# Table of Contents

Overview .......................................................................................................................... 1

Target Audience .................................................................................................................. 1

Example LAN Network ........................................................................................................ 2

Ubuntu 18.04 LTS Desktop .................................................................................................. 4
  Install Ubuntu 18.04 LTS .................................................................................................. 4
  Update Ubuntu .................................................................................................................. 5

Install dnsutils .................................................................................................................... 6

Install dnsmasq .................................................................................................................... 6

dnsmasq Version Check ......................................................................................................... 6

Stop and Disable the System Name Server ........................................................................... 7

Backup the dnsmasq Configuration File ............................................................................... 7

Identify the Primary Ethernet Interface ................................................................................ 7

Edit the dnsmasq Configuration File ................................................................................... 9

Create a Boot Service for dnsmasq ...................................................................................... 10

Restart dnsmasq .................................................................................................................. 11

Verify DNS Resolution ........................................................................................................ 12

Update the Router or Firewall DHCP Settings ..................................................................... 13

LAN Clients - Update DNS Settings .................................................................................... 14
  Hard-coded DNS Settings .................................................................................................. 14
  DNS Automatically Obtain via DHCP ................................................................................ 14

Power Settings – Disable Automatic Suspend ...................................................................... 16

Troubleshooting Client ID (MAC Address) in DNS Queries .................................................... 18

ARP Resolution Troubleshooting ......................................................................................... 19

dnsmasq with Multiple VLANs .......................................................................................... 20

Managed Ethernet Switch: Configure VLANs ...................................................................... 21

Configure IEEE 802.1Q VLAN Support with Netplan .......................................................... 22
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alternate ifupdown VLAN Configuration</td>
<td>24</td>
</tr>
<tr>
<td>VLAN Client ID in DNS Lookups Verification</td>
<td>28</td>
</tr>
<tr>
<td>Appendix</td>
<td>30</td>
</tr>
<tr>
<td>Network Firewall Rules for DNS Traffic</td>
<td>30</td>
</tr>
<tr>
<td>EDNS0 References</td>
<td>30</td>
</tr>
</tbody>
</table>
Overview

This application explains how to configure **dnsmasq** as a simple DNS forwarder to add the Client ID (MAC Address) to DNS queries for integration with MalBlock DNS Security. The client MAC addresses will be detected and tracked by MalBlock for device identification, assignment to user-defined Groups for customized policy and pinpoint identification of malware infected devices:

![MalBlock Device Inventory](image)

**Target Audience**

This document is for IT administrators or similarly trained professionals. It assumes the IT administrator has Local Area Networking (LAN), router, switch and Linux configuration skills. Experience with **WireShark** packet captures is required for troubleshooting.
Example LAN Network

The configuration example below is for a basic LAN where all devices are in the same subnet and no VLANs are configured. Dnsmasq will be installed on an Intel NUC (Next Unit of Computing). The NUC is an affordable commercial grade mini PC with a 3-year warranty that supports the Ubuntu operating system. However, other servers may be used instead of the NUC.

![Small Business Network Diagram](image)

Dnsmasq is free and only runs on Linux-based operating systems. Ubuntu 18.04 LTS is installed on the Intel NUC and the configuration commands here are unique to Ubuntu, which is Debian-based. The commands for Red Hat Enterprise Linux (RHEL), CentOS, SUSE and the other Linux distributions will be different. Red Hat Enterprise Linux users should refer to “How to Configure DNS Caching Server with dnsmasq in RHEL” noting that dnsmasq version 2.76 or higher is required.

Example LAN Network Configuration:

- Router
  - The router @ 10.10.0.1 is the LAN default gateway.

- DHCP server is built into the router.
  - Subnet 10.10.0.0/24
  - Dynamic IP pool range: 10.10.0.50 – 10.10.0.100
o Static IP reservations:
  ▪ 10.10.0.5 is configured for the Intel NUC’s MAC Address outside of the dynamic IP pool. This ensures the NUC’s DNS server IP address is well known and will not change.
  ▪ 10.10.0.10 for the network administrator’s PC. This is optional and not required.

o DNS Primary Server is the NUC’s IP address 10.10.0.5.

- MalBlock DNS Security resolver IPv4 addresses are 72.200.254.11 and 72.200.255.11. dnsmasq will be configured to use these Internet DNS Servers. (IPv6 DNS servers are also available.)

Your LAN configuration will be different; however, the IP addressing scheme will remain consistent.
Ubuntu 18.04 LTS Desktop

A “bare bones” Intel NUC is configured in this tutorial, meaning it does not have an operating system pre-installed. A 256GB Solid-State Drive (SSD) drive and 4GB memory were installed separately.

Ubuntu 16.04 LTS is certified by Intel and Canonical for the NUC and could be installed per the Install Ubuntu Desktop on the Intel® NUC instructions. However 16.04 only supports dnsmasq v2.75, but Cox MalBlock DNS Security requires dnsmasq v2.76 or later for Client ID support with the add-mac=”text” option. Therefore Ubuntu 18.04 LTS is required to install dnsmasq v2.79.

Install Ubuntu 18.04 LTS

Installing the Intel/Canonical certified Ubuntu 16.04 LTS followed by upgrading 18.04 LTS is not recommended due to latent conflicts with the with the standard 18.04 release. Instead, install Ubuntu 18.04 Desktop as a new operating system.

