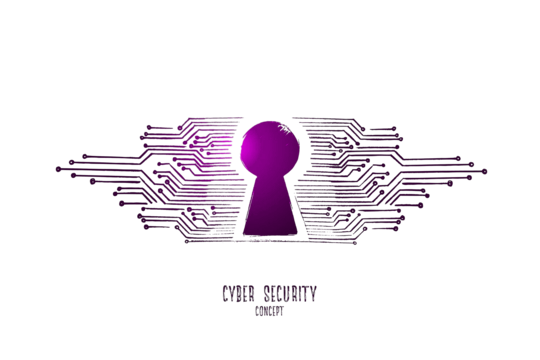
Bastionado de redes 

y sistemas.



**U2\_A1 Redes privadas en mikrotik**

**Francisco Iván Ramírez Ortega**

Curso 2023 - 2024

**IES El Rincón**

Curso de especialización en Ciberseguridad en entornos de las tecnologías de la información.

# Redes privadas en mikrotik

# Introducción

MikroTik es una empresa que desarrolla hardware y software para redes de datos. Su producto más conocido es **RouterOS**, un sistema operativo de enrutamiento que convierte un PC estándar en un enrutador dedicado con todas las funciones. Además de RouterOS, MikroTik también fabrica hardware específico para routers, como routers, switches y dispositivos inalámbricos.

RouterOS, el sistema operativo de MikroTik, es extremadamente versátil y se utiliza en una amplia gama de aplicaciones, incluyendo:

1. **Enrutamiento de red**: RouterOS se utiliza para enrutar el tráfico entre redes, ya sea en redes domésticas, empresariales o de proveedores de servicios de Internet (ISP).
2. **Firewall y seguridad**: RouterOS incluye características de firewall avanzadas que permiten a los administradores de red controlar y asegurar el tráfico que pasa a través del enrutador.
3. **Conectividad inalámbrica**: MikroTik fabrica una variedad de dispositivos inalámbricos, como puntos de acceso Wi-Fi y radios para enlaces punto a punto y punto a multipunto. Estos dispositivos son comúnmente utilizados para proporcionar conectividad inalámbrica en redes empresariales, ISP y proyectos de telecomunicaciones.
4. **Administración de ancho de banda**: RouterOS incluye características de administración de ancho de banda que permiten a los administradores de red controlar y limitar el ancho de banda utilizado por diferentes tipos de tráfico.
5. **Redes privadas virtuales (VPN)**: RouterOS es compatible con una variedad de protocolos VPN, lo que permite a los usuarios crear redes privadas virtuales seguras sobre Internet pública.
6. **Hotspot y gestión de acceso**: RouterOS incluye funciones de hotspot que permiten a los proveedores de servicios de Internet ofrecer acceso a Internet de pago o gratuito a través de redes Wi-Fi públicas.

**Winbox** es una aplicación de software con interfaz gráfica que se utiliza para administrar dispositivos MikroTik.

# Objetivo

1. Mostrar en el router mikrotik las conexiones activas.
2. Permitir que los servidores web de la red interna sean accesibles desde la red externa.

Guia… → [Link](https://docs.google.com/document/d/1oRZIazusI4bqfECP2nq9XXLejYLDS-FSRRWntKTNtXU/edit?usp=sharing)

1. Detectar en la tabla NAT las reglas de POSTROUTING
2. Interpretar resultados de: ip/firewall/nat/print stats
3. Copiar y comentar el fichero de configuración resultante

| Herramientas |
| --- |
| * [Winbox](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwirxe3u-aiEAxUT_rsIHVIECPcQFnoECAkQAQ&url=https%3A%2F%2Fmikrotik.com%2Fdownload&usg=AOvVaw2JCoTY2tqWuPue5n6W40Ts&opi=89978449) (snap install winbox) |
|  |

**ÍNDICE**

[**Redes privadas en mikrotik 2**](#_xfpg6pduf6p7)

[**Introducción 2**](#_b9oaw39qdr1r)

[**Objetivo 3**](#_3uoprqr5rx0x)

[**Herramientas 3**](#_wzp689uhopv7)

[Práctica 4](#_4sltpw5frw14)

[Conexiones activas. 4](#_4sltpw5frw14)

[Permitir que los servidores web de la red interna sean accesibles desde la red externa. 5](#_cl7k52817ivn)

[Servidor 5](#_qxexb09gr08l)

[Generar otra red 5](#_oypopuw5w3do)

[Reglas de POSTROUTING 7](#_i5tx5ina3wj6)

[Resultados de ip/firewall/nat/print stats 7](#_9jjjkhhqfxiy)

[Copiar y comentar el fichero de configuración resultante 8](#_zcik2cey2g98)

[**Otros 10**](#_26lq72cpqxok)

