Scripting on RouterOS For fun and $profit

Presenter: Andrew Cox

Who Am I?

Name: Andrew Cox
Location: Adelaide/Brisbane, Australia

Working for:
www.bigair.net.au - Network Engineer
www.bacb.com.au - Senior Hotspot Engineer

Consulting, Blog and Podcast

www.mikrotik-routeros.com - Consulting

www.thebrotherswisp.com - WISP Podcast

RouterOS Scripting: What?

- on router scripting language

- no external server required

- local scheduler for repeatable events

- access to all terminal usable commands

RouterOS Scripting: Why?

Example uses:

- modifying queues or routing based on bandwidth usage
- automating events that would require manual intervention (outages / errors)
- creating complex trigger systems for alerting (if bandwidth reaches X for Y mins)
- backup and setup procedures (automated router backup email)
- troubleshooting assistance (ping this for me!)

Who actually uses scripting?

Mikrotik:
/system default-configuration print

Mikrotik Wiki/Forum Users:
http://wiki.mikrotik.com/wiki/scripts
100+ user contributed scripts

http://forum.mikrotik.com/viewforum.php?f=9
3300 threads, 16000 posts

Online:
Google search - "mikrotik script"
Over 9000 results

Ok, so how does it work?

Simple Terminal commands:

```
/queue simple add target=192.168.1.100
```

Same thing in Scripting?

```
/queue simple add target=192.168.1.100
```

Lets look at some of the scripting commands

Some Basic Scripting Commands

All prefixed with ':'

:local	*Define a script local variable*
:global	*Define a global variable*
:set	*Assign a variable value*
:put	*Output to the terminal*
:resolve	*Return IP address of a DNS name*
:log	*Add a log entry*

Basic Scripting Example

Resolve an address and add the IP to an address list:

```
:local server "www.mikrotik.com"
:local ipaddress
:set $ipaddress [:resolve $server]
/ip firewall address-list add list=example \
    address=$ipaddress comment="$server"
:log info "Added: $server as $ipaddress"
```

Loops and Conditional Operators

Functions that allow repetitive action and queries.

:for *Performs an action for given number of executions*
:do :while *Perform action against a check*
:foreach *Perform action for each matching*
:if *Perform if condition is met*

Beginner Scripting Example

```
/queue simple add target=192.168.1.100
```

Remember this? How can we save time and perform this for 100 addresses..

```
:local x
:for x from 100 to 200 do={/queue simple
add target-address="192.168.1.$x"}
```

Now you're thinking with scripts!

Lets Review

We can:
- collect data
- modify items
- do tasks en masse

What else?

...

Advanced Scripting Example

```
:local vpninterface "pptp-interface"
:local vpndns "supervpn.awesomecompany.tld"
:local newvpnip [:resolve $vpndns]
:local currentvpnip [/interface pptp-client get $vpninterface connect-to]
:if ($currentvpnip != $newvpnip) do={/interface pptp-client set [find name=$vpninterface] connect-to=$newvpnip}
```