The general installation steps are:

- Download Ubuntu Desktop 18.04 LTS and copy the file to a USB flash drive with a minimum of 4GB capacity.
- Follow the Install Ubuntu Desktop instructions.

Tips:

- Skip Step 3 “Boot from DVD” and go to Step 4 “Boot from USB flash drive”.
- Step 5 “Prepare to install Ubuntu” – check the “Install third-party software for graphics and WiFi hardware…” to obtain drivers for the NUC.

Ubuntu 18.04 LTS works well on the Intel NUC in our testing for business services.
Update Ubuntu

Although updating an operating system is optional, it is wise to do so to obtain the latest patches, security updates, upgrades; as well as to remove unused dependencies, and free up disk space.

To update Ubuntu, open a terminal window and run the following command:

```bash
admin@intelnuc:~$ sudo apt-get update && sudo apt-get upgrade -y && sudo apt-get autoremove && sudo apt-get autoclean
```

[sudo] password for admin:
Hit:1 http://archive.canonical.com/ubuntu bionic InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu bionic InRelease
...

Check the OS version:

```bash
admin@intelnuc:~$ cat /etc/os-release
NAME="Ubuntu"
VERSION="18.04.3 LTS (Bionic Beaver)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 18.04.3 LTS"
VERSION_ID="18.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=bionic
UBUNTU_CODENAME=bionic
```
Install dnsutils
For your reference, the following CLI tools are helpful for troubleshooting purposes:

- `dig` - query the DNS in multiple ways
- `nslookup` - an older form to use
- `nsupdate` - perform dynamic updates (See RFC2136)

Install dnsutils:
```bash
admin@intelnuc:~$ sudo apt-get install dnsutils
```

Install dnsmasq
```bash
admin@intelnuc:~$ sudo apt-get install dnsmasq
```
You may see several dnsmasq startup errors, including:

- `dnsmasq`: failed to create listening socket for port 53: Address already in use
- FAILED to start up
- Failed to start dnsmasq - A lightweight DHCP and caching DNS server

This is expected because `systemd-resolved` has not been disabled as it was needed for DNS resolution to install dnsmasq. The errors will be corrected in the following steps.

**dnsmasq Version Check**

EDNS0 Client ID (i.e. MAC Address) is supported in dnsmasq v2.76 and later.
```bash
admin@intelnuc:~$ dnsmasq -v
Dnsmasq version 2.79 Copyright (c) 2000-2018 Simon Kelley
Compile time options: IPv6 GNU-getopt DBus i18n IDN DHCP DHCPv6 no-Lua TFTP conntrack ipset auth DNSSEC loop-detect inotify
```

⚠️ Important: Dnsmasq is free software and you can redistribute it under the terms of the GNU General Public License, version 2 or 3.
Stop and Disable the System Name Server

Ubuntu uses the systemd-resolved DNS resolver process which must be stopped and disabled to prevent conflicts with dnsmasq.

admin@intelnuc:~$ sudo systemctl stop systemd-resolved
admin@intelnuc:~$ sudo systemctl disable systemd-resolved

Removed /etc/systemd/system/dbus-org.freedesktop.resolve1.service.
Removed /etc/systemd/system/multi-user.target.wants/systemd-resolved.service.

**Important:** The localhost will not have DNS resolution until dnsmasq is installed and configured. Use the steps shown below to install and configure dnsmasq.

Backup the dnsmasq Configuration File

**Tip:** Create a backup file in case you need to restore the original file.

Use this command:

admin@intelnuc:~$ sudo cp /etc/dnsmasq.conf /etc/dnsmasq.conf.bak

Identify the Primary Ethernet Interface

The name of the primary Ethernet interface is required to configure another ID. The name varies with different servers. It may be “eth0” or a Predicable Network Interface Names “eno1”. Run the `ifconfig` command to display the server’s Ethernet interfaces:

admin@intelnuc:~$ ifconfig
eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.10.0.5  netmask 255.255.255.0  broadcast 10.10.0.255
    inet6 fe80::f2f:bf9:ef9f:e65f  prefixlen 64  scopeid 0x20<link>
    ether 1c:69:7a:02:5b:ba  txqueuelen 1000  (Ethernet)
    RX packets 11029  bytes 14535933 (14.5 MB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 4142  bytes 417460 (417.4 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
    device interrupt 16  memory 0xc0a00000-c0a20000

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
    loop txqueuelen 1000  (Local Loopback)
    RX packets 10733230  bytes 751191606 (751.1 MB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 10733230  bytes 751191606 (751.1 MB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
Identify the Primary Ethernet Interface

wlp0s20f3: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
  ether d0:c6:37:d4:63:9a  txqueuelen 1000  (Ethernet)
  RX packets 0  bytes 0 (0.0 B)
  RX errors 0  dropped 0  overruns 0  frame 0
  TX packets 0  bytes 0 (0.0 B)
  TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
Edit the dnsmasq Configuration File

