“All That LLMNR Goodness” - Basic DNS, LLMNR, Responder, And Hashes

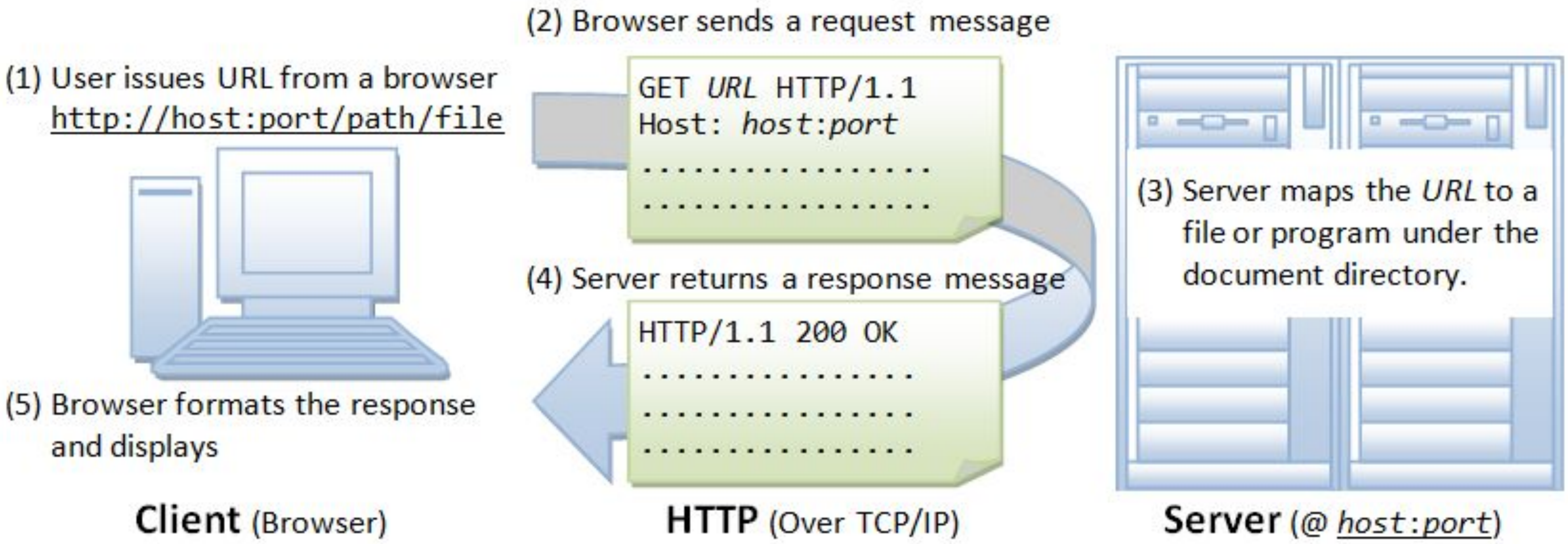
UWB Gray Hats

BUT FIRST…

What is a protocol?

- The “format” in which communication happens
  - Think about languages
- Let’s go back to the web server example
  - HTTP = Hyper Text Transfer Protocol
    - Meaning if you want to have hypertext (webpage content) transferred to you, you must follow the “language” that web servers speak
  - Let’s liken the web server to a hot dog stand
  - When you say “GET /” the web server says “HTTP 1.1 OK” and gives you the webpage
    - Just like if you say “Hi, can you give me a hot dog” the hot dog guy says “yeah” and does so
- Whenever you see “protocol,” it means communication format
- Whenever you see “__P”, it usually means protocol
  - HTTP, FTP, POP, SMTP, IMAP, SNMP, etc.

HTTP Example - what’s missing here?



(1) User issues URL from a browser
http://host:port/path/file

(2) Browser sends a request message

```
GET URL HTTP/1.1
Host: host:port
.................
.................
```

(3) Server maps the *URL* to a file or program under the document directory.