[Reseteo 10](#_4gh5jnk1ww01)

| Práctica |
| --- |
| Conexiones activas. |
| **/interface ethernet print** |
|  |
| Pertenece al bridge por defecto |
|  |
| Permitir que los servidores web de la red interna sean accesibles desde la red externa. |
| Guia → [Link](https://docs.google.com/document/d/1oRZIazusI4bqfECP2nq9XXLejYLDS-FSRRWntKTNtXU/edit?usp=sharing) |
| Servidor |
| A través del módulo [**http**](https://docs.python.org/3/library/http.html) de python podemos crear un servidor web. |
| **echo “**[**<texto html>**](https://www.w3schools.com/w3css/w3css_web_html.asp)**” > index.html**  **sudo python3 -m http.server 80** |
| Generar otra red |
| La idea es conectar dos redes y exponer un servicio. Asi que necesitamos crear otra red. |
| Creamos una nueva red para la interfaz conectada a la otra red. |
| **Ip → addresses → <añadimos la interfaz>** |
|  |
| Dentro del bridge (bridge1), cualquier red puede acceder al servidor web localizado en  **eth2 de la red 192.168.77.0/24**. |
|  |
|  |
|  |
| Reglas de POSTROUTING |
|  |
| Resultados de ip/firewall/nat/print stats |
|  |
| 1. chain: La cadena de reglas a la que pertenece la regla NAT. 2. action: La acción que se toma para el tráfico que coincide con la regla NAT (por ejemplo, dst-nat, src-nat, etc.). 3. packets: El número total de paquetes que han coincidido con la regla NAT. 4. bytes: El número total de bytes que han coincidido con la regla NAT. 5. src-address: Dirección IP de origen del tráfico afectado por la regla NAT. 6. dst-address: Dirección IP de destino del tráfico afectado por la regla NAT. 7. protocol: Protocolo de red utilizado por el tráfico afectado por la regla NAT (por ejemplo, TCP, UDP, etc.). 8. in-interface: Interfaz de entrada del tráfico afectado por la regla NAT. 9. out-interface: Interfaz de salida del tráfico afectado por la regla NAT. |
| Copiar y comentar el fichero de configuración resultante |
| Winbox en Linux funciona mediante un emulador Windows, Wine. Algunos datos como las fechas no las puede usar del sistema. |
|  |
|  |