The default configuration file is located at /etc/dnsmasq.conf. Most of the file is commented out with a leading #. (Note: The Sample file contents below have been snipped for brevity.)

```
# Configuration file for dnsmasq.
#
# Format is one option per line, legal options are the same
# as the long options legal on the command line. See
# "/usr/sbin/dnsmasq --help" or "man 8 dnsmasq" for details.

# Listen on this specific port instead of the standard DNS port
# (53). Setting this to zero completely disables DNS function,
# leaving only DHCP and/or TFTP.
#port=5353

1. Edit the configuration file with the gedit text editor:

   admin@intelnuc:~$ sudo gedit /etc/dnsmasq.conf

2. Scroll to the end of the file and add/edit the lines highlighted in blue. (Note: the Ethernet interface name obtained from ifconfig must configured, e.g. "eno1").

   ...additional file content...

   # Include all the files in a directory except those ending in .bak
   #conf-dir=/etc/dnsmasq.d,.bak

   # Include all files in a directory which end in .conf
   #conf-dir=/etc/dnsmasq.d/*.conf

   # Cox Business MalBlock DNS Security integration with Client ID
   log-facility=/var/log/dnsmasq.log
   interface=eno1
   no-dhcp-interface=eno1        # Do not provide DHCP or TFTP on the specified
       # interface, but do provide DNS service.
   domain-needed               # Never forward plain names (without a dot or
       # domain part)
   bogus-priv                   # Never forward addresses in the non-routed
       # address spaces.
   no-resolv                    # Don't read /etc/resolv.conf
   no-poll                      # Don't use /etc/resolv.conf or any other file
   server=72.200.254.11         # MalBlock DNS primary resolver
   server=72.200.255.11         # MalBlock DNS secondary resolver
   cache-size=0                 # Disable local cache
   add-mac=text                 # Include the MAC address (Client ID) in
       # DNS lookups

3. Save the file and close the editor. Note: gedit may generate warning messages after closing such as:

   ** (gedit:3531): WARNING **: 18:45:58.490: Set document metadata
   failed: Setting attribute metadata::gedit-spell-language not
   supported
```
Warning message(s) occur when gedit is launched from the command line. These messages can be ignored.

Create a Boot Service for dnsmasq

Normally, the dependencies in /lib/systemd/system/dnsmasq.service would prevent dnsmasq from starting too soon before the primary Ethernet interface is up. However, a bug in Ubuntu 18.04 LTS causes dnsmasq to start before the interface is up, which disables dnsmasq and DNS resolution and results in a loss of Internet access.

The workaround is to:

1. Create a boot-up initialization script to start dnsmasq after the interface is up.
   
   ```
   admin@intelnuc:~$ cd /etc/systemd/system/
   admin@intelnuc:~$ sudo mkdir dnsmasq.service.d
   ```

2. Verify the directory has been created.
   
   ```
   admin@intelnuc:/etc/systemd/system$ ls -d dns*
   dnsmasq.service.d
   ```

3. Create a file network-online.conf:
   
   ```
   admin@intelnuc:/etc/systemd/system/dnsmasq.service.d$ sudo touch network-online.conf
   admin@intelnuc:/etc/systemd/system/dnsmasq.service.d$ sudo gedit network-online.conf
   ```

4. Copy and paste the following into the editor:
   
   ```
   [Unit]
   After=network-online.target
   Wants=network-online.target
   ```

5. Save the file and close the editor.

6. Display the file contents to verify accuracy:
   
   ```
   admin@intelnuc:/etc/systemd/system/dnsmasq.service.d$ cat network-online.conf
   [Unit]
   After=network-online.target
   Wants=network-online.target
   ```

7. Return to the home directory.
   
   ```
   admin@intelnuc:/etc/systemd/system/dnsmasq.service.d$ cd $home
   admin@intelnuc:~$
   ```
Restart dnsmasq

1. Reload the dnsmasq.service configuration file:
   
   admin@intelnuc:~$ systemctl daemon-reload

2. Restart dnsmasq to refresh its configuration:
   
   admin@intelnuc:~$ sudo systemctl restart dnsmasq

3. Check the dnsmasq operational status. The following command output confirms dnsmasq is active with no errors. (Note: The network-online.conf Drop-In that was created in step 3 of Create a Boot Service for dnsmasq is part of the runtime.)
   
   admin@intelnuc:/etc/systemd/system$ systemctl status dnsmasq

   ● dnsmasq.service - dnsmasq - A lightweight DHCP and caching DNS server
     Loaded: loaded (/lib/systemd/system/dnsmasq.service; enabled; vendor preset: enabled)
     Drop-In: /etc/systemd/system/dnsmasq.service.d
             └─network-online.conf
     Active: active (running) since Thu 2019-08-22 16:58:57 EDT; 1min 51s ago
     Process: 7113 ExecStartPost=/etc/init.d/dnsmasq systemd-start-resolvconf (code=exited, status=0/SUCCESS)
     Process: 7104 ExecStart=/etc/init.d/dnsmasq systemd-exec (code=exited, status=0/SUCCESS)
     Process: 7103 ExecStartPre=/usr/sbin/dnsmasq --test (code=exited, status=0/SUCCESS)
     Main PID: 7112 (dnsmasq)
     Tasks: 1 (limit: 4915)
     CGroup: /system.slice/dnsmasq.service
             └─7112 /usr/sbin/dnsmasq -x /run/dnsmasq/dnsmasq.pid -u dnsmasq -r /run/dnsmasq/resolv.conf -7 /etc/dnsmasq.d,.dpkg-dist,.dpkg-old,.dpkg-new --local-service --trust

   Aug 22 16:58:57 intelnuc systemd[1]: Starting dnsmasq - A lightweight DHCP and caching DNS server...
   Aug 22 16:58:57 intelnuc dnsmasq[7103]: dnsmasq: syntax check OK.
   Aug 22 16:58:57 intelnuc systemd[1]: Started dnsmasq - A lightweight DHCP and caching DNS server.
Verify DNS Resolution

The easiest method to verify DNS resolution is to open a web browser and visit multiple websites. If dnsmasq is providing DNS resolution, you will be able to use the Internet.