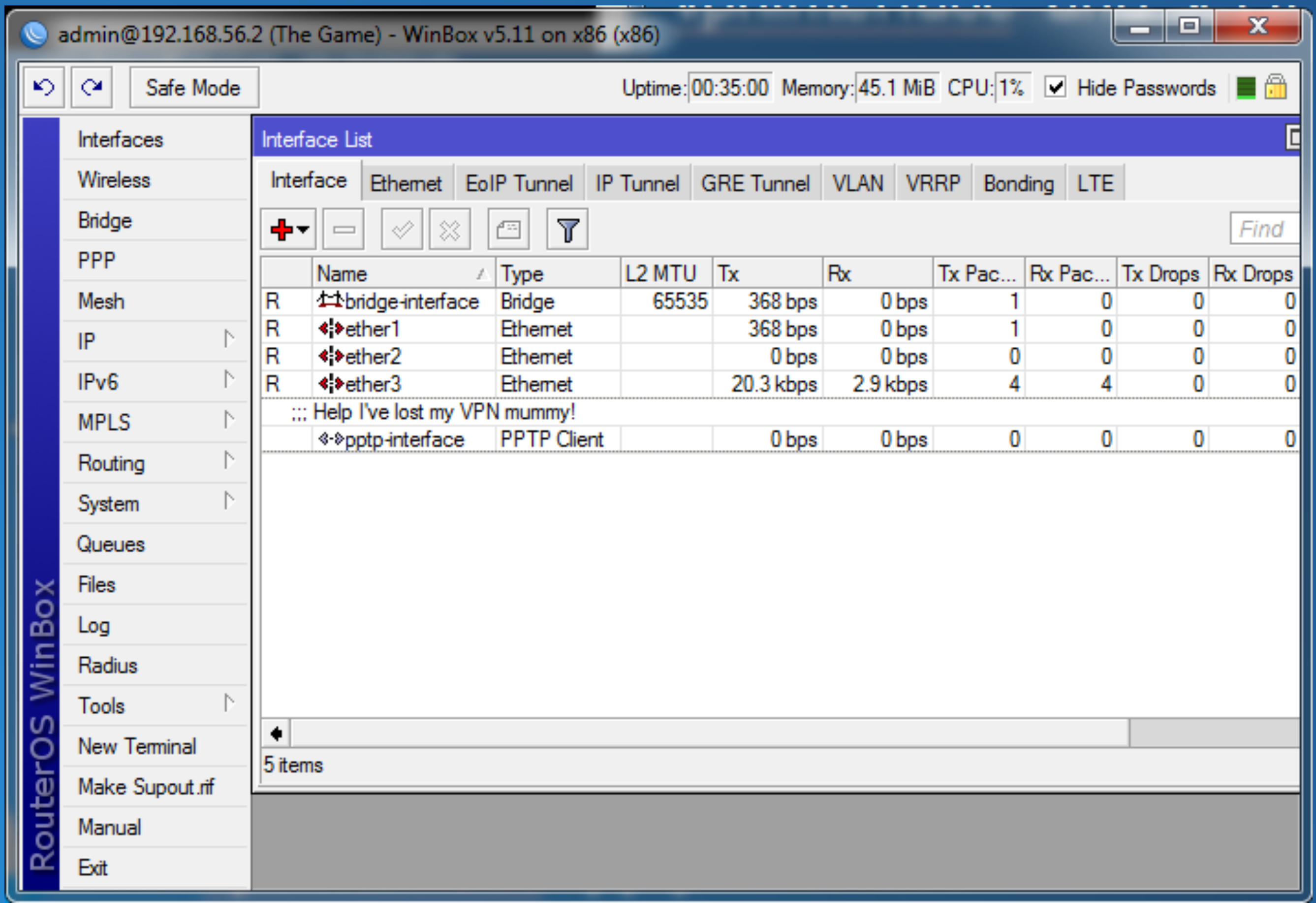
Ok, but what does it do?

...

Advanced Scripting Example: breakdown 1/5

Define a new variable 'vpninterface' and set it to your VPN interface name

```
:local vpninterface "pptp-interface"
```



Uptime: 00:35:00 Memory: 45.1 MiB CPU: 1% ☑ Hide Passwords

	Name	Type	L2 MTU	Tx	Rx	Tx Pac...	Rx Pac...	Tx Drops	Rx Drops
R	bridge-interface	Bridge	65535	368 bps	0 bps	1	0	0	0
R	ether1	Ethernet		368 bps	0 bps	1	0	0	0
R	ether2	Ethernet		0 bps	0 bps	0	0	0	0
R	ether3	Ethernet		20.3 kbps	2.9 kbps	4	4	0	0
	;;; Help I've lost my VPN mummy!								
	pptp-interface	PPTP Client		0 bps	0 bps	0	0	0	0

5 items

...

Advanced Scripting Example: breakdown 2/5

Define a variable to hold your VPN server DNS name

```
:local vpndns "supervpn.awesomecompany.tld"
```

Image Source: http://commons.wikimedia.org/wiki/File:Server-vpn.svg
By RRZEicons [CC-BY-SA-3.0 (http://creativecommons.org/licenses/by-sa/3.0)], via Wikimedia Commons

