ДОДАТОК А. СКРИПТ АВТОМАТИЗАЦІЇ НАЛАШТУВАННЯ БЕЗПЕКИ КОМП'ЮТЕРНОЇ МЕРЕЖІ НА ОСНОВІ ОБЛАДНАННЯ MIKROTIK

```
#Installation script variables
#General settings
:local localSubnet "10.0.0";
:local SystemIdentity "RealMikrotik_GW";
:local AdminUser "newadmin";
:local AdminPass "adminpass";
:local AllowIPRemoteManagement "allowip.company.com";
:local CommentPref "DefConf"
:local InetSpeed "max-net-speed"
#WAN
	#WAN IP type (static or dynamic)
	:local WANConnect "dynamic";
	#Static IP
	:local WANIP "1.1.1.2";
	:local WANIPprefix "29";
	:local WANGW "1.1.1.1";
	:local WANDNS "8.8.8.8,8.8.4.4";
#Queues
	#QoS customize? (1 yes, 0 no)
	:local QueuesInstall 1;
	#Internet access rate for queues. Specify in bytes
	:local InetSpeed "50000000";
#Backup
	#Backup to enail service customize? (1 yes, 0 no)
	:local BackupSend 1;
	#SMTP settings. SMTP-TLS = yes, no, tls-only.
	:local SMTPServer "smtp.gmail.com";
	:local SMTPPort "465";
	:local SMTPUser "mikrotik@company.com";
	:local SMTPPass "mailpass";
	:local SMTPTLS "tls-only";
	:local SMTPFrom "Mikrotik Backup";
```

```
        :local BackupToEmail "it@company.com";
#NTP
        #NTP client customize? (1 yes, 0 no)
        :local NTPUpdate 1;
        #NTP settings. DNS name
        :local ntpsrv1 "0.ua.pool.ntp.org";
        :local ntpsrv2 "1.ua.pool.ntp.org";
#VPN
        #L2TP VPN service customize? (1 yes, 0 no)
        :local VPNInstall 1;
        #L2TP VPN settings
        :local VPNPoolSubnet "10.10.11";
        :local VPNPSK "hyvZmRoFoXBzXcBqhdh6hdP66S7LKbaw";
#CAPsMAN
        #CAPsMAN service customize? (1 yes, 0 no)
                :local CAPsMANInstall 1;
        #CAPsMAN settings
                :local SSIDOffice "OfficeNet";
                :local PassOffice "wifiofficepass";
        #CAPsMAN guest service customize? (1 yes, 0 no)
                :local CAPsMANGuestNetInstall 1;
        #CAPsMAN guest settings
                :local SSIDGuest "GuestNet";
                :local PassGuest "wifiguestpass";
                :local GuestSubnet "192.168.12";
#Wait for interfaces for CAPsMAN
:log info "Wait for interfaces";
:local count 0;
:while ([/interface ethernet find] = "") do={
        :if ($count > 15) do={
        :log warning "DefConf: Unable to find ethernet interfaces";
        /quit;    }
        :delay 1s; :set count ($count +1);};
:local count 0;
:while ([/interface wireless print count-only] < 1) do={
```

```
        :set count ($count +1);
        :if ($count > 20) do={
        :log warning "DefConf: Unable to find wireless interface(s)";
        /quit}
        :delay 1s;};
:log info "Wait for interfaces end";
:log info "Starting_$CommentPref_script";
:log info "Start ether1 marked";
:do {/interface set ether1 comment="$CommentPref: WAN_ISP1 /Control"} on-error={:log error "Not
comment for ether1"};
# Create a bridge and add interfaces to it
:log info "Start Bridge configured";
:do {
        /interface bridge {
        add name=bridge1 priority=0x1000 comment="$CommentPref: localnet bridge"
        :log info "Bridge created";
        :log info "Start Admin MAC installed";
        :local adminmac;
        :local ether1mac "$[/interface ethernet get number=0 mac-address]";
        :if ([:pick $ether1mac 16 17]=0) do={
                :if ([:pick $ether1mac 15 16]~"[A-F]") do={
                        :set adminmac "$[:pick $ether1mac 0 15]9";
                } else={
                :set adminmac "$[:pick $ether1mac 0 15]$([:tonum [:pick $ether1mac 15 16]] - 1)";}
                :set adminmac ("$adminmac"."F");
        } else {:if ([:pick $ether1mac 16 17]~"[A-F]") do={
                        :set adminmac "$[:pick $ether1mac 0 16]9";
        } else={:set adminmac "$[:pick $ether1mac 0 16]$([:tonum [:pick $ether1mac 16 17]] - 1)";
                }}
                set bridge1 auto-mac=no admin-mac=$adminmac;
                :log info "Admin MAC installed";
        };
        :log info "Start Ports added to Bridge";
        :foreach k in=[/interface find where !(slave=yes  || name="ether1" || name~"bridge")] do={
        :local tmpPortName [/interface get $k name];
```

```
        :log info "port: $tmpPortName";
        /interface bridge port add bridge=bridge1 interface=$tmpPortName
comment="$CommentPref";};
        :log info "End Ports added to Bridge";
} on-error={:log error "Error Bridge configured"};
#Create Interface List, add interfaces there
:log info "Start Interface list configured";
:do {   /interface list add name=WAN comment="$CommentPref";
        /interface list add name=LAN comment="$CommentPref";
        /interface list member {
                add interface=bridge1 list=LAN
