Murray's Blog

About the things I make and do

Home Contact Donate About Search

FOLLOW:

NEWER
Reverse Proxy With IIS and Lets Encrypt

OLDER
Spring Planting 2016

RECENTS

CODING
BUILDING A CPRNG CALLED TERNINGER - PART 12 PORTING TO .NET STANDARD
2018-08-23

TECHNICAL
PORTING READABLEPASSPHRASE TO .NET CORE AND PUBLISHING NUGETS
2018-07-26

BIBLICAL
TITUS 2 - GODLY LIVES
2018-07-15

SPEAKING
A FEW PUBLIC SPEAKING TIPS
2018-06-30

CODING
BUILDING A CPRNG CALLED TERNINGER - PART 11 PRODUCTION USE
2018-05-28

TECHNICAL

Create Another WiFi Network on a Mikrotik Router

2016-10-04

Steps to create a second (or third) network on a Mikrotik router, perhaps for a guest network.

Background

I like to separate different WiFi on my network. So my friends use a guest network, the kids are on their own network, etc.

Many home routers have a “tick the box” style of guest WiFi network, you tick the “guest network” box and out pops a virtual guest access point.

Mikrotik routers can do exactly the same thing, except you can have effectively unlimited access points and you need to build the network piece by piece.

CATEGORIES

- Biblical (7)
- Coding (13)
- Garden (1)
- Legal (1)
- Meta (1)
- Speaking (1)
- Technical (36)

ARCHIVES

- August 2018 (1)
- July 2018 (2)
- June 2018 (1)
- May 2018 (1)
- April 2018 (2)
- March 2018 (2)
- February 2018 (3)
- January 2018 (3)
- December 2017 (2)
- November 2017 (1)
- October 2017 (1)
- September 2017 (2)
- August 2017 (1)
- July 2017 (3)

Steps

In this guide, I create a separate WiFi interface for phones and tablets.

Phones rarely need full network access to other local devices. And they have a nasty habit of getting lost, stolen or otherwise broken, so having a separate WiFi access point (and password) means you don’t accidentally disclose your main WiFi password.

I’ll be using WinBox, but I’ll also list the console details via a print command.

1. Create an Interface

First thing to do is create yourself a passphrase for your AP. I generate one from makemeapassword.org and save it in my KeePass database.

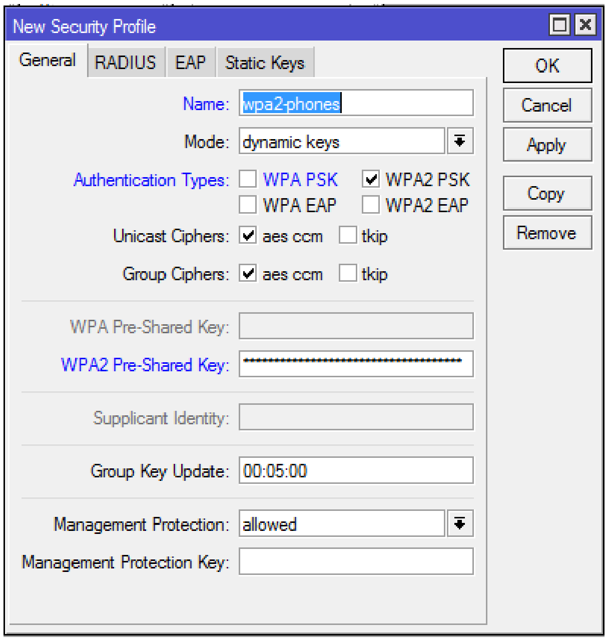
Then create a new *security profile*: Goto **Wireless** -> **Security Profiles** and add a new profile.

Give it an appropriate name (`wpa2-phones` in my case). I disable *WPA* and only use *WPA2*, as I have no legacy devices and it improves security slightly. Finally, don’t forget to enter your passphrase.

