Send a list of connected users to e-mail on Mikrotik

If you need a semi-accurate (more often you run it, more accurate it will be) list of active users on your Mikrotik (including wireless and VPN connections), here is a script for you.

Set your clock correctly and then create a script:

```
:local ActiveAdmins ;
:local int ;
:local ActivePPTP ;
:local ActiveL2TP ;
:local ActivePPP ;
:local ActivePPPoE ;
:local ActiveOVPN ;
:local ActiveSSTP ;
:local VPNs ;
:local WirelessClients ;
:local HotSpotUsers ;
:global oldlist ;
:global newlist ;
:global ActiveServers ;
:global InactiveServers ;

:set ActiveServers "* Active server list *" ;
:set InactiveServers "\n * Inactive server list *" ;
if ( [ping 192.168.1.2 count=1] = 1 ) do={:set ActiveServers "$ActiveServers
\n Accesspoint"} else {:set InactiveServers "$InactiveServers \n
Accesspoint"}
:set ActiveAdmins "$ActiveAdmins \n ** Active admins list ** \n" ;
;:foreach int in=[/user active find] do={
; :set ActiveAdmins "$ActiveAdmins $[/user active get $int via]: $[/user
active get $int name]@$[/user active get $int address]\n" ;
;}

:set ActivePPPoE "* PPPoE Servers * \n" ;
:foreach int in=[/interface pppoe-server find] do={
:if ( [/interface pppoe-server get $int running] = true ) do={
:set ActivePPPoE "$ActivePPPoE $[/interface pppoe-server get $int name] :
$[/interface pppoe-server get $int user]@$[/interface pppoe-server get $int
remote-address] \n" ;
}
}
:set ActivePPPoE "$ActivePPPoE \n * PPPoE Clients * \n" ;
:foreach int in=[/interface pppoe-client find] do={
:if ( [/interface pppoe-client get $int running] = true ) do={
:set ActivePPPoE "$ActivePPPoE $[/interface pppoe-client get $int name] :
$[/interface pppoe-client get $int user] $[/interface pppoe-client get $int
```

```
password] \n" ;
}
}

:set ActivePPP "* PPP Servers * \n" ;
:foreach int in=[/interface ppp-server find] do={
:if ( [/interface ppp-server get $int running] = true ) do={
:set ActivePPP "$ActivePPP $[/interface ppp-server get $int name] :
$[/interface ppp-server get $int port] \n" ;
}
}
:set ActivePPP "$ActivePPP \n * PPP Clients * \n" ;
:foreach int in=[/interface ppp-client find] do={
:if ( [/interface ppp-client get $int running] = true ) do={
:set ActivePPP "$ActivePPP $[/interface ppp-client get $int name] :
$[/interface ppp-client get $int port] $[/interface ppp-client get $int apn]
$[/interface ppp-client get $int user] $[/interface ppp-client get $int
password] \n" ;
}
}

:set ActiveOVPN "* OVPN Servers * \n" ;
:foreach int in=[/interface ovpn-server find] do={
:if ( [/interface ovpn-server get $int running] = true ) do={
:set ActiveOVPN "$ActiveOVPN $[/interface ovpn-server get $int name] :
$[/interface ovpn-server get $int user]@$[/interface ovpn-server get $int
client-address]\n" ;
}
}
:set ActiveOVPN "$ActiveOVPN \n * OVPN Clients * \n" ;
:foreach int in=[/interface ovpn-client find] do={
:if ( [/interface ovpn-client get $int running] = true ) do={
:set ActiveOVPN "$ActiveOVPN $[/interface ovpn-client get $int name] :
$[/interface ovpn-client get $int user]@$[/interface ovpn-client get $int
connect-to] $[/interface ovpn-client get $int password] \n" ;
}
}

:set ActiveSSTP "* SSTP Servers * \n" ;
:foreach int in=[/interface sstp-server find] do={
:if ( [/interface sstp-server get $int running] = true ) do={
:set ActiveSSTP "$ActiveSSTP $[/interface sstp-server get $int name] :
$[/interface sstp-server get $int user]@$[/interface sstp-server get $int
client-address] \n" ;
}
}
:set ActiveSSTP "$ActiveSSTP \n * SSTP Clients * \n" ;
:foreach int in=[/interface sstp-client find] do={