                add interface=ether1 list=WAN};
        :log info "Interface list created";
} on-error={:log error "Error Interface list configured"};
#Assign a local IP and configure a DHCP server
:log info "Start Local and WAN settings configured";
:do {/ip address add address="$localSubnet.1/24" interface=bridge1 comment="$CommentPref";
:log info "Local IP installed";
/ip pool add name="default-dhcp" ranges="$localSubnet.20-$localSubnet.254"
comment="$CommentPref";
/ip dhcp-server add name=Real_DefConf address-pool="default-dhcp" interface=bridge1 lease-
time=72h disabled=no;
/ip dhcp-server network add address="$localSubnet.0/24" gateway="$localSubnet.1"
comment="$CommentPref";
:log info "DHCP Server installed";
:log info "Start WAN IP installed";
:if ($WANConnect != "static" and $WANConnect != "dynamic") do {
:log error message="Error WAN connections type. WAN IP not installed";
} else { :if ($WANConnect = "static") do {
        :do { :log info "Start static WAN IP installed";
        /ip address add address="$WANIP/$WANIPprefix" interface=ether1 comment=
"$CommentPref: WAN ISP1 IP1";
        /ip firewall address-list add list="WAN_ISP1_IP1" address="$WANIP" comment=
"$CommentPref: WAN IP1 on ether1";
        /ip route add dst-address=0.0.0.0/0 gateway="$WANGW";
```

```
/ip dns set servers="$WANDNS";
:log info "Static WAN IP installed";
} on-error={:log error "Error Static WAN IP installed"};
} else { :do {
:log info "Start DHCP WAN IP installed";
/ip dhcp-client add interface=ether1 disabled=no comment="$CommentPref" script=":local count [/ip firewall address-list print count-only where list~\"WAN_ISP1_IP1\"]\r\
\n:if (\$bound=1) do={\r\
\n :if (\$count = 0) do={\r\
\n /ip firewall address-list add list=\"WAN_ISP1_IP1\" address=\$\"lease-address\" comment=\"RealDefConf: WAN IP from DHCP clinet on ether1\"\r\
\n } else={\r\
\n :if (\$count = 1) do={\r\
\n :local test [/ip firewall address-list find where comment=\"RealDefConf: WAN IP from DHCP clinet on ether1\"]\r\
\n :if ([/ip firewall address-list get \$test address] != \$\"lease-address\") do={\r\
\n /ip firewall address-list set \$test address=\$\"lease-address\"\r\
\n }\r\
\n } else={\r\
\n :error \"Multiple address found\"\r\
\n }\r\
\n }\r\
\n } else={\r\
\n /ip firewall address-list remove [find where comment=\"RealDefConf: WAN IP from DHCP clinet on ether1\"]\r\
\n};"
:log info "DHCP WAN IP installed";
} on-error={:log error "Error DHCP WAN IP installed"};
};};} on-error={:log error "Error Local or WAN IP configured"};
#Created Address Lists
:log info "Start Address list configured";
:do {/ip firewall address-list {
add address=0.0.0.0/8 list=BOGONS
add address=10.0.0.0/8 list=BOGONS
add address=100.64.0.0/10 list=BOGONS
```

```
        add address=127.0.0.0/8 list=BOGONS
        add address=169.254.0.0/16 list=BOGONS
        add address=172.16.0.0/12 list=BOGONS
        add address=192.0.0.0/24 list=BOGONS
        add address=192.0.2.0/24 list=BOGONS
        add address=192.168.0.0/16 list=BOGONS
        add address=198.18.0.0/15 list=BOGONS
        add address=198.51.100.0/24 list=BOGONS
        add address=203.0.113.0/24 list=BOGONS
        add address="$localSubnet.0/24" list=LocalNet
        :log info "Address list created";
} on-error={:log error "Error Address lists configured"};
#Configure FireWall
:log info "Start Firewall configured";
:do { :do { :log info "Start Source NAT created";
        /ip firewall nat{
        :if ($WANConnect != "static" and $WANConnect != "dynamic") do {
        :log error message="Error WAN connections type. Source NAT not created";
        } else {:if ($WANConnect = "static") do {
                add chain=srcnat action=src-nat to-addresses="$WANIP" out-interface-list=WAN ipsec-policy=out,none comment="$CommentPref: masquerade over WAN";
                } else {:if ($WANConnect = "dynamic") do {
                        add chain=srcnat action=masquerade out-interface-list=WAN ipsec-policy=out,none comment="$CommentPref: masquerade over WAN";
        };};};};
                :log info "Source NAT created";
        } on-error={:log error "Error Source NAT created"};
        :do {
                :log info "Start NAT Loopback created";
                /ip firewall nat add chain=srcnat action=masquerade out-interface-list=LAN src-address-list=LocalNet comment="$CommentPref: NAT loopback masquerade for LAN";
                :log info "NAT Loopback created";
        } on-error={:log error "Error NAT Loopback created"};
        :do {   :log info "Start Firewall Filter created";
        /ip firewall filter{
```