Advanced Scripting Example: breakdown 3/5

Resolve the VPN domain name to an IP address

```
:local newvpnip [:resolve $vpndns]
```

supervpn.awesomecompany.tld

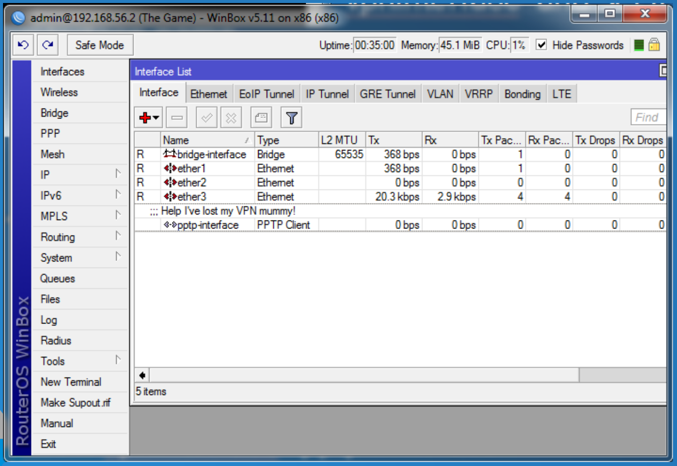
198.51.100.123

...

Advanced Scripting Example: breakdown 4/5

Grab the current IP address set from the VPN client interface, searching for it using the interface name we already know.

```
:local currentvpnip [/interface pptp-client get $vpninterface connect-to]
```



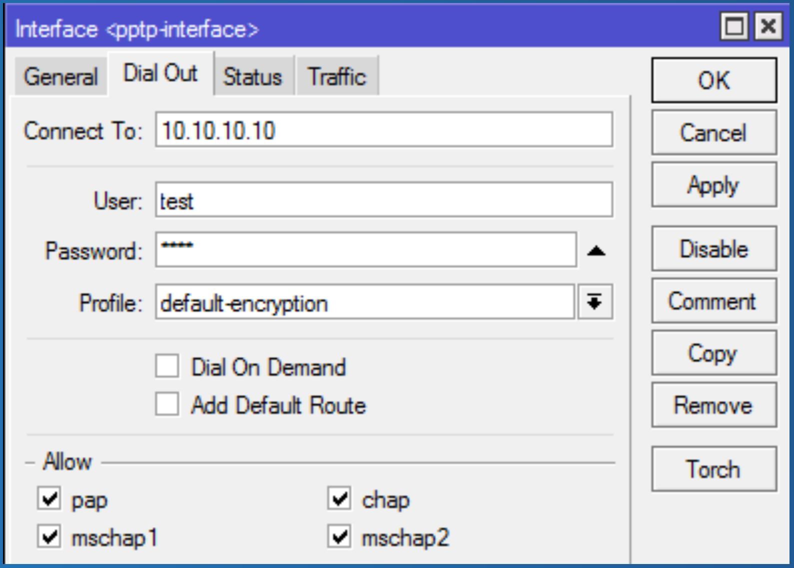
admin@192.168.56.2 (The Game) - WinBox v5.11 on x86 (x86)

Uptime: 00:35:00 Memory: 45.1 MiB CPU: 1% ☑ Hide Passwords

Interface List

	Name	Type	L2 MTU	Tx	Rx	Tx Pac...	Rx Pac...	Tx Drops	Rx Drops
R	bridge-interface	Bridge	65535	368 bps	0 bps	1	0	0	0
R	ether1	Ethernet		368 bps	0 bps	1	0	0	0
R	ether2	Ethernet		0 bps	0 bps	0	0	0	0
R	ether3	Ethernet		20.3 kbps	2.9 kbps	4	4	0	0
	;;; Help I've lost my VPN mummy!								
	pptp-interface	PPTP Client		0 bps	0 bps	0	0	0	0

