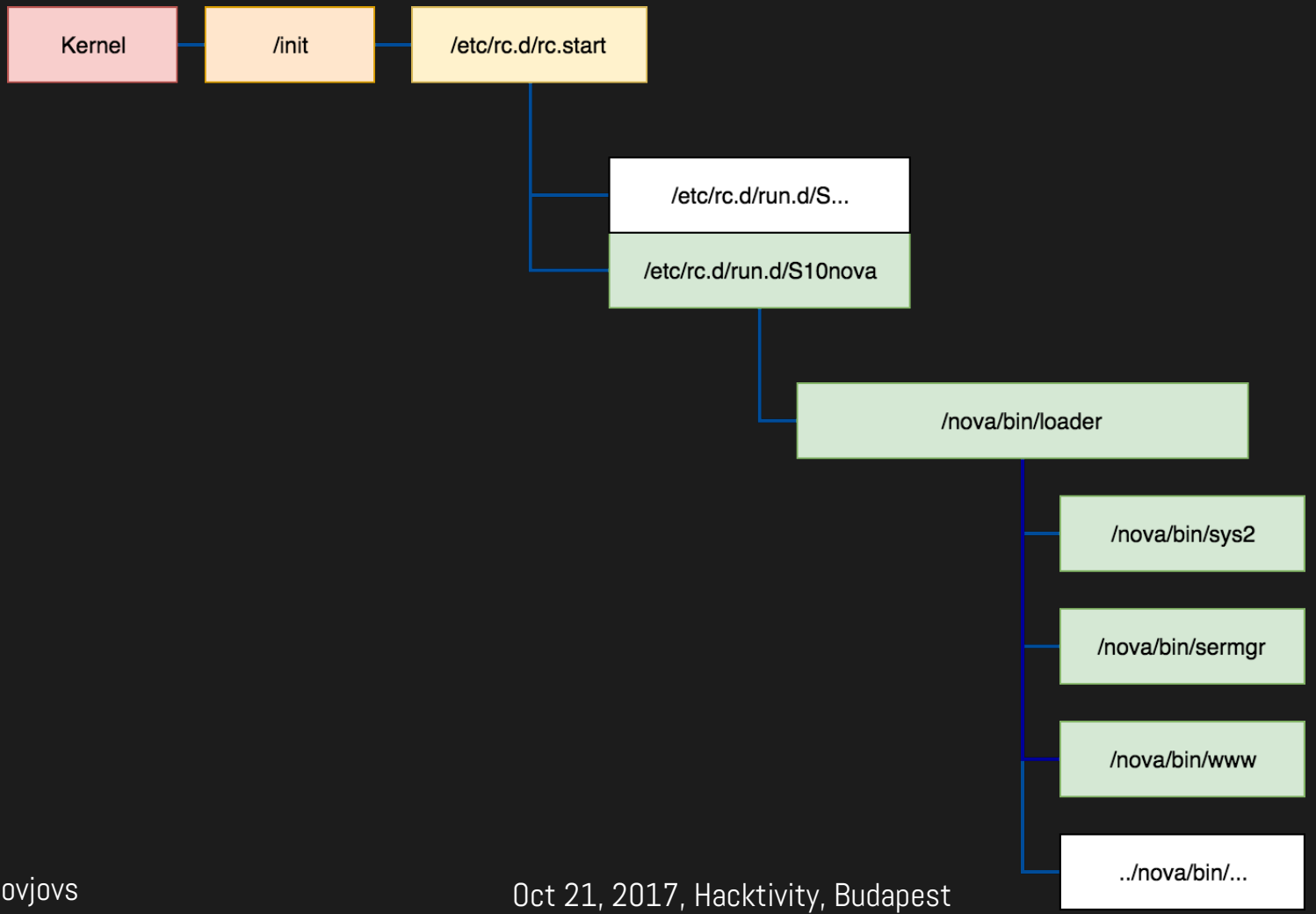
possible.lv





https://github.com/wsxarcher/routeros-linux-patch

RouterOS boot process



Where do we put custom binaries?

possible.lv

```
#!/bin/bash

echo "Starting..."

/etc/rc.d/rc.sysinit || exit 1

export PATH=$(path --colon /sbin):$(path --colon /bin)
export LD_LIBRARY_PATH=/rw/lib:$(path --colon /lib)

# disable core files
ulimit -c 0

# set maximum opened files per process
ulimit -n 500000

# start syslog deamon
for i in $(path --prefix S /etc/rc.d/run.d); do
    if [ -f $i ]; then
        $i || exit 1
    fi
done

exit 0
```