add chain=input action=accept dst-port=8291,22 in-interface-list=WAN protocol=tcp src-address-list=AllowIPRemoteManagement comment="$CommentPref: Allow remote managment from IP"

add chain=input action=add-src-to-address-list in-interface-list=WAN src-address-list="!NotTrapsIP" protocol=tcp psd=10,10s,3,1 address-list=TrapAddress address-list-timeout=7d comment="$CommentPref: Trap for port scanning"

add chain=input action=add-src-to-address-list in-interface-list=WAN src-address-list="!NotTrapsIP" protocol=tcp dst-port=5060,5061,4569,3389,8291,22,23,389,445 connection-nat-state=!dstnat address-list=TrapAddress address-list-timeout=3d comment="$CommentPref: Trap for TCP traffic"

add chain=input action=add-src-to-address-list in-interface-list=WAN src-address-list="!NotTrapsIP" protocol=tcp dst-port=5060,4569,53,161 connection-nat-state=!dstnat address-list=TrapAddress address-list-timeout=3d comment="$CommentPref: Trap for UDP traffic"

add chain=forward action=add-src-to-address-list address-list=DoS_Atack_Address address-list-timeout=3d connection-limit=20,32 connection-nat-state=dstnat in-interface-list=WAN src-address-list=!NotTrapsIP comment="$CommentPref: DoS atack detected from single IP"

add chain=forward action=add-src-to-address-list address-list=DoS_Atack_Address address-list-timeout=3d connection-limit=100,24 connection-nat-state=dstnat in-interface-list=WAN src-address-list=!NotTrapsIP comment="$CommentPref: DoS atack detected from 24 subnet"

add chain=input action=accept connection-state=established,related,untracked comment="$CommentPref: accept established,related,untracked"

add chain=input action=drop connection-state=invalid comment="$CommentPref: drop invalid"

add chain=input action=drop protocol=icmp icmp-options=8:0 in-interface-list=WAN src-address-list="!AllowIPRemoteManagement" comment="$CommentPref: Drop IN echo request"

add chain=input action=accept protocol=icmp comment="$CommentPref: accept ICMP"

add chain=input action=drop in-interface-list=!LAN comment="$CommentPref: drop all not coming from LAN"

add action=accept chain=forward comment="$CommentPref: accept in ipsec policy" ipsec-policy=in,ipsec

add action=accept chain=forward comment="$CommentPref: accept out ipsec policy" ipsec-policy=out,ipsec

add chain=forward action=accept connection-state=established,related,untracked comment="$CommentPref: accept established,related, untracked"

add chain=forward action=drop connection-state=invalid comment="$CommentPref: drop invalid"

add chain=forward action=drop connection-state=new dst-address-list=BOGONS out-interface-list=WAN log=yes log-prefix="BOGONS over WAN" comment="$CommentPref: Reject BOGONS routing over WAN"

add chain=forward action=drop connection-state=new protocol=tcp dst-port=25,587,465 out-interface-list=WAN dst-address-list=!SMTP_External_Servers src-address-list=!SMTP_Internal_Servers/Clients/Clients log=yes log-prefix="SMTP Spam" comment="$CommentPref: Drop out SMTP not allow hosts"

add chain=forward action=drop connection-state=new protocol=tcp dst-port=445 out-interface-list=WAN log=yes log-prefix="SMB Scan" comment="$CommentPref: Drop out SMB not allow hosts"

add chain=forward action=drop connection-state=new connection-nat-state=!dstnat in-interface-list=WAN comment="$CommentPref: drop all from WAN not DSTNATed" };

/ip firewall raw add action=accept chain=prerouting dst-port=8291,22 in-interface-list=WAN protocol=tcp src-address-list=AllowIPRemoteManagement comment="$CommentPref: Allow remote managment from IP"

/ip firewall raw add action=drop chain=prerouting comment="$CommentPref: Drop Address from Trap" src-address-list=TrapAddress;

/ip firewall raw add action=drop chain=prerouting comment="$CommentPref: Drop Address from DoS Atack" src-address-list=DoS_Atack_Address;

:log info "Firewall created";