:if ( [/interface sstp-client get $int running] = true ) do={
```

```
:set ActiveSSTP "$ActiveSSTP $[/interface sstp-client get $int name] :
$[/interface sstp-client get $int user]@$[/interface sstp-client get $int
connect-to] $[/interface sstp-client get $int password] \n" ;
}
}

:set ActivePPTP "* PPTP Servers * \n" ;
:foreach int in=[/interface pptp-server find] do={
:if ( [/interface pptp-server get $int running] = true ) do={
:set ActivePPTP "$ActivePPTP $[/interface pptp-server get $int name] :
$[/interface pptp-server get $int user]@$[/interface pptp-server get $int
client-address] \n" ;
}
}
:set ActivePPTP "$ActivePPTP \n * PPTP Clients * \n" ;
:foreach int in=[/interface pptp-client find] do={
:if ( [/interface pptp-client get $int running] = true ) do={
:set ActivePPTP "$ActivePPTP $[/interface pptp-client get $int name] :
$[/interface pptp-client get $int user]@$[/interface pptp-client get $int
connect-to] $[/interface pptp-client get $int password] \n" ;
}
}

:set ActiveL2TP "* L2TP Servers * \n" ;
:foreach int in=[/interface l2tp-server find] do={
:if ( [/interface l2tp-server get $int running] = true ) do={
:set ActiveL2TP "$ActiveL2TP $[/interface l2tp-server get $int name] :
$[/interface l2tp-server get $int user]@$[/interface l2tp-server get $int
client-address] \n" ;
}
}
:set ActiveL2TP "$ActiveL2TP \n * L2TP Clients * \n" ;
:foreach int in=[/interface l2tp-client find] do={
:if ( [/interface l2tp-client get $int running] = true ) do={
:set ActiveL2TP "$ActiveL2TP $[/interface l2tp-client get $int name] :
$[/interface l2tp-client get $int user]@$[/interface l2tp-client get $int
connect-to] $[/interface l2tp-client get $int password] \n" ;
}
}

:local VPNs "** VPN list ** \n $ActiveL2TP \n $ActivePPTP \n $ActivePPP \n
$ActivePPPoE \n $ActiveOVPN \n $ActiveSSTP"


#:local WirelessClients "* Wireless clients * \n";
#:foreach int in=[/interface wireless registration-table find] do={
# :set WirelessClients "$WirelessClients $[/interface wireless registration-
table get $int mac-address] on $[/interface wireless registration-table get
$int interface] $[/interface wireless registration-table get $int signal-
strength] \n" ;
#}
```

```
#:local HotSpotUsers "* Hotspot users * \n" ;
#:foreach int in=[/ip hotspot active find] do={
# :set WirelessClients "$HotSpotUsers $[/ip hotspot active get $int
user]@$[/ip hotspot active get $int address] on $[/ip hotspot active get
$int server]\n" ;
#}

:global newlist "$ActiveAdmins \n \n $VPNs \n \n ** Wireless list ** \n
$WirelessClients \n $HotSpotUsers \n \n ** Server status ** \n
$ActiveServers \n $InactiveServers"

:local body "$[/system identity get name] report on: $[/system clock get
date]@$[/system clock get time]\n $newlist"

:if ( $oldlist != $newlist) do={
/tool e-mail send body="$body" subject="Active" to="<your email here>"
from="$[/system identity get name]@example.com"
:set oldlist $newlist
:log error "Active list sent"
} else={
:log error "Active no change"
}
```

Do not forget to enter your own e-mail address

Now, create a scheduler which will run this script in a designated interval. Keep in mind that if you run the script more often, you will get more accurate results, but it will have more impact on the router's CPU. So with that in mind, comment out everything you don't need (eg. if you don't have wireless or don't use Open VPN)

From:
https://wiki.plecko.hr/ - **Eureka Moment**

Permanent link:
https://wiki.plecko.hr/doku.php?id=mikrotik:scripting:connected_users_list_to_email

Last update: **2019/10/31 09:05**