/etc/rc.d/rc.start

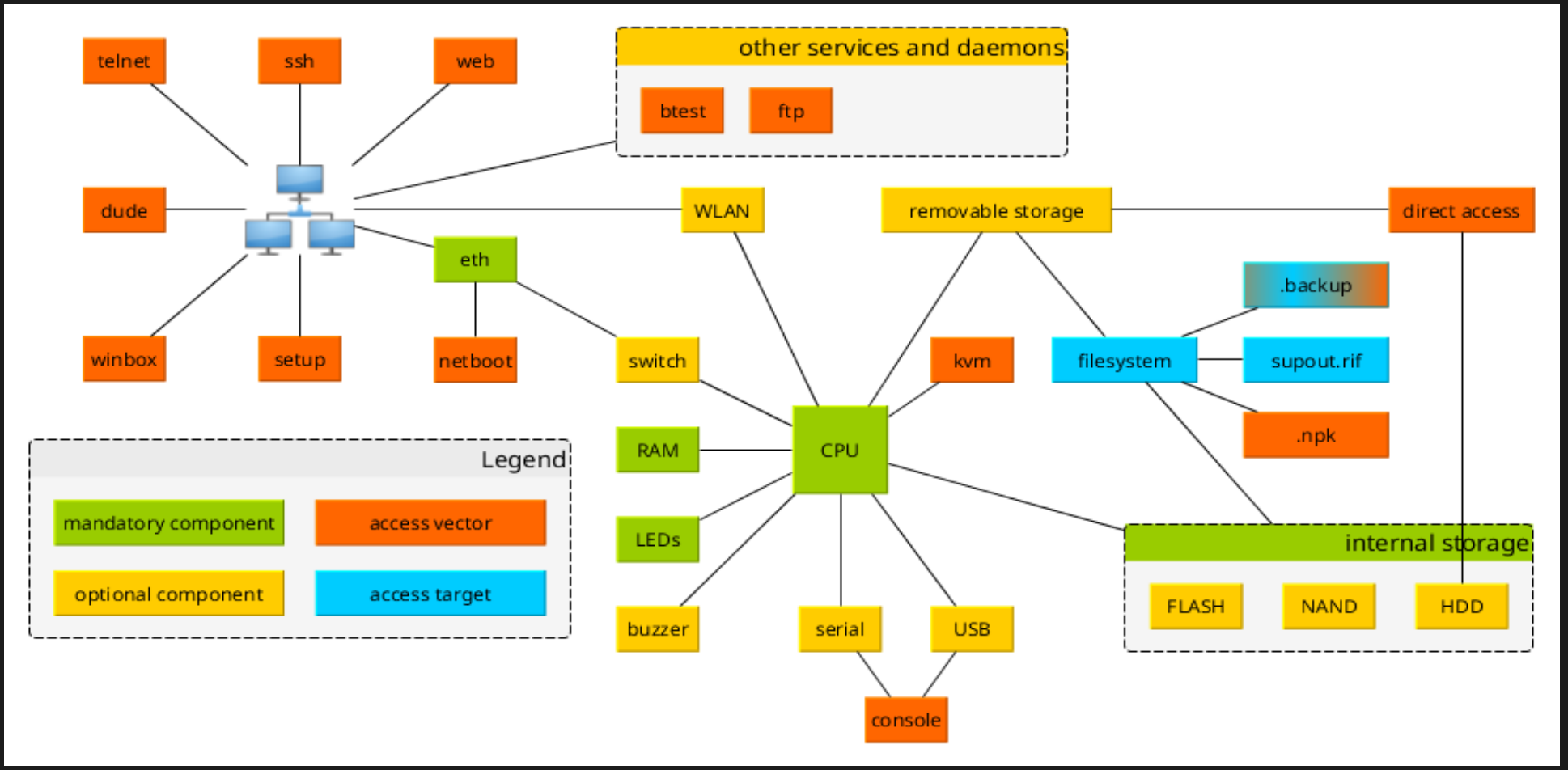
possible.lv

Anywhere!