(4) Server returns a response message

```
HTTP/1.1 200 OK
................
................
................
```

(5) Browser formats the response and displays

Client (Browser)

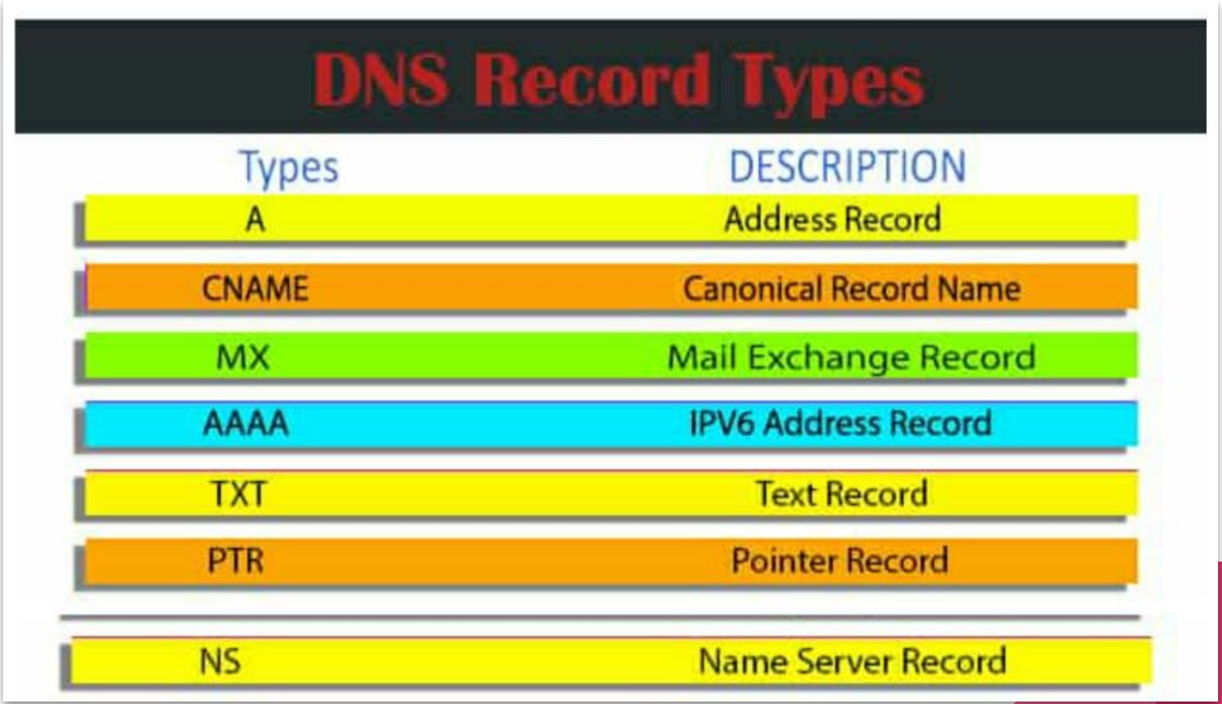
HTTP (Over TCP/IP)

Server (@ *host:port*)

What is DNS? What does it do?

- Domain Name System (DNS) is what matches host names to IP addresses
- Going back to web server example
- Whenever a client (you) wants to access the remote web server, you're going to know the URL of your website - but not the IP address in question
  - You need an IP address to talk to other computers
  - Do you really know Google's IP address(es) by heart? Probably not
  - A URL is NOT a hostname - we'll talk about that in a second
- Your webpage is going to reach out to a DNS server (by IP) requesting a matching "record" for the URL you looked up
  - DNS servers store DNS records. More on this in a bit
  - Question: How do we know the DNS IP address?
    - Because it's manually set by you or DHCP

What is DNS? What does it do?