- June 2017 (2)
- May 2017 (3)
- April 2017 (2)
- February 2017 (1)
- January 2017 (2)
- December 2016 (1)
- November 2016 (1)
- October 2016 (1)
- September 2016 (2)
- August 2016 (1)
- July 2016 (3)
- June 2016 (2)
- April 2016 (5)
- March 2016 (2)
- February 2016 (2)
- January 2016 (5)

TAGS

- .NET (14)
- .NET Core (2)
- .NET Standard (2)
- 4G (2)
- About (1)
- Access Point (1)



Name: wpa2-phones

Mode: dynamic keys

Authentication Types: ☐ WPA PSK ☑ WPA2 PSK ☐ WPA EAP ☐ WPA2 EAP

Unicast Ciphers: ☑ aes ccm ☐ tkip

Group Ciphers: ☑ aes ccm ☐ tkip

WPA Pre-Shared Key:

WPA2 Pre-Shared Key: ************************************

Supplicant Identity:

Group Key Update: 00:05:00

Management Protection: allowed

Management Protection Key:

Wireless Security Profile

```
[admin@Mikrotik-gateway] /interfa

 4   name="wpa2-phones" mode=dyna
     group-ciphers=aes-ccm wpa-pre
     wpa2-pre-shared-key="NotMyRe
     eap-methods="" tls-mode=no-ce
     mschapv2-password="" static-a
     static-algo-2=none static-key
     static-transmit-key=key-0 sta
     radius-mac-authentication=no
     interim-update=0s radius-mac-
     radius-mac-caching=disabled g
     management-protection-key=""
```

Then create a virtual access point: Goto **Wireless** -> **Interfaces** and then add a *Virtual AP*.

On the *General* tab, enter a name for the network interface (which will be used internally on your Mikrotik). Mine is `wlan-phones`.

On the *Wireless* tab, enter an SSID to identify your network: `ligos-phones` for me, then select your newly created security profile

- Accountability Software (1)
- Agreement (1)
- Android (1)
- Authoring (1)
- Autoruns (1)
- Azure (1)
- Backup (3)
- Bandwidth (1)
- Bible (6)
- Bible Study (1)
- Bible Talk (6)
- BitBucket (1)
- Blog (4)
- Blue Screen (1)
- Boot-Failure (1)
- Bricked (1)
- Broken (1)
- Bsod (1)
- Bug Check (1)
- C# (12)
- CPRNG (12)
- CPU (1)
- Camera (3)
- Cellphone (1)

Wireless Virtual AP

Wireless Virtual AP

```
[admin@Mikrotik-gateway] /interfac

 3    name="wlan-phones" mtu=1500
      interface-type=virtual-AP ma
      vlan-id=1 wds-mode=disabled
      default-authentication=yes d
      default-client-tx-limit=0 h
```

2. Assign an IP Pool and Address

A network interface isn’t much use without an IP address.

Goto **IP** -> **Address** and then add a new address.

Choose an appropriate IP address for your new network (I’m using `10.46.2.xxx`) and assign it to your new interface.

Add an IP Address

Add an IP Address

```
[admin@Mikrotik-gateway] /ip addre
Flags: X - disabled, I - invalid,
 #   ADDRESS             NETWORK
 5   ;;; Phone WiFi
     10.46.2.1/24        10.46.2.0
```

On many other routers, you assign an IP address range against the DHCP server. On a Mikrotik you create an IP Pool, which is then used by DHCP (and other things too, I guess, though I have no idea what). So we need a pool before we can configure DHCP.

Goto **IP** -> **Pool** and then add a new pool.

I tend to reserve the bottom ~60 address (from `x.1` to `x.63`) for static allocations, and `x.255` is the broadcast address. Which means a range like `10.46.2.64 - 10.46.2.254` is my pool.

□The Dynamic Pool

The Dynamic Pool

```
[admin@Mikrotik-gateway] /ip pool>
 # NAME                           R/
 4 dhcp-phones                    1(
```

4. Create a DHCP Server

DHCP is used to assign addresses to devices as they connect to the WiFi network. They will use the pool we just created. And also assign a few other special addresses.

Goto **IP** -> **DHCP Server** -> *DHCP* Tab and add a new DCHP Server.

Give it a name (I named mine after the `wlan-phones` interface). Select the interface you created. Extend the lease time to something reasonably long (I use 1 day). And select the address pool you created in the last step.