} on-error={:log error "Error Firewall Filter created"};

} on-error={:log error "Error FireWall configured"};

#Setup of the elementary queues and marking for them

:log info "Start Queues configured";

:do {:if ($QueuesInstall = 1) do {

:do {

:log info "Start Mangle rules created";

/ip firewall mangle {

add action=mark-connection chain=prerouting connection-state=new dst-port=8291,22 new-connection-mark=ManTraff_conn passthrough=yes protocol=tcp comment="$CommentPref: Management traffic connections"

add action=mark-packet chain=prerouting connection-mark=ManTraff_conn new-packet-mark=ManTraff_Packets passthrough=no comment="$CommentPref: Management traffic packets"

add action=mark-connection chain=prerouting connection-state=new dst-address-list=SIP_External_Servers src-address-list=SIP_Internal_Servers/Clients new-connection-mark=SIP_Conn passthrough=yes comment="$CommentPref: SIP traffic connections"

add action=mark-connection chain=prerouting connection-state=new dst-address-list=SIP_Internal_Servers/Clients src-address-list=SIP_External_Servers new-connection-mark=SIP_Conn passthrough=yes comment="$CommentPref: SIP traffic connections"

add action=mark-packet chain=prerouting connection-mark=SIP_Conn new-packet-mark=SIP_Packets passthrough=no comment="$CommentPref: SIP traffic packets"

add action=mark-connection chain=prerouting connection-state=new dst-port=53 new-connection-mark=DNS_conn passthrough=yes protocol=tcp comment="$CommentPref: DNS traffic connections"

add action=mark-connection chain=prerouting connection-state=new dst-port=53 new-connection-mark=DNS_conn passthrough=yes protocol=udp comment="$CommentPref: DNS traffic connections"

add action=mark-packet chain=prerouting connection-mark=DNS_conn new-packet-mark=DNS_Packets passthrough=no comment="$CommentPref: DNS traffic packets"

add action=mark-connection chain=prerouting connection-state=new dst-port=80,443 new-connection-mark=HTTP_Conn passthrough=yes protocol=tcp comment="$CommentPref: Web traffic connections"

add action=mark-packet chain=prerouting connection-mark=HTTP_Conn new-packet-mark=HTTP_Packets passthrough=no comment="$CommentPref: Web traffic packets"

add action=mark-connection chain=prerouting connection-state=new dst-port=3389 new-connection-mark=RDP_Conn passthrough=yes protocol=tcp comment="$CommentPref: RDP traffic connections"

add action=mark-packet chain=prerouting connection-mark=RDP_Conn new-packet-mark=RDP_Packets passthrough=no comment="$CommentPref: RDP traffic packets"

add action=mark-connection chain=prerouting connection-state=new connection-mark=no-mark new-connection-mark=Other_traff_conn passthrough=yes comment="$CommentPref: Other traffic connections"

add action=mark-packet chain=prerouting connection-mark=Other_traff_conn new-packet-mark=Other_traff_packets passthrough=no comment="$CommentPref: Other traffic packets"};

:log info "Mangle rules created";