DNS Record Types

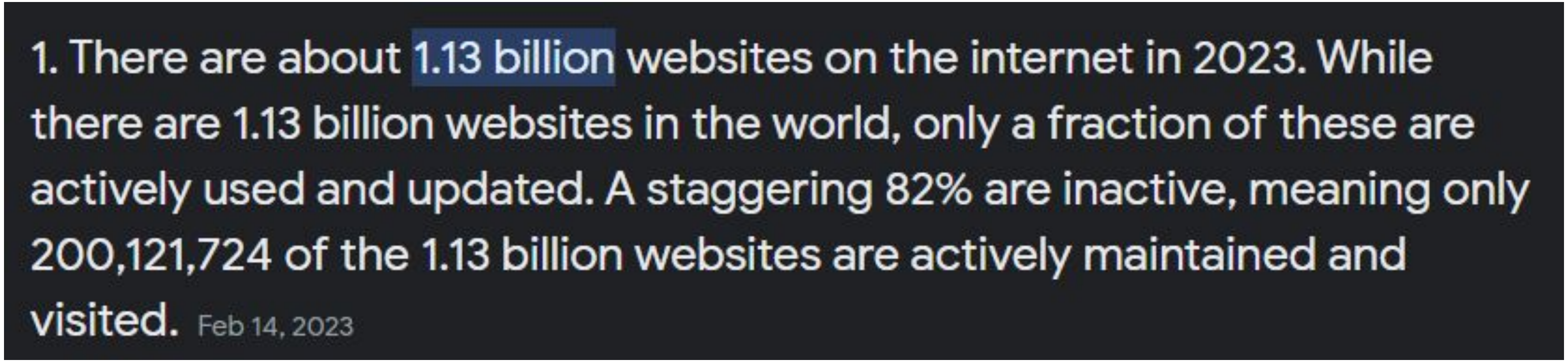
Types	DESCRIPTION
A	Address Record
CNAME	Canonical Record Name
MX	Mail Exchange Record
AAAA	IPV6 Address Record
TXT	Text Record
PTR	Pointer Record
NS	Name Server Record

Nslookup demo

- Try it out yourself!
- Choose your favorite URL
- Perform a DNS lookup
  - On Windows: go to command prompt, execute the nslookup command, enter your URL
  - On Linux/Mac: go to shell and do the same
  - Note the “default DNS server” at the start and the return IP
- Once you have your IP address, enter it into your web browser
- If all went well, your desired webpage should be requested and delivered!

But hold on a second…

- Ok, so I have a DNS server given to me by DHCP. But this is just one server. And there's a LOT of websites out there.



1. There are about 1.13 billion websites on the internet in 2023. While there are 1.13 billion websites in the world, only a fraction of these are actively used and updated. A staggering 82% are inactive, meaning only 200,121,724 of the 1.13 billion websites are actively maintained and visited. Feb 14, 2023

- There's no way every DNS server stores 1.13 billion URLs
  - So how am I able to (potentially) search up EVERY URL?
  - Good question!

What if…

- I have hosts within my network that I want to talk to
  - I only know the hostname because I’m Suraj from marketing, not an IT professional
  - Since this is on my local subnet, there won’t be any DNS records of my desired hosts
  - Since we know that DNS servers can point to other DNS servers, we can
- What if there is no local DNS server?
  - Hint: Ever since Active Directory (AD) started offering DNS (earliest version was 2012,) this pretty much never happens in an enterprise.
- Solution: Link-Local Multicast Name Resolution (LLMNR)
  - If you work in cybersecurity, this acronym should make chills run down your spine

What is LLMNR?

From ChatGPT:

- **Name Resolution Request:** When a computer needs to resolve the hostname of another device on the same local network and DNS is not available or fails to provide the resolution, it sends an LLMNR query. The query includes the hostname it wants to resolve.
- **Multicast Transmission:** Instead of sending the query to a specific DNS server, the LLMNR query is sent as a multicast packet to the local network segment. Multicast packets are directed to a specific group of network devices.
- **Neighbor Response:** If another computer on the local network recognizes the hostname and can provide the corresponding IP address, it responds to the LLMNR query. This response includes the hostname-to-IP address mapping.
- **Local Name Resolution:** The requesting computer receives the response and can then use the provided IP address for the target hostname, allowing communication to occur between the devices on the local network.

