**Link-Local Multicast Name Resolution (LLMNR)**

This one is a biggie, and you’ve probably heard [Jordan](http://www.twitter.com/rev10D), John, me, and all the others say it many many times. LLMNR was (is) a protocol used that allowed name resolution without the requirement of a DNS server. It was (is) able to provide a hostname-to-IP based off a multicast packet sent across the network asking all listening Network-Interfaces to reply if they are authoritatively known as the hostname in the query.  It does this by sending a network packet to port UDP 5355 to the multicast network address (all layer 2).

You into RFC’s?  Me too: <https://tools.ietf.org/html/rfc4795>

**The Impersonator:**

What if you configure a node on the network to authoritatively say that it is, no matter what the query, exactly who the query is looking for. Let’s call this evil node “I’mEveryoneNotReally.” This creates a race-condition for the client. The client who is requesting the information will accept (and wholly trust) whoever answers first as the authoritative answer, because, based on the protocol specifications, the only responses it should receive are authoritative (and trustworthy.)

**The Hash Harvester:**

Windows (other operating systems too!) will use LLMNR in certain circumstances to identify certain machines on the network, such as file-servers. If Windows attempts to use LLMNR to identify the server of a file-share and it receives a reply, it will send the current user’s credentials directly to that server assuming it wouldn’t have replied if it wasn’t the authoritative file-server. If that LLMNR received response was actually an impersonator (I’mEveryoneNotReally), Windows just disclosed that user’s credential hash to a third-party. What’s worse? The impersonator may forward that packet to the actual file-server, so the user never realizes anything is amiss.

**Do we even need LLMNR?**

It was useful back in the day when DNS servers required costly processing power and system admins didn’t want them in every subnet (still don’t!) AdHoc networks can benefit greatly from them as well, but AdHoc networks are pretty uncommon these days. It made sense for quick resolution of names that were on the same subnet.  Problem is hackers realized the protocol didn’t have effective protections to prevent unauthorized nodes from authoritatively claiming they were anyone (everyone.) That said, in almost all cases LLMNR is no longer needed because proper DNS is configured. Disabling LLMNR closes a very serious risk vector.

**Disclaimer:**

I’ve got 99 problems, LLMNR isn’t one.

You are responsible for doing your own research in this matter and making the decision for what works best for your organization. If disabling this breaks stuff, try to un-disable it and fix what broke.  In my opinion, this is legacy protocol and presents enough risk that you are better to make whatever breaks work without LLMNR enabled.

**Disable LLMNR with Active Directory GPO:**

Active Directory has a GPO you can configure to prevent its domain workstations from using LLMNR.

Create a New or Update an existing Group Policy and Edit accordingly:

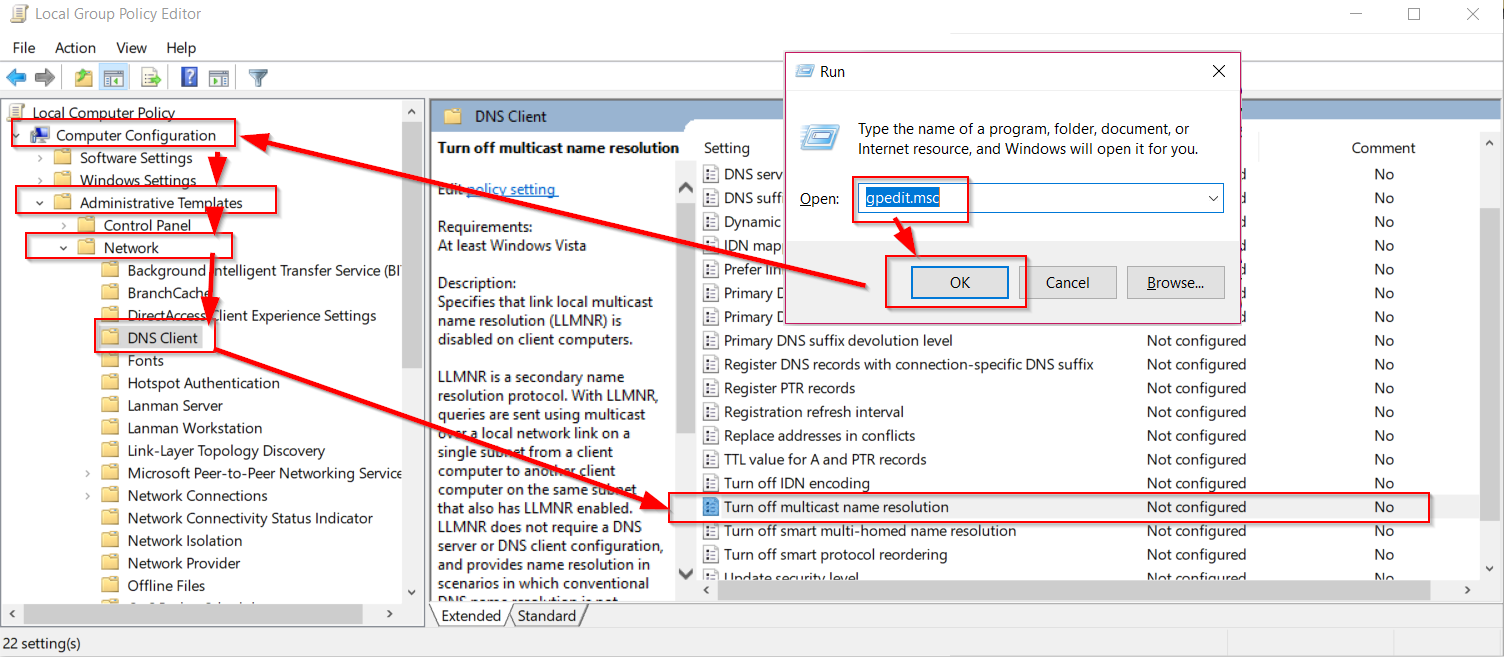
**Computer Configuration -> Administrative Templates -> Network -> DNS ClientEnable Turn Off Multicast Name Resolution**policy by changing its value to**Enabled**