| Otros |
| --- |
| Reseteo |
| 1. Desenchufar el dispositivo |
| 1. Mantener pulsado el botón RES (reseteo) |
| 1. Enchufar el aparato y esperar al **pitido.** |
| Interfaz gráfica |
| **System -> Reset Configuration** |
|  |
| Interfaz de comandos |
| **/system reset-configuration** |
| Configuración por defecto |
|  |
| #| Welcome to RouterOS!  #| 1) Set a strong router password in the System > Users menu  #| 2) Upgrade the software in the System > Packages menu  #| 3) Enable firewall on untrusted networks  #| -----------------------------------------------------------------------------  #| RouterMode:  #| \* WAN port is protected by firewall and enabled DHCP client  #| \* Ethernet interfaces (except WAN port/s) are part of LAN bridge  #| LAN Configuration:  #| IP address 192.168.88.1/24 is set on bridge (LAN port)  #| DHCP Server: enabled;  #| DNS: enabled;  #| WAN (gateway) Configuration:  #| gateway: ether1 ;  #| ip4 firewall: enabled;  #| NAT: enabled;  #| DHCP Client: enabled;  :global defconfMode;  :log info "Starting defconf script";  #-------------------------------------------------------------------------------  # Apply configuration.  # these commands are executed after installation or configuration reset  #-------------------------------------------------------------------------------  :if ($action = "apply") do={  # wait for interfaces  :local count 0;  :while ([/interface ethernet find] = "") do={  :if ($count = 30) do={  :log warning "DefConf: Unable to find ethernet interfaces";  /quit;  }  :delay 1s; :set count ($count +1);  };  /interface list add name=WAN comment="defconf"  /interface list add name=LAN comment="defconf"  /interface bridge  add name=bridge disabled=no auto-mac=yes protocol-mode=rstp comment=defconf;  :local bMACIsSet 0;  :foreach k in=[/interface find where !(slave=yes || name="ether1" || name~"bridge")] do={  :local tmpPortName [/interface get $k name];  :if ($bMACIsSet = 0) do={  :if ([/interface get $k type] = "ether") do={  /interface bridge set "bridge" auto-mac=no admin-mac=[/interface get $tmpPortName mac-address];  :set bMACIsSet 1;  }  }  :if (([/interface get $k type] != "ppp-out") && ([/interface get $k type] != "lte")) do={  /interface bridge port  add bridge=bridge interface=$tmpPortName comment=defconf;  }  }  /ip pool add name="default-dhcp" ranges=192.168.88.10-192.168.88.254;  /ip dhcp-server  add name=defconf address-pool="default-dhcp" interface=bridge lease-time=10m disabled=no;  /ip dhcp-server network  add address=192.168.88.0/24 gateway=192.168.88.1 comment="defconf";  /ip address add address=192.168.88.1/24 interface=bridge comment="defconf";  /ip dns {  set allow-remote-requests=yes  static add name=router.lan address=192.168.88.1 comment=defconf  }  /ip dhcp-client add interface=ether1 disabled=no comment="defconf";  /interface list member add list=LAN interface=bridge comment="defconf"  /interface list member add list=WAN interface=ether1 comment="defconf"  /ip firewall nat add chain=srcnat out-interface-list=WAN ipsec-policy=out,none action=masquerade comment="defconf: masquerade"  /ip firewall {  filter add chain=input action=accept connection-state=established,related,untracked comment="defconf: accept established,related,untracked"  filter add chain=input action=drop connection-state=invalid comment="defconf: drop invalid"  filter add chain=input action=accept protocol=icmp comment="defconf: accept ICMP"  filter add chain=input action=accept dst-address=127.0.0.1 comment="defconf: accept to local loopback (for CAPsMAN)"  filter add chain=input action=drop in-interface-list=!LAN comment="defconf: drop all not coming from LAN"  filter add chain=forward action=accept ipsec-policy=in,ipsec comment="defconf: accept in ipsec policy"  filter add chain=forward action=accept ipsec-policy=out,ipsec comment="defconf: accept out ipsec policy"  filter add chain=forward action=fasttrack-connection connection-state=established,related comment="defconf: fasttrack"  filter add chain=forward action=accept connection-state=established,related,untracked comment="defconf: accept established,related, untracked"  filter add chain=forward action=drop connection-state=invalid comment="defconf: drop invalid"  filter add chain=forward action=drop connection-state=new connection-nat-state=!dstnat in-interface-list=WAN comment="defconf: drop all from WAN not DSTNATed"  }  /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  }  #-------------------------------------------------------------------------------  # Revert configuration.  # these commands are executed if user requests to remove default configuration  #-------------------------------------------------------------------------------  :if ($action = "revert") do={  /user set admin password=""  /system routerboard mode-button set enabled=no  /system routerboard mode-button set on-event=""  /system script remove [find comment~"defconf"]  /ip firewall filter remove [find comment~"defconf"]  /ip firewall nat remove [find comment~"defconf"]  /interface list member remove [find comment~"defconf"]  /interface detect-internet set detect-interface-list=none  /interface detect-internet set lan-interface-list=none  /interface detect-internet set wan-interface-list=none  /interface detect-internet set internet-interface-list=none  /interface list remove [find comment~"defconf"]  /tool mac-server set allowed-interface-list=all  /tool mac-server mac-winbox set allowed-interface-list=all  /ip neighbor discovery-settings set discover-interface-list=!dynamic  :local o [/ip dhcp-server network find comment="defconf"]  :if ([:len $o] != 0) do={ /ip dhcp-server network remove $o }  :local o [/ip dhcp-server find name="defconf" !disabled]  :if ([:len $o] != 0) do={ /ip dhcp-server remove $o }  /ip pool {  :local o [find name="default-dhcp" ranges=192.168.88.10-192.168.88.254]  :if ([:len $o] != 0) do={ remove $o }  }  :local o [/ip dhcp-client find comment="defconf"]  :if ([:len $o] != 0) do={ /ip dhcp-client remove $o }  /ip dns {  set allow-remote-requests=no  :local o [static find comment="defconf"]  :if ([:len $o] != 0) do={ static remove $o }  }  /ip address {  :local o [find comment="defconf"]  :if ([:len $o] != 0) do={ remove $o }  }  :foreach iface in=[/interface ethernet find] do={  /interface ethernet set $iface name=[get $iface default-name]  }  /interface bridge port remove [find comment="defconf"]  /interface bridge remove [find comment="defconf"]  /interface bonding remove [find comment="defconf"]  /interface wireless cap set enabled=no interfaces="" caps-man-addresses=""  /caps-man manager set enabled=no  /caps-man manager interface remove [find comment="defconf"]  /caps-man manager interface set [ find default=yes ] forbid=no  /caps-man provisioning remove [find comment="defconf"]  /caps-man configuration remove [find comment="defconf"]  }  :log info Defconf\_script\_finished;  :set defconfMode; |

| Bibliografía |
| --- |
| C[omandos](https://www.mkesolutions.net/pdf/routeros-cheat-sheet-v1.1.pdf) |
|  |
|  |