How can an attacker abuse LLMNR?

Let’s hear some ideas

How to Disable LLMNR & Why You Want To

Demo: Responder

- Open up a Kali machine
- Open up terminal
- Run sudo responder -I eth0
  - Runs responder on the eth0 port
  - Keep this process running
- Now, go to a Windows machine on the same network
- Try to access a fileshare in the search bar by searching “\\fileshare__”
  - This uses LLMNR! Think about why.
- Cancel responder, then go to /usr/share/responder/logs
- Look for your corresponding hash

But First… Brief Intro to Password Hashes

- Hash - Unique value generated from giving a unique input to a hashing function
  - Hash functions are one way - easy to calculate a hash, but virtually impossible to reverse a hash
  - This is NOT a random number - x input should always produce y output
  - Hash collisions are where multiple different inputs can produce the same output
- Operating systems store hashes to keep account passwords secure
  - When you try to sign in, the computer calculates a hash of your given password using the same function as the one which created the real password hash. If they match, you sign in.
  - In Linux, you can see all the password hashes of the system in the document /etc/shadow
- With LLMNR, the password isn’t sent out - only the hash
- Attackers will try to find out the prehashed value

Hashcat

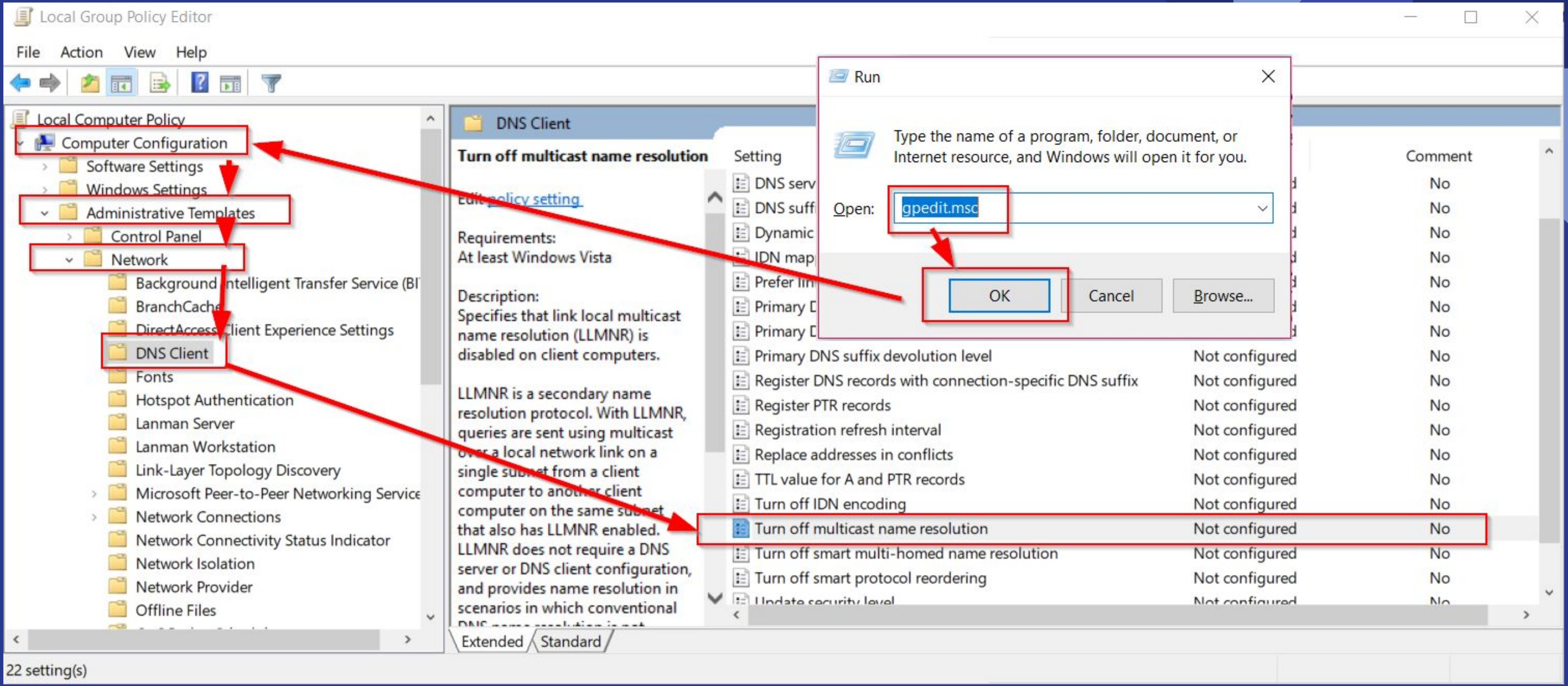
- Offline hash cracker (when you hear “hash cracker,” this is what it means)
- Fast due to GPU utilization
- hashcat -a 0 -m 5600 ./my_hash.txt /usr/share/wordlists/your_wordlist.txt
    - –a = “attack mode” (mode 0 is wordlist mode)
    - -m = “hash mode” (mode 5600 corresponds to NetNTLMv2 - look up “hashcat example hashes”)
        - This is you selecting which hashing algorithm you want to use
    - my_hash.txt - the file containing the hash(es) you want to be cracked
    - your_wordlist.txt - the file containing the passwords to be tried via the chosen hashing function
- Hashcat attempts to run various passwords through the chosen hashing algorithm several millions of times per second
    - The passwords can be brute forced, taken from a wordlist, etc.