} on-error={:log error "Error Mangle rules created"};

```
	:do {	:log info "Start Queues created";
		/queue type add kind=pcq name=SIP pcq-classifier=src-address,dst-address,src-port,dst-port pcq-dst-address6-mask=128 pcq-rate=160k pcq-src-address6-mask=128 pcq-limit=10KiB;
	/queue simple {
		add dst=ether1 name=ISP1 target=bridge1 total-max-limit="$InetSpeed"
		add dst=ether1 name=SIP target=bridge1 packet-marks=SIP_Packets parent=ISP1 priority=1/1 total-queue=SIP total-max-limit="$InetSpeed"
		add dst=ether1 name=ManTraff target=bridge1 packet-marks=ManTraff_Packets parent=ISP1 priority=2/2 total-max-limit="$InetSpeed"
		add dst=ether1 name=DNS target=bridge1 packet-marks=DNS_Packets parent=ISP1 priority=3/3 total-max-limit="$InetSpeed"
		add dst=ether1 name=RDP target=bridge1 packet-marks=RDP_Packets parent=ISP1 priority=4/4 total-queue=pcq-download-default total-max-limit="$InetSpeed"
		add dst=ether1 name=HTTP target=bridge1 packet-marks=HTTP_Packets parent=ISP1 priority=6/6 total-queue=pcq-download-default total-max-limit="$InetSpeed"
		add dst=ether1 name=Other target=bridge1 packet-marks=Other_traff_packets parent=ISP1 priority=7/7 total-queue=pcq-download-default total-max-limit="$InetSpeed"};
	:log info "Queues created";
		} on-error={:log error "Error Queues created"};
	} else {:log warning "QueuesInstall != 1. Queues are not configured";};
} on-error={:log error "Error Queues configured"};
#Disable unused services
:log info "Start Services configured";
:do {
	/ip service {set telnet disabled=yes
		set ssh disabled=yes
		set ftp disabled=yes
		set www disabled=yes
		set api disabled=yes
		set api-ssl disabled=yes};
} on-error={:log error "Error services configured"};
#By default, disable the helper'y
:log info "Start Service ports configured";
:do {	/ip firewall service-port {set ftp disabled=yes
```

```
        set tftp disabled=yes
        set irc disabled=yes
        set h323 disabled=yes
        set sip disabled=yes
        set pptp disabled=yes
        set dccp disabled=yes
        set sctp disabled=yes};
} on-error={:log error "Error service ports configured"};
#Configuring backup
:log info "Start Backup tasks configured";
:do {
:if ($BackupSend = 1) do {
        #Configure mail
        :do {:log info "Start E-mail settings created";
        /tool e-mail set address="$SMTPServer" from="$SMTPFrom" password="$SMTPPass" port="$SMTPPort" start-tls="$SMTPTLS" user="$SMTPUser";
        :log info "E-mail settings created";
        } on-error={:log error "Error E-mail settings created"};
        #Create a backup script
        :do {
        :log info "Start Backup script created";
        /system script add name=Backup_to_email policy=read,write,policy,sensitive,test source="/system backup save encryption=aes-sha256 name=email_backup;\r\
        \n:delay 5;\r\
        \n/tool e-mail send file=\"email_backup.backup\" to=\"\$BackupToEmail\" from=\"\$SMTPUser\" body=\"See attached file\" subject=\"\$[/system identity get name] \$[/system clock get time] \$[/system clock get date] Backup\";\r\
        \n:delay 5;\r\
        \n/file remove [find name=\"email_backup.backup\"];"
        :log info "Backup script created";
        } on-error={:log error "Error Backup script created"};
        #Create a script launch schedule
        :do {:log info "Start Scheduler backup created";
```

```
/system scheduler add interval=1d name=Backup on-event="/system script run Backup_to_email" policy=read,write,policy,sensitive,test start-date=jan/01/1970 start-time=00:00:00 comment="$CommentPref: Creat and send to email config Backup";
:log info "Scheduler backup created";
} on-error={:log error "Error Scheduler backup created"};
} else {:log warning message="BackupSend != 1. Backup are not configured";};
} on-error={:log error "Error Backup configured"};
#Configuring NTP servers
:log info "Start NTP configured";
:do {:if ($NTPUpdate = 1) do {
:do {:log info "Start NTP update script created";
/system script add name=NTPServerUpdate policy=read,write,test source=("#Resolve the two ntp pool hostnames\r\
\n:local ntpipb [:resolve "."$ntpsrv1"."];\r\
\n:local ntpipa [:resolve "."$ntpsrv2"."];\r\\n\r\
\n# Get the current settings\r\
\n:local ntpcura [/system ntp client get primary-ntp];\r\
\n:local ntpcurb [/system ntp client get secondary-ntp];\r\ \n\r\
\n# Change if required\r\
\n:if (\$ntpipa != \$ntpcura) do={\r\
\n	:put \"Changing primary NTP\";\r\
\n	/system ntp client set primary-ntp=\"\$ntpipa\";\r\
\n	};\r\
\n:if (\$ntpipb != \$ntpcurb) do={\r\
\n	:put \"Changing secondary NTP\";\r\
\n	/system ntp client set secondary-ntp=\"\$ntpipb\";\r\
\n	};\r\
\n/system ntp client set enabled=yes;")
log info "NTP update script created";
on-error={:log error "Error NTP update script created"};
:do {	:log info "Start Scheduler NTP update created";