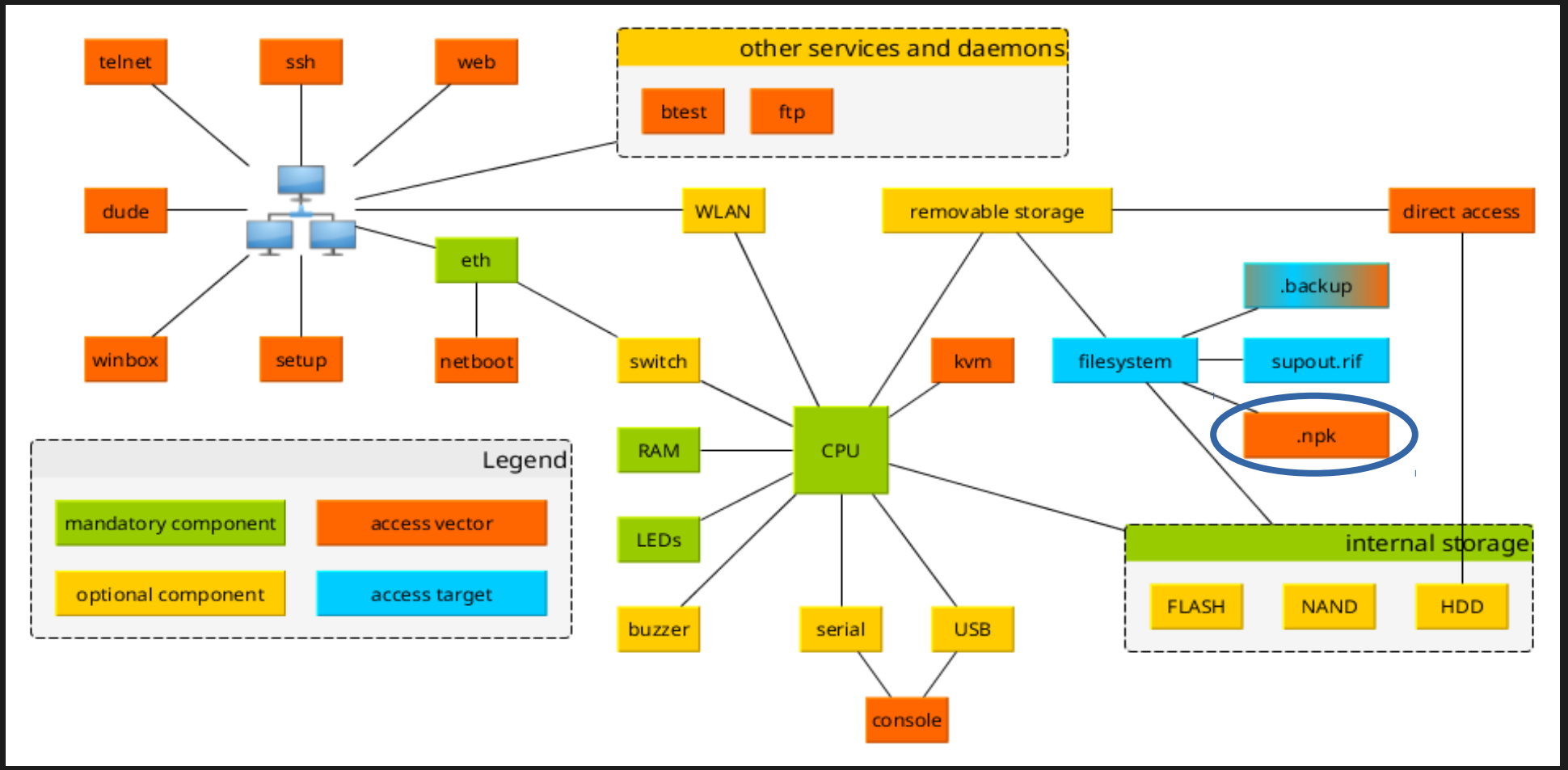
- “path” looks for specified path in prefixed directories
  - Used throughout their scripts
  - Makes using custom scripts easier

```
# path --colon /etc/rc.d/
/pckg/dude/etc/rc.d/:/pckg/ntp/etc/rc.d/:/pckg/ups/etc/rc.d/:/pckg/ipv6/etc/rc.d/:/pckg/security/etc/rc.d/:/etc/rc.d/
# path --colon /bin/
/flash//bin/:/pckg/kvm/bin/:/bin/
#
```