Continuing with Responder

- hashcat -a 0 -m 5600 ./SMB-NTLMv2-SSP-192.168.1.8.txt /usr/share/wordlists/rockyou.txt
- Let it run
  - I specifically added the target password to rockyou.txt, so it might not be in the list for future
  - Hash cracker choices can save you a lot of time - brute force is guaranteed but slow.
- When it finishes, you should get a big block of text (the cracked hash in question) with the password at the end
- Think about the implications of this attack

How to counter this evil crazy exploit?

DISABLE LLMNR please and thank you



Local Computer Policy
Computer Configuration
Software Settings
Windows Settings
Administrative Templates
Control Panel
Network
Background Intelligent Transfer Service (BI
BranchCache
DirectAccess Client Experience Settings
DNS Client
Fonts
Hotspot Authentication
Lanman Server
Lanman Workstation
Link-Layer Topology Discovery
Microsoft Peer-to-Peer Networking Service
Network Connections
Network Connectivity Status Indicator
Network Isolation
Network Provider
Offline Files

DNS Client

Turn off multicast name resolution

Edit policy setting

Requirements:
At least Windows Vista

Description:
Specifies that link local multicast name resolution (LLMNR) is disabled on client computers.

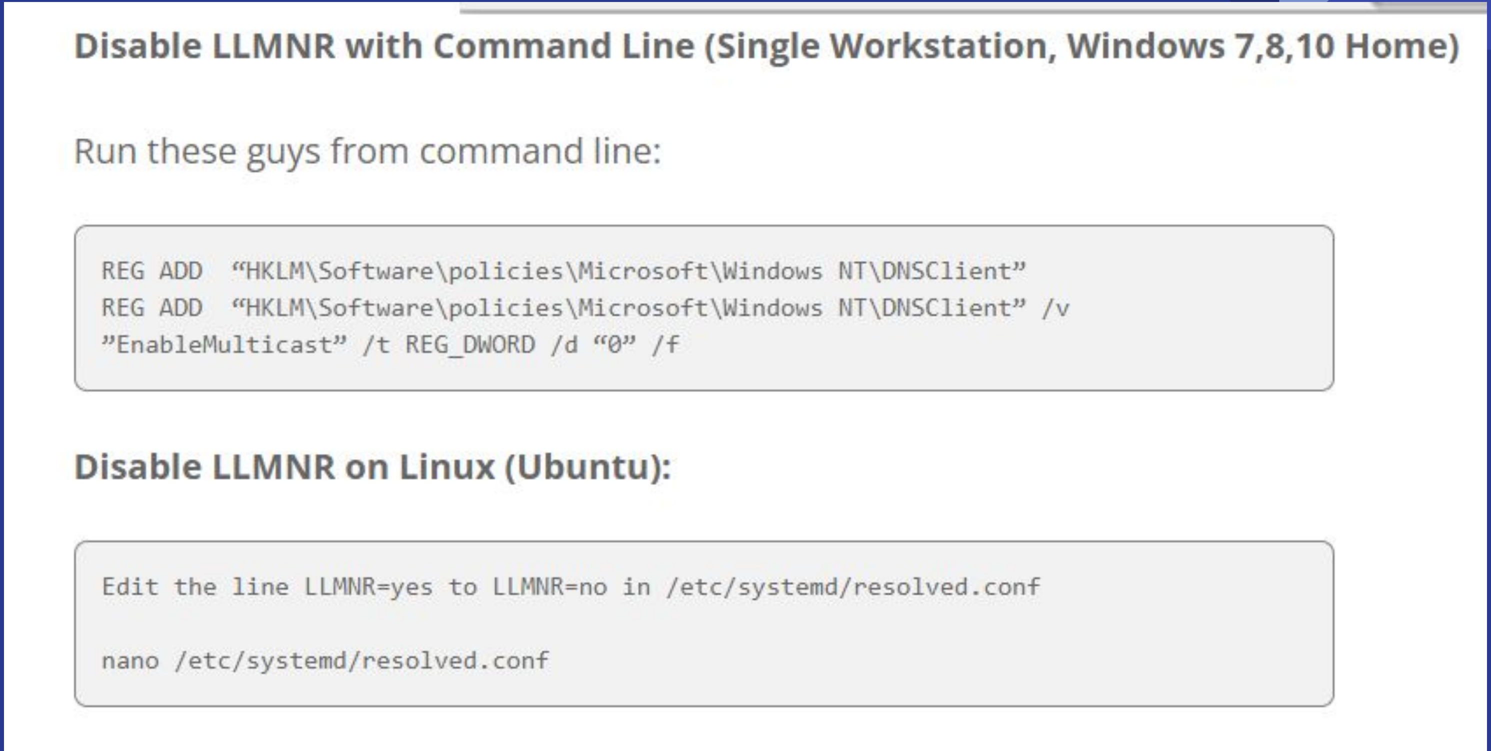
LLMNR is a secondary name resolution protocol. With LLMNR, queries are sent using multicast over a local network link on a single subnet from a client computer to another client computer on the same subnet that also has LLMNR enabled. LLMNR does not require a DNS server or DNS client configuration, and provides name resolution in scenarios in which conventional

Setting		Comment
DNS serv		No
DNS suff		No
Dynamic		No
IDN map		No
Prefer lin		No
Primary D		No
Primary D		No
Primary DNS suffix devolution level	Not configured	No
Register DNS records with connection-specific DNS suffix	Not configured	No
Register PTR records	Not configured	No
Registration refresh interval	Not configured	No
Replace addresses in conflicts	Not configured	No
TTL value for A and PTR records	Not configured	No
Turn off IDN encoding	Not configured	No
Turn off multicast name resolution	Not configured	No
Turn off smart multi-homed name resolution	Not configured	No
Turn off smart protocol reordering	Not configured	No
Update security level	Not configured	No

22 setting(s)

Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: gpedit.msc



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
```

Summary

- What is a protocol?
- What is DNS? How does it work?
- What is LLMNR? How is it different from DNS?
- Why is LLMNR insecure?
- How do we capture hashes from LLMNR?
- What even is a hash?
- How do we crack hashes?