Alternatively, you can test DNS resolution using nslookup:

```
admin@intelnuc:~$ nslookup cox.com
Server: 127.0.0.1
Address: 127.0.0.1#53

Non-authoritative answer:
Name: cox.com
Address: 45.60.49.167
Name: cox.com
Address: 45.60.45.167
```

nslookup from the Intel NUC hosting dnsmasq will show the localhost server IP @ 127.0.0.1. Two IPv4 addresses are resolved for cox.com.

Throughout this guide, the Intel NUC IP address is 10.10.0.5. It was obtained from the ifconfig command output noted in the Identify the Primary Ethernet Interface section.

Use the following command to force nslookup to use the dnsmasq server IP address. The format is: nslookup <domain name> <dns server ip address>

```
admin@intelnuc:~$ nslookup nasa.gov 10.10.0.5
Server: 10.10.0.5
Address: 10.10.0.5#53

Non-authoritative answer:
Name: nasa.gov
Address: 52.0.14.116
Name: nasa.gov
Address: 23.22.39.120
Name: nasa.gov
Address: 2600:1f18:1f:db01:11af:58af:ae11:f645
```

nasa.gov resolves to two (2) IPv4 and two (2) IPv6 addresses.
Update the Router or Firewall DHCP Settings

The Intel NUC or other server running dnsmasq must be assigned a static IP address and the DHCP server must point LAN clients to the DNS Server. Instructions are router specific, however this is a routine configuration covered in the firewall or router manufacturer’s documentation. For example, Linksys WRT54G DNS Settings can be found at this link: https://setuprouter.com/router/linksys/wrt54g/dns.htm

The Linksys router Primary DNS is pointed to the Intel NUC at @ 10.10.0.5:

**Figure 3. Linksys WRT54G DHCP / DNS Configuration**

![Linksys WRT54G DHCP / DNS Configuration](image)

1. Click the Save Settings button at the bottom of the screen when finished.
LAN Clients - Update DNS Settings

Hard-coded DNS Settings

LAN clients with hard-coded DNS server settings must be manually reconfigured to point their DNS to the dnsmasq server at 10.10.0.5 or the actual IP address of your DNS server. A best practice is all LAN devices should obtain the DNS server IP address(es) automatically from DHCP. Refer to the device manufacturer’s guide to configure DNS.

DNS Automatically Obtain via DHCP

LAN clients will not update their DNS server settings until the DHCP configuration is refreshed. Therefore, clients will not immediately use the Intel NUC and dnsmasq as the DNS server; and will not be protected by MalBlock DNS Security.

Figure 4 displays a Windows 10 PC receives the “DNS server address automatically” during the DHCP offer, and will not update the settings until the DHCP configuration is refreshed:

Figure 4. Microsoft Windows TCP/IPv4 Properties: DNS Server

Use any option below to force a LAN client to renew its DHCP configuration:

- If the client is on the WiFi network, disconnect and reconnect to the WiFi network.
- Disconnect the Ethernet cable at the router or switch for each LAN client – computer, server or IoT device to force a disconnect/reconnect and DHCP renew.
- Login to the managed Ethernet switch (if available) and administratively force the Ethernet port Down (disable or shutdown) and then Up. This has the same effect as physically disconnecting the Ethernet cable.

- Microsoft Windows:
  Run Command Prompt as an administrator and run the following commands:
  ```
ipconfig /release
ipconfig /renew
```

- Ubuntu:
  ```
  admin@intelnuc:~$ sudo systemctl restart NetworkManager.service
  ```

- Reboot the device if no other option is available.

  **Note:** New LAN devices should appear in the MalBlock DNS Security dashboard within 15 minutes after pointing traffic to the dnsmasq server.
Power Settings – Disable Automatic Suspend

Ubuntu Desktop automatic suspend must be disabled so the DNS server can be powered up and online 24x7.

Figure 5. **System Ubuntu 18.04 LTS**

Use the following steps to disable automatic suspend.

1. Click the System icon at the top-right corner of the screen. (See System Ubuntu 18.04 LTS Figure 5.)

2. Click the Settings tool icon. (See System Ubuntu 18.04 LTS Figure 5.)

3. Click the Power menu entry, then set Automatic Suspend [when idle] to Off. (See Figure 6 System Ubuntu 18.04 LTS.)

4. Click the red x icon in the upper right corner of the screen to close the Settings window. (See Figure 6 System Ubuntu 18.04 LTS.)
Figure 6. **Power Settings: Automatic Suspend Off**

![Power Settings: Automatic Suspend Off](image)
Troubleshooting Client ID (MAC Address) in DNS Queries

Dnsmasq is configured with the `add-mac=text` option such that the client MAC address is included in the DNS queries. MalBlock will detect the MAC address and add it to the MalBlock Device Inventory. Malware-infected devices are then identified by MAC address for investigation and remediation.