High level overview of RouterOS

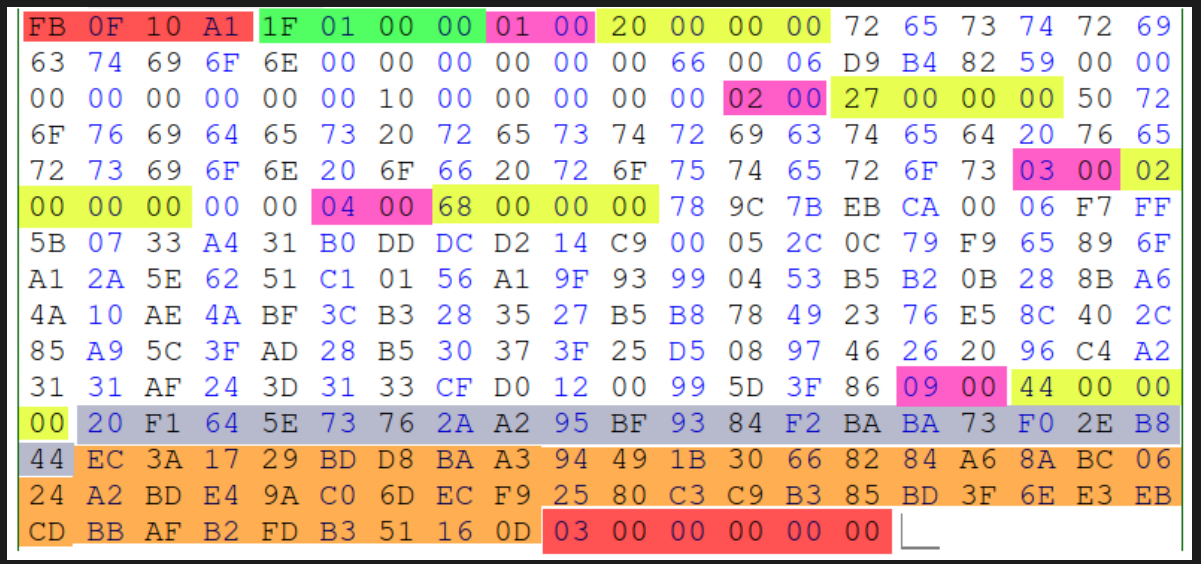


NPK format


telnet
ssh
web
other services and daemons
btest
ftp
dude
WLAN
removable storage
direct access
eth
.backup
winbox
setup
netboot
switch
kvm
filesystem
supout.rif
.npk
RAM
CPU
Legend
mandatory component
access vector
optional component
access target
LEDs
internal storage
FLASH
NAND
HDD
buzzer
serial
USB
console

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



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

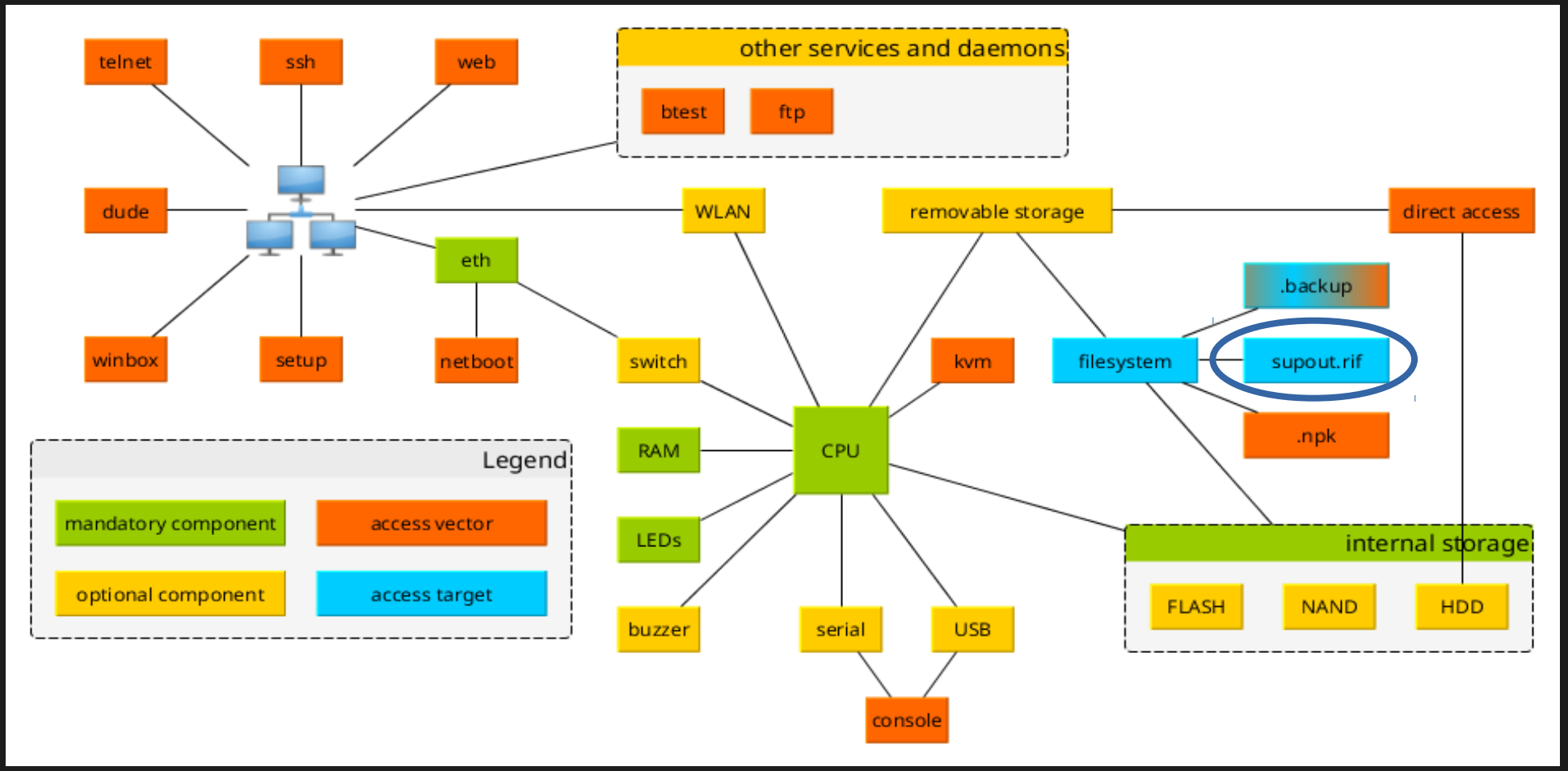
```
[admin@MikroTik] > /system package
[admin@MikroTik] /system package> print
Flags: X - disabled
 #   NAME                      VERSION
 0   system                    6.38.4
 1 X restriction               6.0
[admin@MikroTik] /system package>
```