	/system scheduler add comment="$CommentPref: Check and set NTP servers" disabled=no interval=12h name=CheckNTPServers on-event="/system script run NTPServerUpdate" policy=read,write,test start-date=jan/01/1970 start-time=16:00:00;
	:log info "Scheduler NTP update created";
```

```
        } on-error={:log error "Error Scheduler NTP update created"};
        } else {:do {    :log info "Start NTP settings created";
                /system ntp client set primary-ntp="$ntpsrv1"
                /system ntp client set secondary-ntp="$ntpsrv2"
                /system ntp client set enabled=yes;
                :log info "NTP settings created";
                 on-error={:log error "Error NTP settings created"};}
} on-error={:log error "Error NTP update tasks configured"};
#Configuring L2TP VPN server
:log info "Start L2TP configured";
:do {:if ($VPNInstall = 1) do {
        :do {:log info "Start Preparation for L2TP";
        /interface list add name=VPN_L2TP_Users comment="$CommentPref";
        :log info "Interface list VPN_L2TP_Users created";
        /ip firewall filter{add chain=input action=accept protocol=udp dst-port=1701,500,4500 place-before=[find where comment="$CommentPref: drop all not coming from LAN"] comment="$CommentPref: Allow port for L2TP server"
        add chain=input action=accept protocol=ipsec-esp place-before=[find where comment="$CommentPref: drop all not coming from LAN"] comment="$CommentPref: Allow esp protocol for L2TP/Ipsec server"};
        :log info "Firewall created";
        /ip pool add name=VPN_Users ranges="$VPNPoolSubnet.0/24" comment="$CommentPref";
        :log info "IP pool created";
        /ppp profile add name=L2TP_Profiles local-address="$VPNPoolSubnet.1" remote-address=VPN_Users address-list=VPN_L2TP_Users interface-list=VPN_L2TP_Users change-tcp-mss=yes use-compression=no use-encryption=no only-one=yes;
        :log info "L2TP profiles created";
        } on-error={:log error "Error Preparation for L2TP"};
        :do {:log info "Start L2TP server created";
                /interface l2tp-server server set enabled=yes default-profile=L2TP_Profiles authentication=mschap2 use-ipsec=required ipsec-secret="$VPNPSK" caller-id-type=number;
                :log info "L2TP server created";
        } on-error={:log error "Error L2TP server created"};
        } else {:log warning message="VPNInstall != 1. L2TP VPN server are not configured";};
} on-error={:log error "Error L2TP configured"};
```

```
#Configuring CAPsMAN
:log info "Start CAPsMAN configured";
:do {:if ($CAPsMANInstall = 1) do {
	:log info "Start CAPsMAN settings created";
	:do {:log info "Start Channel settings created";
	/caps-man channel {
	add band=2ghz-b/g/n control-channel-width=20mhz extension-channel=disabled frequency=2412,2437,2462 name=2.4Channels reselect-interval=1d tx-power=24
	add band=5ghz-a/n/ac control-channel-width=20mhz extension-channel=Ce frequency=5180,5200,5220,5240,5260,5280,5300,5320 name=5Channels reselect-interval=1d tx-power=24 skip-dfs-channels=yes};
	:log info "Channel settings created";
	} on-error={:log error "Error Channel settings created"};
	:do {:log info "Start Security settings created";
	/caps-man security add authentication-types=wpa2-psk encryption=aes-ccm group-encryption=aes-ccm disable-pmkid=yes name=OfficeNetPass passphrase="$PassOffice";
	:log info "Security settings created";
	} on-error={:log error "Error Security settings created"};
	:do {:log info "Start Access-list settings created";
		/caps-man access-list {
		add action=accept allow-signal-out-of-range=5s disabled=no interface=any mac-address=00:00:00:00:00:00 signal-range=-85..0 ssid-regexp=""
		add action=reject allow-signal-out-of-range=always disabled=no interface=any mac-address=00:00:00:00:00:00 signal-range=-120..120 ssid-regexp=""};
	:log info "Access-list settings created";
	} on-error={:log error "Error Access-list settings created"};
	:do {:log info "Start Datapath settings created";
	/caps-man datapath add name=OfficeNet bridge=bridge1 client-to-client-forwarding=yes local-forwarding=yes interface-list=LAN;
	:log info "Datapath settings created";
	} on-error={:log error "Error Datapath settings created"};
	:do {:log info "Start Configuration settings created";
		/caps-man configuration {
```

```
        add channel=2.4Channels country=russia3 datapath=OfficeNet distance=indoors guard-interval=long max-sta-count=32 mode=ap multicast-helper=default name=OfficeNet2 rates=StandartDataRates rx-chains=0,1 security=OfficeNetPass ssid="$SSIDOffice-2.4Ghz" tx-chains=0,1
        add channel=5Channels country=russia3 datapath=OfficeNet distance=indoors guard-interval=long max-sta-count=32 mode=ap multicast-helper=default name=OfficeNet5 rates=StandartDataRates rx-chains=0,1 security=OfficeNetPass ssid="$SSIDOffice-5Ghz" tx-chains=0,1};
        :log info "Configuration settings created";