The Client ID in DNS queries can be verified with WireShark as shown in Figure 7.

1. The client PC **10.10.0.10** sends a query for nasa.gov to the DNS server **10.10.0.5**.

2. The dnsmasq server **10.10.0.5** adds the PC’s MAC Address in the **EDNS Option Code 65073** payload, then forwards the query to the MalBlock Internet DNS resolver **72.200.255.11**.

**Figure 7. Client ID (MAC Address) in DNS Lookup Wireshark Packet Capture**

![Wireshark Packet Capture](image)

The client MAC Address **14:b3:1f:25:63:73** is **not forwarded** by the MalBlock DNS resolvers to other Internet recursive and authoritative resolvers to protect your privacy.
ARP Resolution Troubleshooting

Dnsmasq requires Address Resolution Protocol (ARP) to learn the MAC Address of the LAN client making a DNS lookup. If you do not see the client MAC address in the Wireshark packet capture, install `arp-scan` on the dnsmasq server (Intel NUC):

```
admin@intelnuc:~$ sudo apt-get install -y arp-scan
```

1. Run a scan for all devices visible on the primary Ethernet interface (on which dnsmasq is configured to listen) for DNS queries. (Reminder: Interface eno1 was obtained from `ifconfig` and configured as the listening interface in `dnsmasq.conf`.)

2. Replace the interface ID eno1 with your actual value, which may be eth0 or another name.

```
admin@intelnuc:~$ sudo arp-scan --interface=eno1 --localnet
```

**Interface: eno1, datalink type: EN10MB (Ethernet)**

Starting `arp-scan` 1.9 with 256 hosts (http://www.nta-monitor.com/tools/arp-scan/)

<table>
<thead>
<tr>
<th>Internet Address</th>
<th>Physical Address</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.10.0.1</td>
<td>e8:98:6d:37:93:11</td>
<td>(Unknown)</td>
</tr>
<tr>
<td>10.10.0.3</td>
<td>74:83:c2:70:12:0d</td>
<td>(Unknown)</td>
</tr>
<tr>
<td>10.10.0.2</td>
<td>80:2a:a8:1f:e3:6b</td>
<td>(Unknown)</td>
</tr>
<tr>
<td>10.10.0.10</td>
<td>14:b3:1f:25:63:73</td>
<td>(Unknown)</td>
</tr>
<tr>
<td>10.10.0.222</td>
<td>b8:27:eb:70:ce:c9</td>
<td>Raspberry Pi Foundation</td>
</tr>
</tbody>
</table>

**Result:** This confirms the Dell PC at IP address 10.10.0.10 having the MAC Address 14:b3:1f:25:63:73 is visible to dnsmasq running on the Intel NUC. The Intel NUC IP 10.10.0.5 and MAC address are not displayed in ARP neighbor scan because the NUC is the localhost, not a neighbor.

An ARP scan from the Dell PC displays the Intel NUC IP address 10.10.0.5 and MAC: C:\>arp -a

```
Interface: 10.10.0.10 --- 0x19
```

<table>
<thead>
<tr>
<th>Internet Address</th>
<th>Physical Address</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.10.0.1</td>
<td>e8-98-6d-37-93-11</td>
<td>dynamic</td>
</tr>
<tr>
<td>10.10.0.5</td>
<td>1c-69-7a-02-5b-ba</td>
<td>dynamic</td>
</tr>
<tr>
<td>10.10.0.222</td>
<td>b8-27-eb-70-ce-c9</td>
<td>dynamic</td>
</tr>
<tr>
<td>224.0.0.22</td>
<td>01-00-5e-00-00-16</td>
<td>static</td>
</tr>
<tr>
<td>224.0.0.251</td>
<td>01-00-5e-00-00-fb</td>
<td>static</td>
</tr>
<tr>
<td>224.0.0.252</td>
<td>01-00-5e-00-00-fc</td>
<td>static</td>
</tr>
<tr>
<td>239.255.255.250</td>
<td>01-00-5e-7f-ff-fa</td>
<td>static</td>
</tr>
<tr>
<td>255.255.255.255</td>
<td>ff-ff-ff-ff-ff-ff</td>
<td>static</td>
</tr>
</tbody>
</table>

In summary:

- Client ID in DNS lookups requires ARP visibility from the dnsmasq server to the various LAN clients. If the client isn't listed in the ARP table then dnsmasq will be unable to learn the MAC Address and it won't be displayed in the MalBlock Device ID inventory.