□DHCP Server

DHCP Server

```
[admin@Mikrotik-gateway] /ip dhcp-
Flags: X - disabled, I - invalid
 #   NAME          INTERFACE
 4   wlan-phones   wlan-phones
```

- Certificate (1)
- Certify (1)
- Church (1)
- Cloud (2)
- Codeplex (1)
- Compatibility (1)
- Computer (1)
- Conceptual (1)
- Conditions (1)
- Configuration (1)
- Content (1)
- Costings (1)
- Covernant Eyes (1)
- Crash (3)
- Crashplan (2)
- Crypto (12)
- Cryptography (1)
- Custom Domain (1)
- Day Night Cycle (1)
- DeleteFacebook (1)
- Diagnose (1)
- Dice (12)
- Disassemble (1)
- Disaster-Recovery (1)

- Driver Verifier (1)
- Dropbox (2)
- Easter (3)
- Easter Friday (3)
- Email (1)
- File History (2)
- Firewall (2)
- Fortuna (12)
- Garden (1)
- Gateway (3)
- Godliness (1)
- Good Friday (3)
- Guest (2)
- HDD (1)
- HTTPS (1)
- Hard Disk (1)
- Hardware (3)
- Herbs (1)
- Hexo (4)
- Hexo-Bootstrap-Series (4)
- Home Router (3)
- Home Security (3)
- Honour (1)
- Hosting (1)

Now, jump over to the *Networks* tab and add new configuration.

The **Address** field is what ties the *Address Pool*, *DHCP Server* and *Network Configuration* all together. It should be the same as the IP address you chose, but with a zero at the end, and the netmask afterwards. `10.46.2.0/24` fits my example so far. The Netmask should be `255.255.255.0` or `24`, unless you know much more about subnets than I do.

I also set the router to be the DNS server and NTP server. And the domain to `ligos.local`.

DHCP Config

DHCP Config

```
[admin@Mikrotik-gateway] /ip dhcp-
 # ADDRESS          GATEWAY      [
 0 ;;; Phone WiFi
   10.46.2.0/24     10.46.2.1    1
```

5. Assign an IPv6 Pool and Address

I also have a public IPv6 range assigned by my ISP, so I add an IPv6 address as well. You need to create an IPv6 pool first, based on your public address assignment, before you can advertise it on an interface or assign an address. Also, because there's much more auto discovery built into IPv6, config is much less complicated.

Goto **IPv6** -> **Pool** and then add a new pool.

I'm simply assigning `/64` subnets (from my `/56` public allocation) to each network. This gives me 255 subnets for 255 networks (which is plenty, given I don't event have 255 devices!) There's no static assignments, so no address range exclusions like for IPv4.

The IPv6 Pool

The IPv6 Pool

```
[admin@Mikrotik-gateway] /ipv6 poo
Flags: D - dynamic
 #   NAME
 3   phones-ipv6-pool
```

Now, goto **IPv6** -> **Address**. You'll note that link local addresses (starting with `fe80`) have been dynamically created for your new interface. This is totally normal.

Now, add a new address. I use the same address for the router as for the pool. And set the correct pool and interface.

The IPv6 Address

The IPv6 Address

```
[admin@Mikrotik-gateway] /ipv6 add
Flags: X - disabled, I - invalid,
 #    ADDRESS
 8 DL fe80::4c5e:cff:feb8:d8d1/64
 9  G 2001:44b8:3168:9b03::/64
```

6. Add Firewall Rules

Before everything will work, you'll need a few firewall rules.

- How-To (4)
- Hyper-V (1)
- IIS (2)
- Idiot Poof (1)
- Installation (2)
- Internals (1)
- Internet (1)
- Intro (1)
- Isolated (3)
- Keepass (1)
- Keybase (1)
- LSI (1)
- LTE (2)
- Laptop (1)
- Layered Service Provider (1)
- Legal (1)
- Lets Encrypt (1)
- Malware (1)
- Messaging (1)
- Meta (1)
- Metadata (1)
- Microsoft Account (1)
- Migration (1)
- Mikrotik (13)

- Mobile Internet (2)
- Mobile Phone (1)
- Model N15W1 (1)
- Modem (1)
- Motherboard (1)
- MotionEye (2)
- NTP (1)
- Netboot (1)
- Netflow (1)
- Netinstall (1)
- Network (4)
- Networking (3)
- New User (1)
- Nfdump (1)
- Nmap (1)
- NuGet (1)
- Nuke it from Orbit (1)
- Outage (1)
- PGP (1)
- Password (1)
- Philippians (1)
- Planting (1)
- Pool (1)
- Porting (1)

I've created a defacto routing policy based on *Address Lists*. By adding the new network masks to existing *Address Lists*, everything just works without any further changes to firewall rules. Though I'll list the firewall rules as well, for your reference.