        } on-error={:log error "Error Configuration settings created"};
        :do {   :log info "Start Provisioning settings created";
                /caps-man provisioning {
                add action=create-disabled hw-supported-modes=gn master-configuration=OfficeNet2 name-format=prefix-identity name-prefix=2Ghz
                add action=create-disabled hw-supported-modes=ac master-configuration=OfficeNet5 name-format=prefix-identity name-prefix=5Ghz};
                :log info "Provisioning settings created";
        } on-error={:log error "Error Provisioning settings created"};
        /caps-man manager set enabled=yes;
        :log info "CAPsMAN enabled";
        :do {:log info "Start CAPsMAN guest settings created";
                :if ($CAPsMANGuestNetInstall = 1) do {
                :do {:log info "Start Guest bridge created";
                /interface bridge {
                add name=bridge10 disabled=no auto-mac=yes protocol-mode=rstp comment="$CommentPref";
                :local adminmac;
                :local bridge1mac "$[/interface ethernet get [..bridge find name=bridge1] mac-address]";
                :if ([:pick $bridge1mac 16 17]=0) do={
                :if ([:pick $bridge1mac 15 16]~"[A-F]") do={
                :set adminmac "$[:pick $bridge1mac 0 15]9";
                } else={:set adminmac "$[:pick $bridge1mac 0 15]$([:tonum [:pick $bridge1mac 15 16]] - 1)";
                                                        };
```

```
:set adminmac ("$adminmac"."F");
} else {:if ([:pick $bridge1mac 16 17]~"[A-F]") do={
:set adminmac "$[:pick $bridge1mac 0 16]9";
} else={:set adminmac "$[:pick $bridge1mac 0 16]$([:tonum [:pick $bridge1mac 16 17]] - 1)";};};
set bridge10 auto-mac=no admin-mac=$adminmac;};
/ip address add address="$GuestSubnet.1/24" interface=bridge10 comment="$CommentPref";
:log info "Guest bridge created";
} on-error={:log error "Error Guest bridge created"};
:do {:log info "Start Routing settings created";
/ip route rule add action=lookup-only-in-table interface=bridge10 table=WiFi_Guest;
/ip route add dst-address="$GuestSubnet.0/24" gateway=bridge10 routing-mark=WiFi_Guest;
:if ($WANConnect != "static" and $WANConnect != "dynamic") do {
:log error message="Error WAN connections type";
} else {:if ($WANConnect = "static") do {
/ip route {
add copy-from=[find connect=yes gateway=ether1] routing-mark=WiFi_Guest
add copy-from=[find dst-address=0.0.0.0/0] routing-mark=WiFi_Guest};
} else {:local script [/ip dhcp-client get value-name=script [/ip dhcp-client find where comment="$CommentPref"]];
/ip dhcp-client set [/ip dhcp-client find where comment="$CommentPref"] script="$script\r\
\n:local rmark \"WiFi_Guest\";\r\
\n:local WanNet [/ip address get value-name=network [/ip address find where interface=ether1 dynamic=yes]];\r\
\n:local count [/ip route print count-only where comment=\"WANGW\" routing-mark=\$rmark];\r\
\n:local countnet [/ip route print count-only where comment=\"WANNET\" routing-mark=\$rmark];\r\
\n:if (\$bound=1) do={\r\
\n :if (\$countnet = 0) do={\r\
\n /ip route add dst-address=\"\$WanNet\" gateway=ether1 comment=\"WANNET\" routing-mark=\$rmark;\r\
\n } else={\r\
\n :if (\$countnet = 1) do={\r\
\n :local test [/ip route find where comment=\"WANNET\" routing-mark=\$rmark];\r\
```

```
\n :if ([/ip route get \$test dst-address] != \"\$WanNet\") do={\r\
\n /ip route set \$test dst-address=\"\$WanNet\";\r\
\n }; }else={:error \"Multiple routes found\"; }; };\r\
\n :if (\$count = 0) do={/ip route add gateway=\$\"gateway-address\" comment=\"WANGW\" routing-mark=\$rmark;\r\
\n } else { :if (\$count = 1) do={\r\
\n :local test [/ip route find where comment=\"WANGW\" routing-mark=\$rmark];\r\
\n :if ([/ip route get \$test gateway] != \$\"gateway-address\") do={\r\
\n /ip route set \$test gateway=\$\"gateway-address\"; };\r\
\n } else={:error \"Multiple routes found\"; };}} else={\r\
\n /ip route remove [find where comment=\"WANGW\" routing-mark=\$rmark];\r\
\n /ip route remove [find where comment=\"WANNET\" routing-mark=\$rmark];\r\
\n};"};};
:log info "Routing settings created";
} on-error={:log error "Error Routing settings created"};
        :do {:log info "Start DHCP guest settings created";
        /ip pool add name="wifi-guest-dhcp" ranges="$GuestSubnet.20-$GuestSubnet.254" comment="$CommentPref";
        /ip dhcp-server add name=Real_DefConf_CAPsMAN address-pool="wifi-guest-dhcp" interface=bridge10 lease-time=3h disabled=no;
        /ip dhcp-server network add address="$GuestSubnet.0/24" gateway="$GuestSubnet.1" comment="$CommentPref";
        :log info "DHCP guest settings created";
        } on-error={:log error "Error DHCP guest settings created"};
        :do {:log info "Start Security guest settings created";
        /caps-man security add authentication-types=wpa2-psk encryption=aes-ccm group-encryption=aes-ccm disable-pmkid=yes name=GuestNetPass passphrase="$PassGuest";