- The Client ID of the localhost running dnsmasq will not be displayed in the MalBlock Device Inventory because ARP is a neighbor discovery protocol.
**dnsmasq with Multiple VLANs**

Virtual LAN (VLAN) means to logically partition and isolate a network to simplify design and management. Because VLANs are isolated broadcast domains, ARP resolution between the dnsmasq server on VLAN1 will not work for clients on other VLANs 20, 30, 40, etc. Therefore, the Intel NUC dnsmasq server in VLAN 1 cannot discover the MAC Address of clients in the other VLANs / subnets. This constraint is noted in the [dnsmasq main page](https):

```bash
--add-mac=[base64|text]
```

Add the MAC address of the requestor to DNS queries which are forwarded upstream. This may be used to DNS filtering by the upstream server. The MAC address can only be added if the requestor is on the same subnet as the dnsmasq server. The mechanism used to achieve an EDNS0 option is not yet standardized, so it should be considered experimental. Exposing MAC addresses this way may have security and privacy implications. The warning about caching given for `--add-subnet` applies to `--add-mac`, too. An alternate encoding of MAC, as `base64`, is enabled by adding the `base64` parameter; and a human-readable encoding of hex-and-colons is enabled by adding the `text` parameter.

The solution is to configure VLAN trunking on the managed switch and create VLAN interfaces on the Intel NUC so it can participate in VLAN broadcast ARP messages.

Figure 8 is an illustration of a network with VLAN 1 (default, untagged) and tagged VLANs 10, 20 and 30.

**Figure 8. VLAN Network Diagram**

![VLAN Network Diagram](image)

Each VLAN is a Layer 3 subnet configured on the managed Ethernet switch. VLAN trunks to the Intel NUC are configured to enable dnsmasq ARP resolution to discover the MAC address of all VLAN clients. This is suitable for a small network. A more advanced solution for larger networks is to deploy a DNS server that supports multiple [Network Interface Controller (NIC)](https), one for each VLAN/subnet.
Managed Ethernet Switch: Configure VLANs

Configure the manage switch interface connected to the Intel NUC as a Trunk port to tag the VLANs.

- The NUC is connected to switch port #8.
- VLAN 1 is the default untagged VLAN.
- All other VLANs must be Tagged for the NUC to have membership and ARP resolution.
- An Ubiquiti EdgeSwitch example is shown in Figure 9.
- Refer to the VLAN configuration instructions for your switch.
- Apply the updated configuration.

Figure 9. Managed Switch VLAN Tagging
Configure IEEE 802.1Q VLAN Support with Netplan

Ubuntu 18.04 LTS Desktop by default uses Netplan and YAML configuration files. Netplan replaced the older ifupdown package and /etc/network/interfaces config. See the next chapter for the older ifupdown method if you’re not using Netplan.

The steps to configure VLAN support in Netplan are:

1. Make a backup copy of the .yaml file:
2. admin@intelnuc:~$ cd /etc/netplan
3. Make a backup copy:
4. admin@intelnuc:/etc/netplan$ sudo cp 01-network-manager-all.yaml 01-network-manager-all.yaml.bak
5. Edit the file with gedit:
6. admin@intelnuc:/etc/netplan$ sudo gedit 01-network-manager-all.yaml
7. Add the following lines highlighted in blue to configure the VLAN interfaces, use interface names to match your scenario. YAML requires strict indentation using spaces, not tabs. Lines are indented with two spaces here:

```
# Let NetworkManager manage all devices on this system network:
version: 2
renderer: NetworkManager

ethernets:
  eno1:
    dhcp4: true

vlans:
  eno1.10:
    id: 10
    link: eno1
    dhcp4: yes

  eno1.20:
    id: 20
    link: eno1
    dhcp4: yes

  eno1.30:
    id: 30
    link: eno1
    dhcp4: yes
```

8. Apply the new network configuration:
9. admin@intelnuc:/etc/netplan$ sudo netplan apply
10. Verify the new configuration is active with the `ifconfig` or "ip a" command. The following shows the primary ethernet interface (eno1) and VLANs are up and are assigned a DHCP IP address. The important values are highlighted in yellow:

   admin@intelnuc:/etc/netplan$ ifconfig

   eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
   inet 10.10.0.5 netmask 255.255.255.0 broadcast 10.10.0.255
   inet6 fe80::1e69:7aff:fe02:5bba prefixlen 64 scopeid 0x20<link>
   ether 1c:69:7a:02:5b:ba txqueuelen 1000 (Ethernet)
   RX packets 976 bytes 597123 (597.1 KB)
   RX errors 0 dropped 0 overruns 0 frame 0
   TX packets 1464 bytes 171059 (171.0 KB)
   RX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
   device interrupt 16 memory 0xc0a00000-c0a20000

   eno1.10: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
   inet 10.10.1.51 netmask 255.255.255.0 broadcast 10.10.1.255
   inet6 fe80::1e69:7aff:fe02:5bba prefixlen 64 scopeid 0x20<link>
   ether 1c:69:7a:02:5b:ba txqueuelen 1000 (Ethernet)
   RX packets 123 bytes 8188 (8.1 KB)
   RX errors 0 dropped 0 overruns 0 frame 0
   TX packets 99 bytes 10772 (10.7 KB)
   RX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

   eno1.20: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
   inet 10.10.2.51 netmask 255.255.255.0 broadcast 10.10.2.255
   inet6 fe80::1e69:7aff:fe02:5bba prefixlen 64 scopeid 0x20<link>
   ether 1c:69:7a:02:5b:ba txqueuelen 1000 (Ethernet)
   RX packets 9 bytes 1166 (1.1 KB)
   RX errors 0 dropped 0 overruns 0 frame 0
   TX packets 91 bytes 10002 (10.0 KB)
   RX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

   eno1.30: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
   inet 10.10.3.51 netmask 255.255.255.0 broadcast 10.10.3.255
   inet6 fe80::1e69:7aff:fe02:5bba prefixlen 64 scopeid 0x20<link>
   ether 1c:69:7a:02:5b:ba txqueuelen 1000 (Ethernet)
   RX packets 38 bytes 24118 (24.1 KB)
   RX errors 0 dropped 0 overruns 0 frame 0
   TX packets 126 bytes 15969 (15.9 KB)
   RX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