Part types

possible.lv

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

supout.rif


telnet
ssh
web
other services and daemons
btest
ftp
dude
WLAN
removable storage
direct access
eth
.backup
winbox
setup
netboot
switch
kvm
filesystem
supout.rif
.npk
RAM
CPU
Legend
mandatory component
access vector
optional component
access target
LEDs
internal storage
FLASH
NAND
HDD
buzzer
serial
USB
console

What is supout.rif?

possible.lv

- Support output
  - ridiculously intricate format
  - or RouterOS information file, maybe, idk ¯_(ツ)_/¯

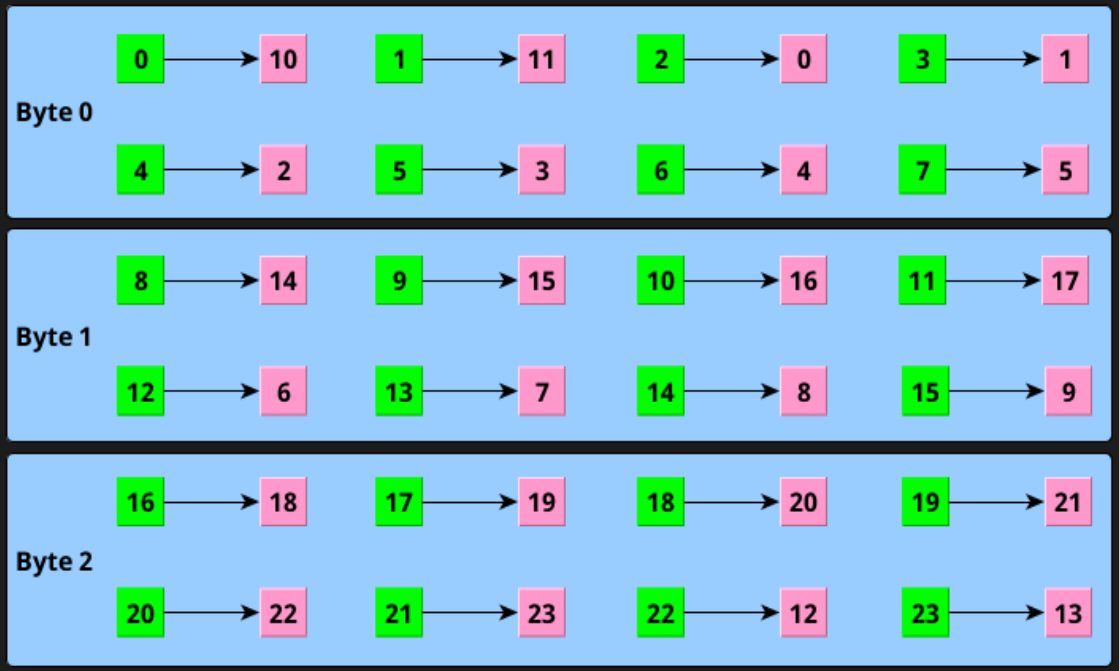
```
[admin@MikroTik] > /system sup-output
  created: 1%
-- [Q quit|D dump|C-z pause]
```

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
```

supout.rif section decoding

- swap bits around
  - per three bytes
- base64
- section decodes to:
  - name + ‘\0’ + zlib_compressed_content



supout.rif section decoding

possible.lv

```
tribitmap=[10,11,0,1,2,3,4,5,14,15,16,17,6,7,8,9,18,19,20,21,22,23,12,13]

def tribit(content):
    result=""
    for i in xrange(0, len(content) - 1,3):
        goodtribit=0
        badtribit=ord(content[i])*0x10000+ord(content[i+1])*0x100+ord(content[i+2])
        for mangle in tribitmap:
            goodtribit = (goodtribit<<1) + (1 if ((badtribit & (0x800000>>mangle))>0) else 0)

        for move in [16,8,0]:
            result=result+chr((goodtribit >> move)& 0xff)