5 items



Interface <pptp-interface>

Connect To: 10.10.10.10

User: test

Password: ****

Profile: default-encryption

☐ Dial On Demand

☐ Add Default Route

Allow

☑ pap ☑ chap

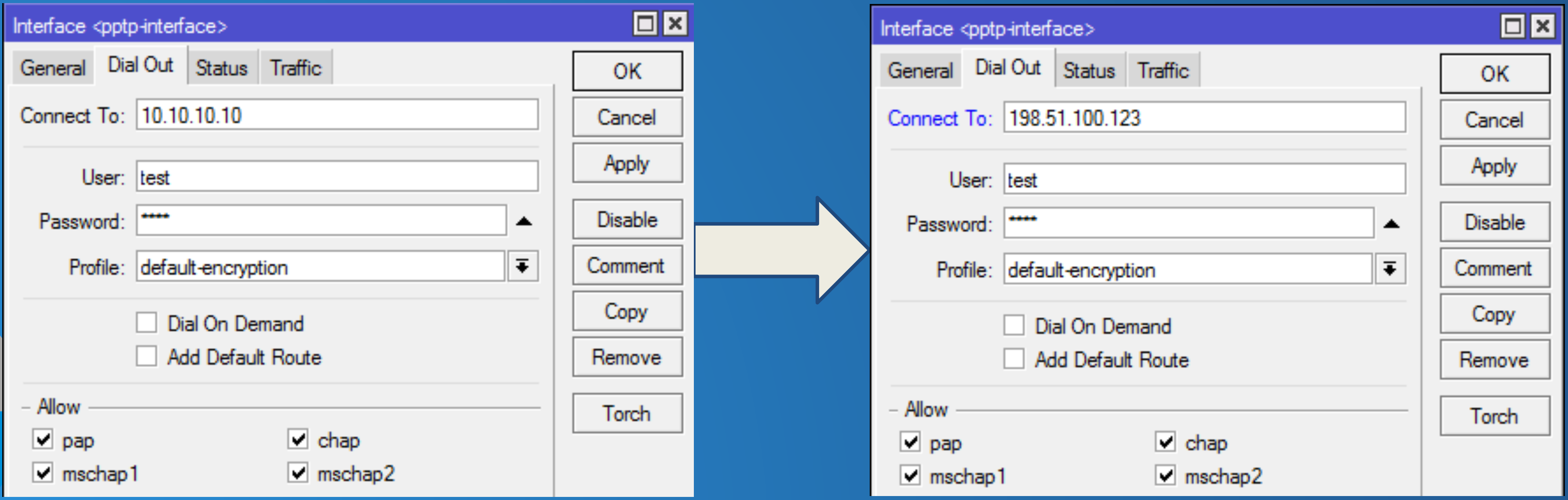
☑ mschap1 ☑ mschap2

...

Advanced Scripting Example: breakdown 5/5

Compare the current and the new address.
If they don't match, the interface address needs to be updated to connect to the new server.

```
:if ($currentvpnip != $newvpnip) do={/interface pptp-client
set [find name=$vpninterface] connect-to=$newvpnip}
```



Interface <pptp-interface>

Connect To: 10.10.10.10
User: test
Password: ****
Profile: default-encryption
☐ Dial On Demand
☐ Add Default Route
Allow
☑ pap ☑ chap
☑ mschap1 ☑ mschap2

Interface <pptp-interface>

Connect To: 198.51.100.123
User: test
Password: ****
Profile: default-encryption
☐ Dial On Demand
☐ Add Default Route
Allow
☑ pap ☑ chap
☑ mschap1 ☑ mschap2

How does this apply in the real world?

- automated backups
  - router configuration
  - router bandwidth graphs
- automated user management
  - billing
  - speed changes
  - user-manager modifications
- semi-automated configuration setup
- on the fly bandwidth/queue management
- feature additions
- automated scanning (wireless, lan, etc)

Real-world example:
Data limits on hotspot trial users

This feature does not exist in the standard hotspot trial user options!

Scheduled to run every 5m:

```
:local counter
:local datadown
:local username
:local macaddress
:foreach counter in=[/ip hotspot active find ] do={
:set datadown [/ip hotspot active get $counter bytes-out]
:if ($datadown>50000000) do={
:set username [/ip hotspot active get $counter user]
:set macaddress [/ip hotspot active get $counter mac-address]
/ip hotspot user remove [/ip hotspot user find where name=$username]
/ip hotspot user add name=$username limit-bytes-out=50000000 mac-address=$macaddress
/ip hotspot active remove $counter
:log info "Logged out $username - Reached 50MB download quota"
}}
```

Scheduled to run every 24 hours:

```
:foreach counter in=[/ip hotspot user find ] do={/ip hotspot user remove \$counter}
```

Questions?

Links and Such

My Blog: http://www.mikrotik-routeros.com
Podcast: http://www.thebrotherswisp.com
Email: admin@mikrotik-routeros.com

My MikroTik Forum username: omega-00

Other awesome networking blogs to check out:
http://www.gregsowell.com
http://www.3dbwireless.com/boyd/
http://www.mtin.net/blog/

The Original and best MikroTik Manual:
http://wiki.mikrotik.com