Result: All interfaces have a valid IPv4 address in the correct Layer 3 subnet and are "Up" and “Running".
Alternate ifupdown VLAN Configuration

If your Ubuntu system is not using Netplan, this section describes how to configure VLANs based on the older ifupdown and VLAN packages using the /etc/network/interfaces configuration file.

1. Install the VLAN package:

   admin@intelnuc:~$ sudo apt-get install vlan
   admin@intelnuc:~$ cd /etc/network
   admin@intelnuc:/etc/network$ cat interfaces

   # interfaces(5) file used by ifup(8) and ifdown(8)
   auto lo
   iface lo inet loopback

2. Make a backup copy:

   admin@intelnuc:/etc/network$ sudo cp interfaces interfaces.bak

3. Edit the interfaces file:

   admin@intelnuc:/etc/network$ sudo gedit interfaces

4. Define VLANs interfaces adding the following lines, modify to match your scenario. Recall the Intel NUC Ethernet interface name is eno1 as shown by the ifconfig command. Replace eno1 with the name of your interface:

   # interfaces(5) file used by ifup(8) and ifdown(8)
   auto lo
   iface lo inet loopback

   auto enol
   iface enol inet dhcp

   auto enol.10
   iface enol.10 inet dhcp
       vlan-raw-device enol

   auto enol.20
   iface enol.20 inet dhcp
       vlan-raw-device enol

   auto enol.30
   iface enol.30 inet dhcp
       vlan-raw-device enol

5. Save the changes to the interfaces file and close the editor.
6. Display the file contents:

   admin@intelnuc:/etc/network$ cat interfaces
   # interfaces(5) file used by ifup(8) and ifdown(8)
   auto lo
   iface lo inet loopback
   auto enol
   iface enol inet dhcp

   # begin VLAN interfaces
   auto enol.10
   iface enol.10 inet dhcp
     vlan-raw-device enol

   auto enol.20
   iface enol.20 inet dhcp
     vlan-raw-device enol

   auto enol.30
   iface enol.30 inet dhcp
     vlan- raw-device enol

7. Restart networking to load the new VLAN configuration:

   admin@intelnuc:/etc/network$ sudo systemctl restart NetworkManager.service
8. Check the networking status:

```bash
admin@intelnuc:/etc/network$ sudo systemctl status NetworkManager.service
```

- NetworkManager.service - Network Manager
  - Loaded: loaded (/lib/systemd/system/NetworkManager.service; enabled; vendor preset: enabled)
  - Active: **active (running)** since Sun 2019-08-25 10:36:05 EDT; 9s ago
  - Docs: man:NetworkManager(8)
  - Main PID: 4775 (NetworkManager)
  - Tasks: 5 (limit: 4915)
  - CGroup: /system.slice/NetworkManager.service
    └─4775 /usr/sbin/NetworkManager --no-daemon
    └─4792 /sbin/dhclient -d -q -sf
      └─/usr/lib/NetworkManager/nm-dhcp-helper -pf /run/dhclient-eno1.pid -lf /var/lib/NetworkManager/dhclient-8603d6f7-1338-36c2-bf63-a971c442e27c-eno1.lease -cf /var/lib/N