    return result
```

supout.rif from inside

possible.lv

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

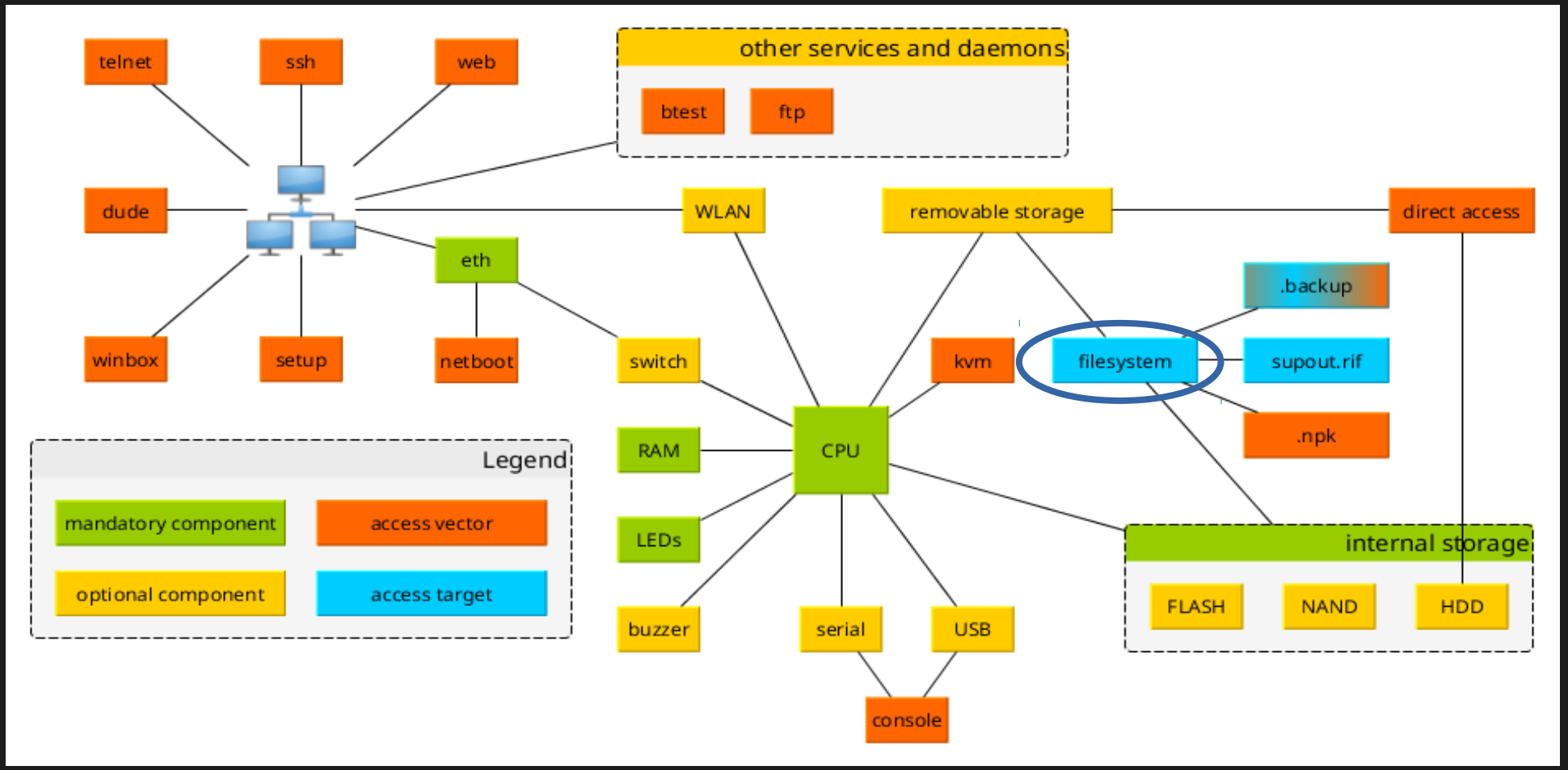
```
$ ls supout.rif_contents/
01_.debug       16_arp              31_profile        46_wirelesselog
02_.profile     17_ip               32_dhcp           47_bfd
03_.proc        18_nexthop          33_neighbor       48_bgp
04_.startup     19_route            34_dhcp6          49_mme
05_.livetrace   20_user             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: mikrotik.com xss

Demo: decode_supout.py

Config files



Configuration

- Config is stored in /rw/store as pairs of files
  - IDX = index
  - DAT = data

```
batman
bgconf
bserv.dat
bserv.idx
cerm.dat
cerm.idx
cert
cmanifacecf
cmanifacecf
command
dhcp
diskd
dude
echosave
graphing
group.dat
group.idx
hotspot
igmpproxy
ipsec
log-actions.idx
log-rules.dat
rip
smbusers.idx
snmp-communities.dat
snmp-communities.idx
snmpd.dat
snmpd.idx
ssh
sstp
system.dat
system.idx
tftp.dat
tftp.idx
um4.dat
um4.idx
unicl
user
user.dat
user.idx
webproxy
wireless.dat
wireless.idx
wirelessccl.dat
```

IDX format

possible.lv

- Record ID (long)
  - if ID is 0xFFFFFFFF, field has no content
  - used for offsetting
- length (long)
- separator (long)
  - usually 0x05000000

DAT format

- LENGTH (short)
- M2 RECORD of length
  - Config ID (3 bytes)
  - type (1 byte)
    - content depends on to type

```
btype ........
      0000000, - boolean
      ,,1,1,,, - M2 block (len = short)
      ,,11,,,, - binary data block (len = short)
      ,,,,,,,1 - one byte
      ,,,,,,1, - ???
      ,,,,,1,, - ???
      ,,,11,,, - 128bit int
      ,,,,1,,, - int (four bytes)
      ,,,1,,,, - long (8 bytes)
      ,,1,,,,, - string
      ,1,,,,,, - ??? unused? or long array of?
      1,,,,,,, - short array of

types (MT notation)
    (CAPITAL X = list of x)

    a,A, (0x18) IPv6 address (or duid)
    b,B, bool
      M, multi
    q,Q, (0x10) big number
    r,R, (0x31) mac address
    s,S, (0x21) string
    u,U, unsigned integer
```