There are 4 categories I have at the moment:

1. **all_internal** - a list of all my internal networks. I need to add my new `10.46.2.0/24` network here.
2. **internal_trusted** - networks which may access LAN resources. As my new phones network doesn't need blanket local access, I don't add it.
3. **internal_restricted** - networks which cannot access LAN resources (unless I add explicit rules). I add `10.46.2.0/24` here.
4. **named_blah** - specific named devices. Because you can't use DNS names in firewall rules.

Note the `10.46.1.0/26` network in **internal_trusted**. Although `10.46.1.0/24` is restricted by default, I trust a small part of that network (this lets my kids' devices access printers and SMB shares).

```
[admin@Mikrotik-gateway] /ip firew
Flags: X - disabled, D - dynamic
 #   LIST                       A
30   all_internal               1
31   all_internal               1
32   all_internal               1
12   internal_trusted           1
14   internal_trusted           1
19   internal_restricted        1
20   internal_restricted        1
23   named_loki                 l
24   named_printer              p
```

- Presentation (2)
- Printer (1)
- Privacy (1)
- Property (1)
- Public Speaking (7)
- Python (1)
- RNG (12)
- Random (12)
- Raspberry Pi (3)
- ReadablePassphrase (1)
- ReadablePassphraseGenerator (1)
- Remove (1)
- Repair (1)
- Restore (2)
- Reverse Proxy (1)
- Router (5)
- S/MIME (1)
- SMB (1)
- SMIME (1)
- SSD (1)
- Safe Mode (1)
- Scheduler (1)
- Script (1)
- Secure (1)

The most important firewall rule is the NAT rule, which translates public IP addresses to private ones. Without this, no Internet connectivity is possible.

```
[admin@Mikrotik-gateway] /ip firew
Flags: X - disabled, I - invalid,

10    ;;; Main NAT rule
      chain=srcnat action=masquera
      log-prefix=""
```

The **filters** tab are where the firewall rules actually live. They enforce whatever policies I have, that is, what may access what. There are three categories of rules I have:

- Stats rules - these are just to track GBs and number of packets.
- Allow rules - to allow particular connections.
- Deny rules - the Mikrotik firewall allows everything by default, so you need some rules to reverse that behaviour.

Note that most rules are applied to the `forward` chain. This is the one used when forwarding packets between networks (as opposed to packets within the same networks).

```
[admin@Mikrotik-gateway] /ip firew
Flags: X - disabled, I - invalid,
 1    ;;; Incoming Stats
      chain=forward action=passthr
 3    chain=forward action=passthr
 6    chain=forward action=passthr

 8    ;;; Outgoing Stats
      chain=forward action=passthr
 9    chain=forward action=passthr
12    chain=forward action=passthr
```

- Security (3)
- Seedlings (1)
- Self-Hosting (1)
- Sermon (6)
- Shame (1)
- Slack (1)
- Speech (2)
- Spin 5 SP513 (1)
- Spring (1)
- Step-By-Step (3)
- Storage Spaces (2)
- Submission (1)
- Surveillance (3)
- Swedish Method (1)
- Switch (1)
- Sysinternals (2)
- Syslog (1)
- System-Image (1)
- Tear Down (1)
- Terms (1)
- Terninger-Series (12)
- Threading (3)
- Titus (1)
- Troubleshooting (2)

```
29    ;;; Allow restricted networl
      chain=forward action=accept
      dst-port=80,443 log=no log-p

30    ;;; Allow DNS access for al
      chain=input action=accept pr

31    ;;; Allow NTP access for al
      chain=input action=accept pr

32    ;;; Allow SMB / CIFS access
      chain=forward action=accept
      dst-port=445 log=no log-pre

33    ;;; Allow restricted networl
      chain=forward action=accept
      dst-port=80,443,22 log=no lo

34    ;;; Allow printer access fro
      chain=forward action=accept
      log-prefix=""

43    ;;; Full access to INTERNAL
      chain=input action=accept sr


44    ;;; Drop access to LAN from
      chain=forward action=reject
      dst-address-list=internal_tr

49    ;;; Drop external access by
      chain=input action=drop pro

50    ;;; drop external access by
      chain=input action=drop pro
```