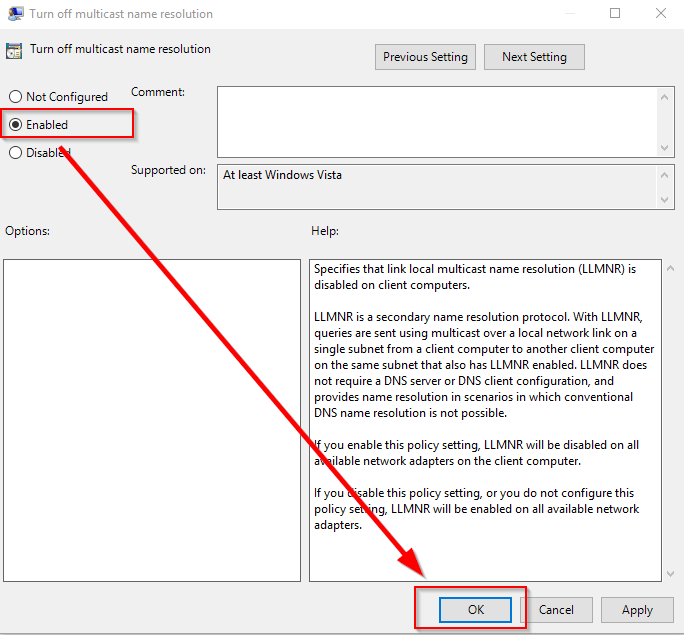
See screenshots below, essentially this operation is the same as using the Local Security Policy editor, with exception of making the modification on a Group Policy.

Disable LLMNR with Local Group Policy (Windows 7,8,10 Pro)

Use Local Group Policy editor by running gpedit.msc and modifying the policy.

Computer Configuration -> Administrative Templates -> Network -> DNS Client  
Enable Turn Off Multicast Name Resolution policy by changing its value to Enabled





**Disable LLMNR with Command Line (Single Workstation, Windows 7,8,10 Home)**

Run these guys from command line:

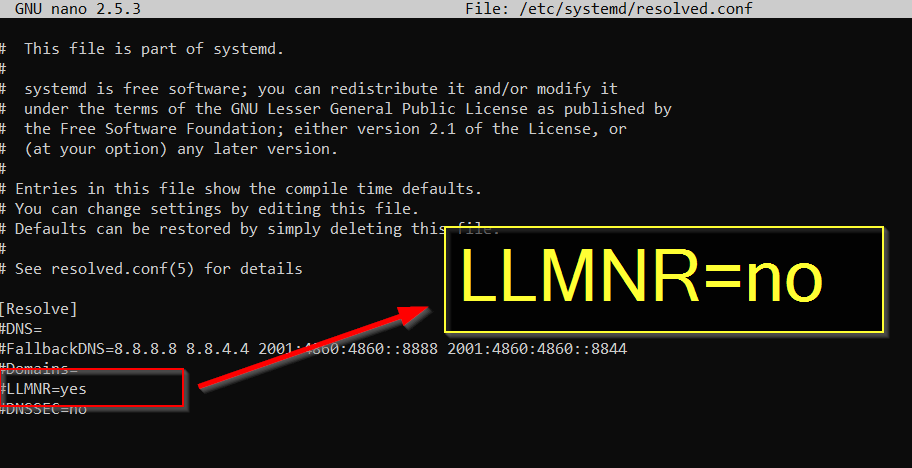
REG ADD  “HKLM\Software\policies\Microsoft\Windows NT\DNSClient”

REG ADD “HKLM\Software\policies\Microsoft\Windows NT\DNSClient” /v ” EnableMulticast” /t REG\_DWORD /d “0” /f

**Disable LLMNR on Linux (Ubuntu):**

Edit the line LLMNR=yes to LLMNR=no in /etc/systemd/resolved.conf

nano /etc/systemd/resolved.conf



Reboot.

**Links:**

RFC 4795: <https://tools.ietf.org/html/rfc4795>

How to benefit from Link-Local Multicast Name Resolution: <https://blogs.technet.microsoft.com/networking/2008/04/01/how-to-benefit-from-link-local-multicast-name-resolution/>

Vulnerability in DNS Resolution Could Allow Remove Code Execution: <https://docs.microsoft.com/en-us/security-updates/securitybulletins/2011/ms11-030>

**Black Hills Information Security Links:**

LLMNR Blog Posts: <https://www.blackhillsinfosec.com/?s=LLMNR>

Active Directory Blog Posts: <https://www.blackhillsinfosec.com/?s=Active+Directory>