Peculiarities / features

possible.lv

- Field IDs shared with web
- Winbox protocol derived from DAT format
  - Working directly with files?
  - Dangerous!

mt_dat_decoder.py module

possible.lv

- DEMO

Where's my password?

possible.lv

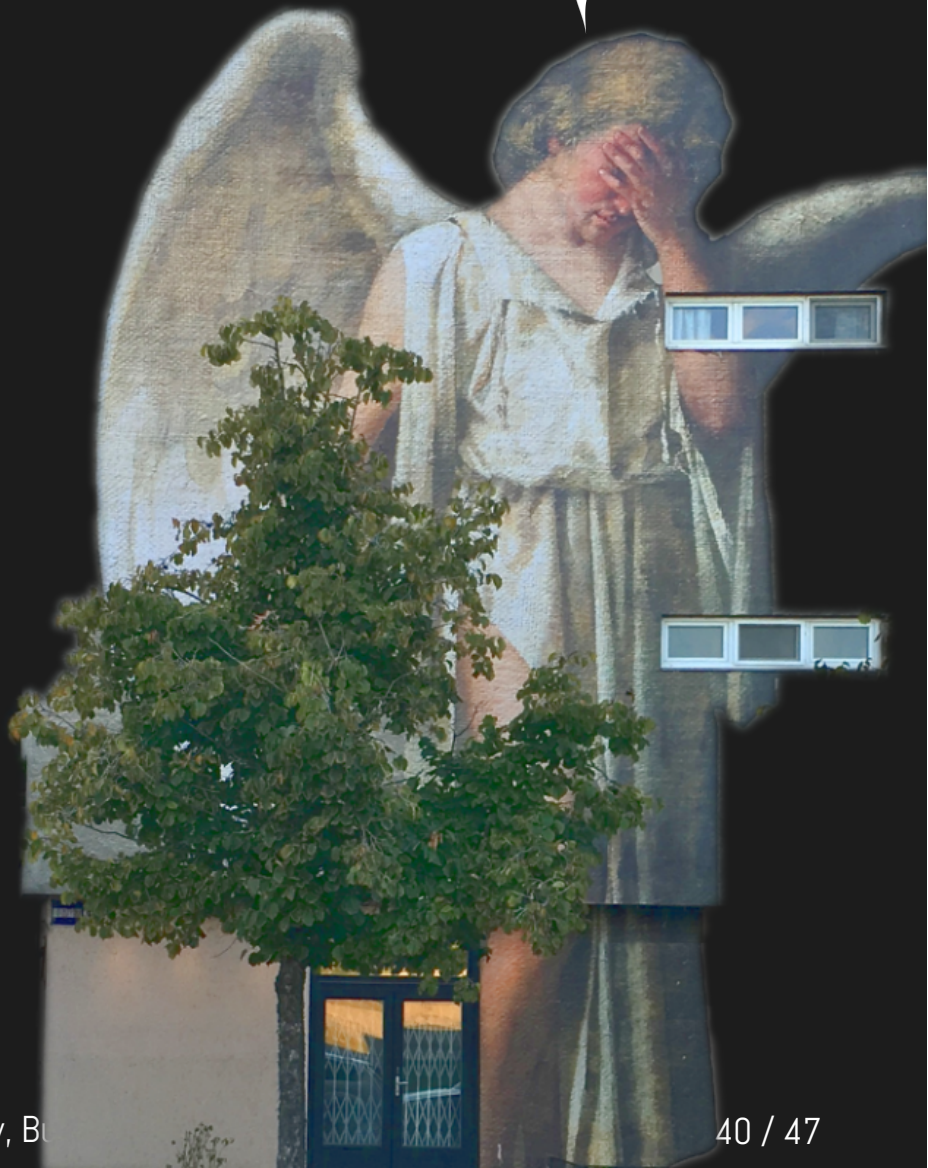
- Calm down! It's encrypted!

```
# ls /rw/store/
batman                  log-actions.idx         smbusers.idx
bfd                     log-rules.dat           snmp-communities.dat
bgconf                  log-rules.idx           snmp-communities.idx
bserv.dat               mactel.dat              snmpd.dat
bserv.idx               mactel.idx              snmpd.idx
cerm.dat                mcast                   ssh
cerm.idx                mpls                    sstp
cert                    mproxy.dat              system.dat
cmanifacecfg.dat        mproxy.idx              system.idx
cmanifacecfg.idx        net                     tftp.dat
command                 ospfconf                tftp.idx
dhcp                    ospfv3                  um4.dat
diskd                   ovpn                    um4.idx
dude                    ppp                     unicl
echosave                pptp                    [illegible]
graphing                radius                  user.dat
group.dat               radius.dat              user.idx
group.idx               radius.idx              [illegible]
hotspot                 radvd                   wireless.dat
igmpproxy               resolver                wireless.idx
ipsec                   rip                     wirelessccl.dat
```