IPv6 firewall is considerably simpler: just the accounting rules. Although that’s probably more due to my laziness than best practise.

```
[admin@Mikrotik-gateway] /ipv6 fir
Flags: X - disabled, I - invalid,
 1    ;;; Incoming Stats
      chain=forward action=passthr
 3    chain=forward action=passthr
```

- Ts&Cs (1)
- UEFI (1)
- Uninstall (1)
- Upgrade (1)
- User (1)
- User Interface (2)
- VLAN (1)
- Vegies (1)
- Video (3)
- WIndows Upgrade (1)
- Wbadmin (1)
- WiFi (4)
- Windbg (1)
- Windows (3)
- Windows 10 (3)
- Windows Activation (1)
- Windows-10 (1)
- Winsock (1)
- X.509 (1)
- Yubikey (1)
- anti-virus (1)
- eM Client (1)
- malware (1)
- remove (1)

```
chain=forward action=passthr

;;; Outgoing Stats
chain=forward action=passthr
chain=forward action=passthr
chain=forward action=passthr
```

7. Testing

Once configured, you should be able to ping the new IP addresses you just created.

And the final test is to connect a phone to the new WiFi network. Make sure it gets an IP address (if not, the WiFi interface itself or the DHCP server is mis-configured). And try to access the Internet (if you can't, the NAT rule or another firewall rule is probably broken).

It's also useful to keep an eye on the **Log**, as errors may appear in there to help you track down problems. And look against firewall rules to see when packet counts increase, that is a hint where things might be getting blocked.

Phones Connected to the Phone WiFi!

Phones Connected to the Phone WiFi!

Conclusion

You can create many new WiFi networks on a Mikrotik router to segregate and restrict devices.

The process is more involved than on most home routers, but considerably more flexible.

- security (1)
- uninstall (1)
- vLAN (1)
- virus (1)
- web.config (1)

TAG CLOUD

.NET .NET Core .NET Standard 4G About Access Point Accountability Software Agreement Android Authoring Autoruns Azure Backup Bandwidth Bible Bible Study Bible Talk BitBucket Blog Blue Screen Boot-Failure Bricked Broken Bsod Bug Check C# CPRNG CPU Camera Cellphone Certificate Certify Church Cloud Codeplex Compatibility Computer Conceptual Conditions Configuration Content Costings Covernant Eyes Crash Crashplan Crypto Cryptography Custom Domain Day Night Cycle DeleteFacebook Diagnose Dice Disassemble Disaster-Recovery Driver Verifier Dropbox Easter Easter Friday Email File History Firewall Fortuna Garden Gateway Godliness Good Friday Guest HDD HTTPS Hard Disk Hardware Herbs Hexo Hexo-Bootstrap-Series Home Router Home Security Honour Hosting How-To Hyper-V IIS Idiot Poof Installation Internals Internet Intro Isolated Keepass Keybase LSI LTE Laptop Layered Service Provider Legal Lets Encrypt Malware Messaging Meta Metadata Microsoft Account Migration Mikrotik Mobile Internet Mobile Phone Model N15W1 Modem Motherboard MotionEye NTP Netboot Netflow Netinstall Network Networking New User Nfdump Nmap NuGet Nuke it from Orbit Outage PGP Password Philippians Planting Pool Porting Presentation Printer Privacy Property Public Speaking Python RNG Random Raspberry Pi ReadablePassphrase ReadablePassphraseGenerator Remove Repair Restore Reverse Proxy Router S/MIME SMB SMIME SSD Safe Mode Scheduler Script Secure Security Seedlings Self-Hosting Sermon Shame Slack Speech Spin 5 SP513 Spring Step-By-Step Storage Spaces Submission Surveillance Swedish Method Switch Sysinternals Syslog System-Image Tear Down Terms Terninger-Series Threading Titus Troubleshooting Ts&Cs UEFI Uninstall Upgrade User User Interface VLAN Vegies Video WIndows Upgrade Wbadmin WiFi Windbg Windows Windows 10 Windows Activation Windows-10

$ Donate Comments Share

#Firewall #Guest #Mikrotik #Network #WiFi

Comments for this thread are now closed ✕

Comments Community 1 Login

♡ Recommend 1 Share

Sort by Best

Sayeed • 2 years ago

Well explanation about MikroTik router wifi.

^ | v • Share ›

Winsock X.509 Yubikey anti-virus eM Client malware
remove security uninstall vLAN virus web.config

Murray's Blog