        :log info "Security guest settings created";
        } on-error={:log error "Error Security guest settings created"};
        :do {:log info "Start Datapath guest settings created";
        /caps-man datapath add bridge=bridge10 client-to-client-forwarding=no local-forwarding=no name=GuestNet;
        :log info "Datapath guest settings created";
        } on-error={:log error "Error Datapath guest settings created"};
        :do {:log info "Start Configuration guest settings created";
```

```
/caps-man configuration {add channel=2.4Channels country=ukraine datapath=GuestNet distance=indoors guard-interval=long max-sta-count=32 mode=ap multicast-helper=default name=GuestNet2 rates=StandartDataRates rx-chains=0,1 security=GuestNetPass ssid="$SSIDOffice-Guest" tx-chains=0,1
add channel=5Channels country=ukraine datapath=GuestNet distance=indoors guard-interval=long max-sta-count=32 mode=ap multicast-helper=default name=GuestNet5 rates=StandartDataRates rx-chains=0,1 security=GuestNetPass ssid="$SSIDOffice-Guest" tx-chains=0,1};
:log info "Configuration guest settings created";
} on-error={:log error "Error Configuration guest settings created"};
:do { :log info "Start Provisioning guest settings created";
    /caps-man provisioning {
    set [find master-configuration=OfficeNet2] slave-configurations=GuestNet2
    set [find master-configuration=OfficeNet5] slave-configurations=GuestNet5};
        :log info "Provisioning guest settings created";
    } on-error={:log error "Error Provisioning guest settings created"};
:do { :log info "Start Queues guest settings created";
/ip firewall mangle {
add action=mark-connection chain=prerouting connection-state=new in-interface=bridge10 connection-mark=no-mark new-connection-mark=Guest_traff_conn passthrough=yes place-before=[find comment="$CommentPref: Other traffic connections"] comment="$CommentPref: Guest traffic connections"
add action=mark-packet chain=prerouting connection-mark=Guest_traff_conn new-packet-mark=Guest_traff_packets passthrough=no place-before=[find comment="$CommentPref: Other traffic connections"] comment="$CommentPref: Guest traffic packets"};
:local GuestInetSpeed ($InetSpeed / 2);
/queue simple {
add dst=ether1 name=Guest target=bridge10 packet-marks=Guest_traff_packets parent=ISP1 priority=8/8 total-queue=pcq-download-default total-max-limit="$GuestInetSpeed" place-before=[find name=Other];};
:log info "Queues guest settings created";
} on-error={:log error "Error Queues guest settings created"};
} else {:log warning message="CAPsMANGuestNetInstall != 1. CAPsMAN guest network are not configured"; };
} on-error={:log error "Error CAPsMAN guest settings created"};
```

```
        } else { :log warning message="CAPsMANInstall != 1. CAPsMAN are not configured";};
} on-error={:log error "Error CAPsMAN configured"};
#Configuring default settings
:log info "Start Standart settings configured";
:do {   /ip firewall connection tracking set tcp-established-timeout=1h;
        /tool mac-server ping set enabled=no;
        /ip neighbor discovery-settings set discover-interface-list=LAN;
        /tool mac-server set allowed-interface-list=LAN;
        /tool mac-server mac-winbox set allowed-interface-list=LAN;
        /ip dns set allow-remote-requests=yes;
        /system identity set name="$SystemIdentity";
        :log info "Standart settings created";
} on-error={:log error "Error Standart configured"};
#Configuring SNMP
:log info "Start SNMP settings configured";
:do { /snmp community set [find default=yes] name=$CommunityName security=private
authentication-password=$AuthPass authentication-protocol=SHA1 encryption-password=$EncrPass
encryption-protocol=AES;
        /snmp set enabled=yes trap-community=$CommunityName trap-version=3 engine-
id=[/interface ethernet get number=0 mac-address];
        :log info "SNMP settings created";
} on-error={:log error "Error SNMP settings configured"};
#Configuring Users
:log info "Start Users settings configured";
:do {   /user add name=$AdminUser password=$AdminPass group=full;
        /user remove admin;
        :log info "Users settings created";
} on-error={:log error "Error Users settings configured"};
:log info "$CommentPref_DefConf_script_finished";
```

№ док.	Позначення	Найменування	Дод. від.
		Текстові документи	
1	ГЮІК.ХХХХХХ.1648Ст.04ПЗ	Пояснювальна записка	109 с.
2		Графічні документи	
		Презентаційний матеріал	А4
3		Інші документи	
		Рецензія	1 с.
		Відгук керівника	1 с.
4		Електронна версія ПЗ	1 шт.

Ізм.	Аркуш	№ докум.	Підпис	Дата	ГЮІК.ХХХХХХ.1648Ст.04.ПЗ		
Розроб.		Гавриленко А.С.			Аудит інформаційної безпеки в комп'ютерних мережах на базі Mikrotik	Літ.	Аркуш 1 / Аркушів 1
Перевір.		Федюшин О.І.				Т	
							ХНУРЕ Кафедра БІТ
Н.контр.		Шевченко Л.П.					
Затверд.		Халімов Г.З.					