Windows DNS integration with DHCP

Windows DNS is a requirement for the Domain Controller on a Windows Domain. However, integrating that with a DHCP server is not well documented.

This will give instructions for a setup where an opnSense Firewall/Router running ISC DHCP will send updates to a Windows DNS server informing it of new DHCP leases.

While it is appearantly possible set up the DNS to accept updates from insecure servers, best practice is to have a shared key to authorize the DHCP server to send updates to the DNS server.

Note: This document is in the “collection of information” stage. This has not been tested at this time. Once this has been tested, this paragraph will be removed.

Create TSIG Key

A Transaction Signature will authorize the opnSense ISC DHCP to communicate with the Windows DNS server. This is a simple procedure, but all commands are not available on all machines.

You will need two things; a name and a secret key. The name can be anything you want as it simply identifies which key the DHCP and DNS servers will use. I prefer camel case and nothing but alphabetic characters for the name. We will use **YourKeyName** in this document.

The secret key is simply some random string of characters, base 64 encoded. You can use anything here. A random string of characters is best, and a minimum of 12 is desirable, but in this case, more is better. We will use **secretkey** in this document, but do **not** use that. Something like **pKbusdI_3Wnj4j81og3Vmu1YY3N7jNc3** is preferred. That was generated on a Unix machine with the command

```
< /dev/urandom tr -dc _A-Z-a-z-0-9 | head -c${1:-32};echo;
```

The only thing we need to do now is create the TSig key from your chosen Secret key. Try the following, in order, and stop when the first one works.

Using Basic Command Line

If you are comfortable on the command line, PowerShell or BASH, this will be the simplest.

```
# bash (Linux) example (should work with any shell)
echo 'secretkey' | base64
```

The result of the above is a single string. In this case, it is **c2VjcmV0a2V5Cg==** which is your TSig.

```
// Windows PowerShell script
$keyName = "YourKeyName"
$algorithm = "HMACSHA256"
```

```
$secret =
[Convert]::ToBase64String([Text.Encoding]::UTF8.GetBytes("YourSecretKey"))
$tsigKey = "key $keyName. { algorithm $algorithm; secret $secret; };"
$tsigKey
```

This will output the following line. I have highlighted the key name, algorithm and TSig

key **YourKeyName**. { algorithm **HMACSHA256**; secret **WW91clNlY3JldEtleQ==**; };

Using the DNS Manager

1. Open the DNS manager by running dnsmgmt.msc
2. Right click on the DNS server anme in the left panel and select **Properties**
3. Go to the **Security** tab
4. Click on **Add**
5. Enter the name you chose for the key (*YourKeyName*)
6. Choose an **Algorithm**. HMAC-SHA256 recommended
7. Enter your chosen secret key in the **Secret** box
8. Click Ok
9. *How do we get the key out of there???*

Using PowerShell

This will not work unless you have DNS Server Tools installed (we don't). It will randomly generate the secret key

```
# only works
$key = New-DnsServerSigningKey -Algorithm HMACSHA256 -KeyName "YourKeyName"
-KeyLength 256
$key | Format-List
```

Add key to Windows DNS Server

You should now have the three things you need; the key name, algorithm and encoded secret key.

1. Open the DNS Manager on your Windows Domain Controller
2. Right-click on the DNS server and select Properties.
3. Go to the Security tab and click Add to add the TSIG key
4. Enter the
    1. Key Name
    2. Algorithm
    3. Secret you generated (the result of all the processing)
5. Set permissions for the key to allow updates

Setting up DHCP

In this case, we will be setting the ISC DHCP4 service on an opnSense router to notify the DNS server of any new DHCP leases given out. You will need to log into your opnSense router via the webui as root, or a user with comparable permissions.

Using ISC DHCP Server

1. Navigate to Services | ISC DHCP4
2. Select the correct interface (ie, LAN)
3. Scroll down to the Dynamic DNS section
4. Enable **Dynamic DNS Updates** by checking the box
5. Fill in the form
    1. DNS Server: enter the IP address of the Windows DNS Server
    2. Key Name: Enter the name, (**YourKeyName**)
    3. Key: Enter the TSig key you generated
6. Ensure you have the same DNS Server as the Windows DNS Server
7. Ensure you have the same Domain Name as the DNS Server (ie, the Windows Domain Name)
8. Save, then Apply

Note that opnSense says “ISC DHCP is end-of-life and no longer receives updates or security patches. It is strongly recommended to migrate to KEA or Dnsmasq.”, however, I was unable to locate the information I needed to set up KEA. However, I'm putting my notes below just in case I want to revisit it.

Using KEA DHCP Server

Don't really have a way of testing this, but adding here anyway. opnSense does not provide a webui interface for this, but you can log in from the command line and edit /etc/kea/kea-dhcp4.conf

Add the following block within the block that contains the overall dhcp server definition, replacing the correct server IP, key name and TSig value

```
        "ddns": {
            "enable-updates": true,
            "server": "192.168.1.1",  // IP of your Windows DNS server
            "key-name": "YourKeyName", // TSIG key name
            "key": "TSig" // Base64-encoded TSIG secret
        }
```

The example I saw showed this block.

```
{
    "Dhcp4": {
        "subnet-data": [
            {
                "subnet": "192.168.1.0/24",
```

```
                "pools": [
                    {
                        "pool": "192.168.1.10 - 192.168.1.100"
                    }
                ],
                "option-data": [
                    {
                        "name": "domain-name",
                        "data": "yourdomain.local"
                    },
                    {
                        "name": "domain-name-servers",
                        "data": "192.168.1.1"
                    }
                ]
            }
        ],
        "ddns": {
            "enable-updates": true,
            "server": "192.168.1.1",  // IP of your Windows DNS server
            "key-name": "YourKeyName", // TSIG key name
            "key": "base64-encoded-secret" // Base64-encoded TSIG secret
        }
    }
}
```

Restart KEA after this either via the webui, or with the command

```
service kea reload
```

Again, this is something I found with duck.io, and it did not give attributions, so I don't know. I did find that others said that KEA will only put staticly defined leases, and will not send notification for dynamic. I did some additional research and it appears the kea-dkcp-ddns service is meant to communicate with Bind9, so that may be something to pursue.

Testing

All you need here is to find a workstation and renew the dhcp lease, then see if it updates the DNS. The following commands do that for you (command line)

```
# Windows
ipconfig /renew
# Unix (just about any of them, including BSD, as root)
dhclient -r
# Linux using ifconfig, as root
ifconfig eth0 down && ifconfig eth0 up
# Linux, with systemctl
sudo systemctl restart NetworkManager
# Linux, with Network Manager, where eth0 is your interface
```

```
nmcli device reapply eth0
# MacOS (where en0 is your interface, use ifconfig to find the name)
ipconfig set en0 DHCP
```

Now, simply check the DNS server to see if it has the dhcp lease in it. If not, check your DHCP and DNS logs to see what went wrong.

Links

Sorry, I did not keep good track of where I went. I spent a lot of time on duck.ai, but like most AI searches, it gave a lot of bogus information. Here is my 'sorta' list

- https://kb.isc.org/docs/kea-administrator-reference-manual
- https://docs.opnsense.org/manual/isc.html

- Unordered List Item