Aug 25 10:36:05 intelnuc NetworkManager[4775]: <info>
[1566743765.8243] dhcp4 (eno1): lease time 432000
Aug 25 10:36:05 intelnuc NetworkManager[4775]: <info>
[1566743765.8243] dhcp4 (eno1): nameserver '10.10.0.5'
Aug 25 10:36:05 intelnuc NetworkManager[4775]: <info>
[1566743765.8243] dhcp4 (eno1): nameserver '10.10.0.222'
Aug 25 10:36:05 intelnuc NetworkManager[4775]: <info>
[1566743765.8243] dhcp4 (eno1): state changed unknown -> bound
Aug 25 10:36:05 intelnuc NetworkManager[4775]: <info>
Aug 25 10:36:05 intelnuc NetworkManager[4775]: <info>
Aug 25 10:36:05 intelnuc NetworkManager[4775]: <info>
Aug 25 10:36:05 intelnuc dhcpclient[4792]: bound to 10.10.0.5 -- renewal in 175674 seconds.
Aug 25 10:36:05 intelnuc NetworkManager[4775]: <info>
Aug 25 10:36:09 intelnuc NetworkManager[4775]: <info>
[1566743769.2067] manager: startup complete
9. Verify the primary ethernet interface (eno1) and VLANs are up and are assigned a DHCP IP address. The important items are highlighted in yellow:

```
admin@intelnuc:/etc/network$ ifconfig
eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.10.0.5 netmask 255.255.255.0 broadcast 10.10.0.255
    inet6 fe80::f2f:bf9:ef9f:e65f prefixlen 64 scopeid 0x20<link>
    ether 1c:69:7a:02:5b:ba txqueuelen 1000 (Ethernet)
    RX packets 246477 bytes 16260353 (16.2 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 576036 bytes 684797799 (684.7 MB)
    TX errors 0 dropped 0 collisions 0
    device interrupt 16 memory 0xc0a00000-0xc0a20000

eno1.10: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.10.1.51 netmask 255.255.255.0 broadcast 10.10.1.255
    inet6 fe80::1e69:7aff:fe02:5bba prefixlen 64 scopeid 0x20<link>
    ether 1c:69:7a:02:5b:ba txqueuelen 1000 (Ethernet)
    RX packets 1033 bytes 59502 (59.5 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 172 bytes 19532 (19.5 KB)
    TX errors 0 dropped 0 collisions 0

eno1.20: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.10.2.51 netmask 255.255.255.0 broadcast 10.10.2.255
    inet6 fe80::1e69:7aff:fe02:5bba prefixlen 64 scopeid 0x20<link>
    ether 1c:69:7a:02:5b:ba txqueuelen 1000 (Ethernet)
    RX packets 6 bytes 832 (832.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 173 bytes 19600 (19.6 KB)
    TX errors 0 dropped 0 collisions 0

eno1.30: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.10.3.50 netmask 255.255.255.0 broadcast 10.10.3.255
    inet6 fe80::1e69:7aff:fe02:5bba prefixlen 64 scopeid 0x20<link>
    ether 1c:69:7a:02:5b:ba txqueuelen 1000 (Ethernet)
    RX packets 225 bytes 56526 (56.5 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 298 bytes 38199 (38.1 KB)
    TX errors 0 dropped 0 collisions 0
```

Result: All interfaces have an IPv4 address in the correct Layer 3 subnet and are “Up” and “Running”.

VLAN Client ID in DNS Lookups Verification

1. Launch Wireshark on the Intel NUC dnsmasq server and start a packet capture.
2. Browse the web from a client on a VLAN.

VLAN Client ID (MAC Address) in DNS Lookup Wireshark Packet Capture

Figure 10 depicts an Apple iPhone on VLAN 30 (WiFi) performing a DNS lookup for forecast.weather.gov. The table below the image describes each highlighted value.

<table>
<thead>
<tr>
<th>Value (beginning at top left)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.10.3.14</td>
<td>iPhone Client on VLAN 30 (WiFi)</td>
</tr>
<tr>
<td>10.10.05</td>
<td>Intel NUC DNS Server</td>
</tr>
<tr>
<td>72.200.254.11</td>
<td>MalBlock DNS Security Server</td>
</tr>
<tr>
<td>EDNS0 version: 0</td>
<td>EDNS0 record</td>
</tr>
<tr>
<td>Option Code: Unknown (65073)</td>
<td>EDNS option code 65073</td>
</tr>
<tr>
<td>c0:b6:58:39:6e:3a</td>
<td>EDNS Option Data: Client MAC Address</td>
</tr>
</tbody>
</table>

If Client ID is working, the EDNS0 record for Option Code 65073 will contain the client MAC Address.
Use the following steps to locate the DNS query packet in Wireshark. (See 0.)

1. From the main menu, click the Edit heading in the top toolbar and click Find Packet.
2. Click the String option in the search bar.
3. Enter weather.gov in the search string.
4. Click the Find button.
Appendix

Network Firewall Rules for DNS Traffic

A best practice for security hardening is to implement network firewall rules to block rogue DNS clients and restrict dnsmasq server traffic to DNS port 53. For example, the following security policy rules (outlined in Figure 12) are implemented on a Palo Alto firewall.

<table>
<thead>
<tr>
<th>Rule #: Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule 13: DNS Private Resolvers</td>
<td>Allows LAN client to dnsmasq server DNS traffic on port 53. Recall the Intel NUC is running dnsmasq. Because only DNS traffic is allowed, the server is protected from client hacking attempts via VNC, ssh, telnet, etc.</td>
</tr>
<tr>
<td>Rule 14: DNS Internet Servers</td>
<td>Allows the Intel NUC dnsmasq server to send DNS traffic only to the Cox MalBlock resolver IP addresses.</td>
</tr>
<tr>
<td>Rule 15: Block Rogue DNS Clients</td>
<td>Blocks DNS traffic from any LAN/VLAN client that is not using the MalBlock resolvers, including the dnsmasq server. This prevents hard-coded, misconfigured or DNS hijacker infected clients from bypassing MalBlock. Policy violations will be displayed in the firewall traffic logs for investigation.</td>
</tr>
</tbody>
</table>

Figure 12. Palo Alto Firewall Security Policy Rules for DNS Traffic

Policy rules are evaluated in sequential order. Non-compliant DNS traffic that does not match one of the rules will be denied by the interzone-default rule.

Equivalent firewall rules can be implemented on most stateful and Next Generation firewalls. If not using Zones, substitute the LAN/VLAN Layer 3 subnet IP ranges for the traffic source match conditions.

EDNS0 References


End of Document