The password is

- hashed
- salted
- md5

- Oh, wait, no. That's the key.



possible.lv

The password

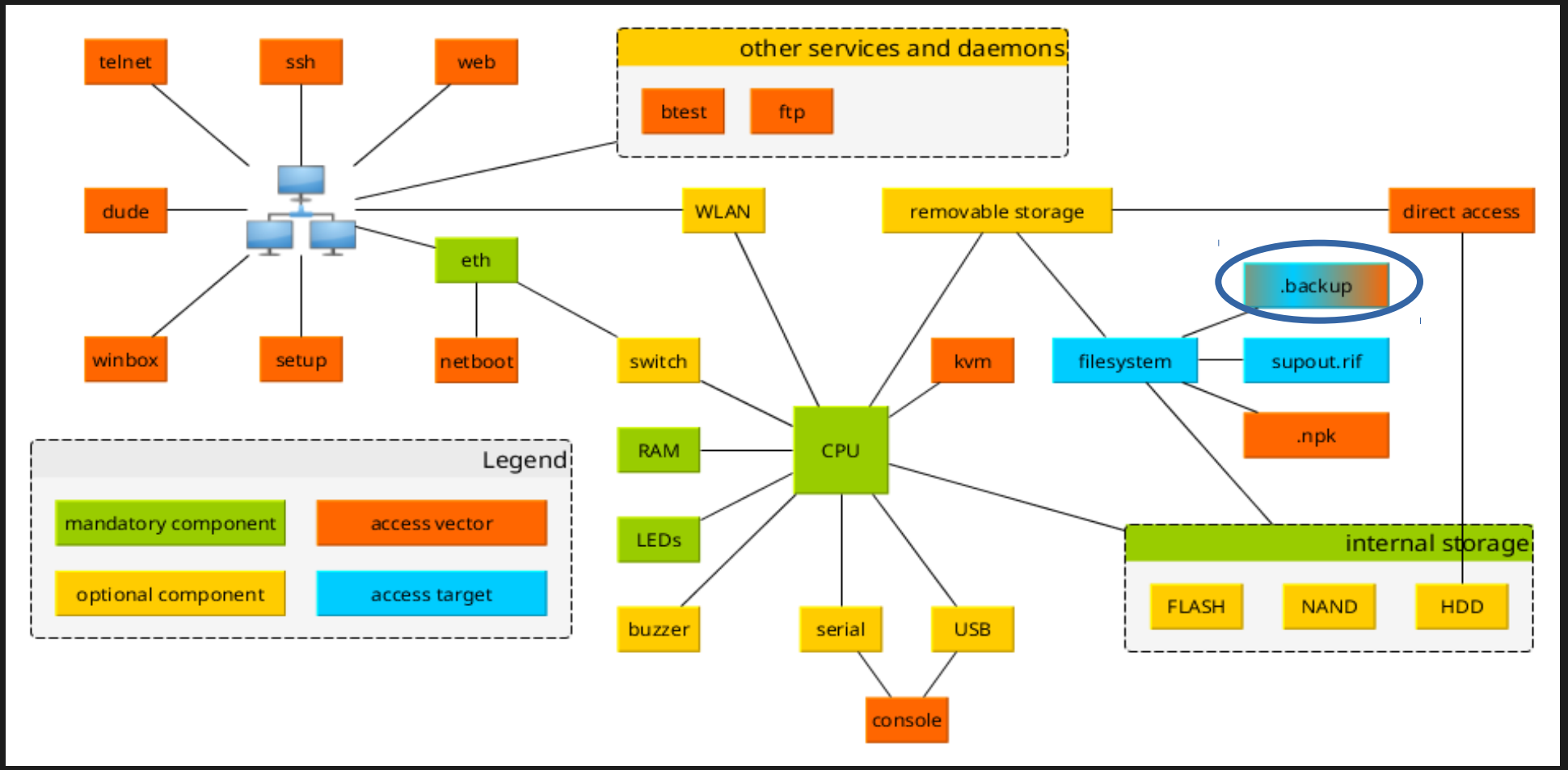
key = md5(username + "283i4jfkai3389")

password = user["password"] xor key

The password tool

- DEMO

Backup files



Backup file layout

possible.lv

- Header (long)
  - 0x88ACA1B1 – backup
  - 0xEFA89172 – encrypted backup
- Length of backup file (long)
- Records of:
  - Path name, idx contents, dat contents
- Each record consists of length (long) and binary data

The bug

- mkdir –p pathname(“/flash/rw/store/”+filename)
- write idx to “/flash/rw/store/”+filename+“.idx”
- write dat to “/flash/rw/store/”+filename+“.dat”

decode_backup.py

- DEMO

possible.lv

The end.

- Tools & jailbreak available

  https://github.com/0ki/mikrotik-tools
- Latest appliance:

  http://02.lv/f/2017/09/15/MT_JB_0.89.ova