PENTERA | Pentera Labs™ Research Series

Death by Default: Neglected network protocols you should know

A look at three common, legacy network protocols from a security perspective.

Yotam Mazurik

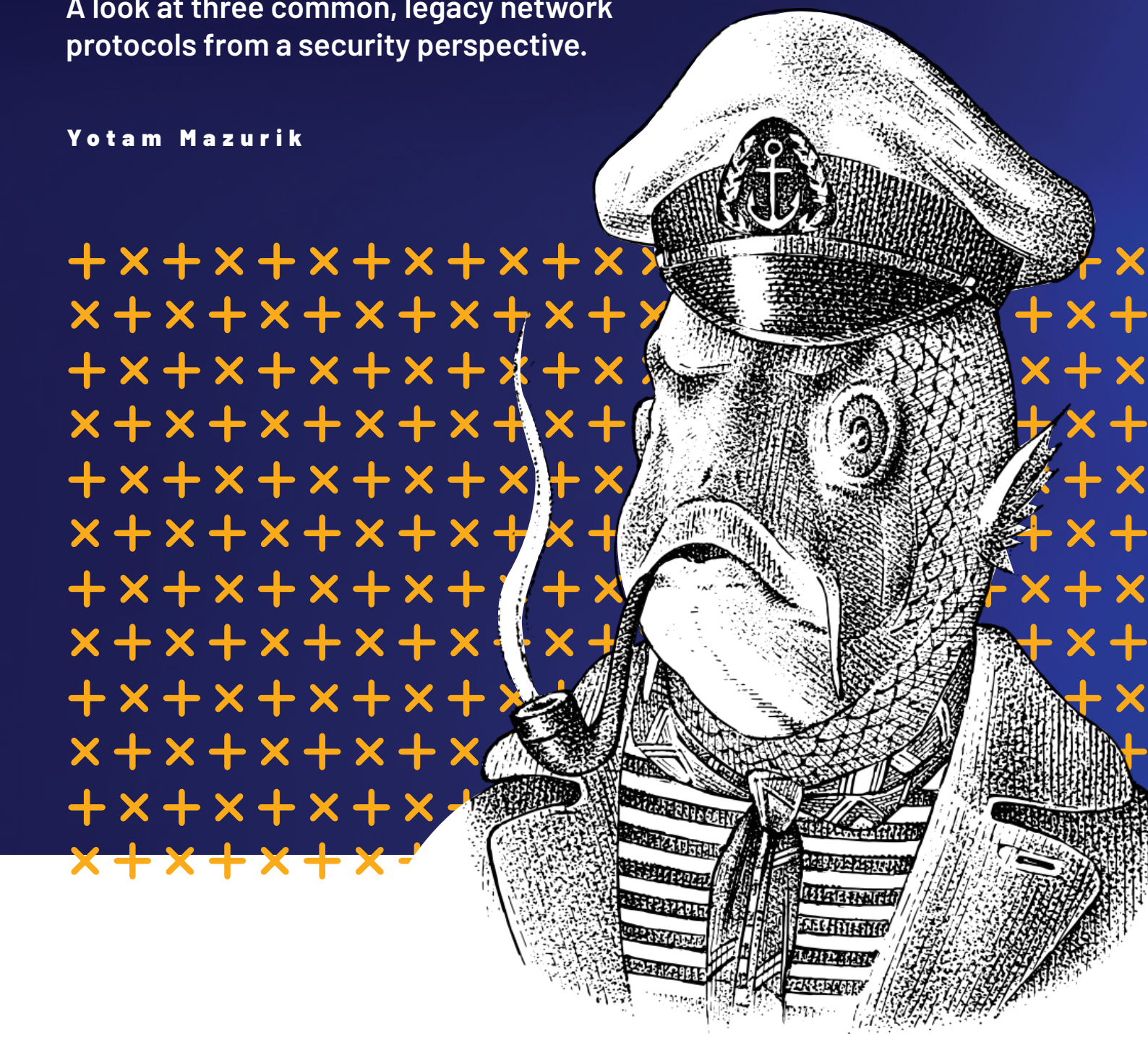


Table of contents

Purpose

To raise awareness and provide examples of network protocols that may have been neglected and therefore, abused.

Executive summary

I nternet technology is advancing at a rapid pace creating new attack surfaces or techniques, sometimes, before security teams are aware or able to update existing ones.

In this article, we will describe three network protocols that are implemented in most operating systems, and were used in different phases of cyber attacks over the last couple decades. Despite this, they aren't mentioned in some of the major security frameworks, and may not be top-of-mind for security teams.

This research is a great example of why it's important to stay-up-to-date with what's happening in technology, so we don't come to rely exclusively on known security mitigations.

Does it apply to my organization?

Yes, the research was conducted on common, widely-used network protocols that are implemented in most operating systems

Who should read this?

Security researchers, Pentesters, CISOs, Blue Teamers

Intro

Network protocols are established standards that determine how network devices transmit data and communicate with each other. These protocols can also be used by cyber criminals to attack networks. In this article, I will dive into three network protocols, MS-LLTD, MDNS and ICMPv6 that you should keep your eye on. I will explain how they function and demonstrate why defenders should secure them and how pen testers can attempt to abuse them.

The first two protocols are used in the Discovery phase, so I start by providing background on the topic before delving into the protocols. The third protocol can be exploited for an actual attack, so I provide background information about the attack and then show how to exploit the protocol to carry one out. Then, I will provide mitigation strategies.

I chose to focus on these three protocols because they are widely adopted in most common platforms and are likely to appear in most internal networks. They can also be used or exploited in ways organizations may not be familiar with. Therefore, I encourage you to read more about them and about additional pentesting-related research at Pentera Labs, where we publish our original findings and analyses of the work we conduct.

Background: The Discovery Phase

Network discovery is the process that allows devices to find each other on a network. In this phase, the physical device, usually defined by an ethernet address in 802.x networks, is mapped to any other type of logical information (usually IPv4 and IPv6 addresses). Network discovery will be the first action taken by a device to communicate with other devices in the network.

Discovery will also be one of the attacker's first actions in an attack. An attacker will usually start by mapping the environment to have a better understanding of the devices it's connected to. This makes discovery a critical phase for pen testers

Protocol 1: MS-LLTD

MS-LLTD is Microsoft's implementation of the LLTD protocol. LLTD, the "Link-Layer Topology Discovery" protocol, as defined in RFC 4957, is the standard for sending link-layer event notifications. The core purpose of this protocol is to enable devices to discover the link-layer topology of the network.

Microsoft divides devices using this protocol into two: Enumerators/Mappers and Responders. An enumerator/mapper device will initiate a "Quick Discovery" process. During this process, a broadcast message is sent to the ethernet broadcast address. Any "LLTD capable" device (responder) will respond with a "Hello" message containing a list of information about the station itself (ipv4, ipv6, host id, etc.) in a predefined structure called TLV.

MS-LLTD also defines other types of "tests" that are based on the "Quick Discovery" process mechanism. For discovery purposes, the "Quick Discovery" is more than enough for us pen-testers. This paper will cover the "Quick Discovery" method but feel free to research and leverage the other methods too.

Procedure of Use – Quick Discovery

First, we will send an LLTD Quick Discovery packet to the ethernet broadcast address.

```
Ethernet II, Src: VMware_e0:ef:22 (00:0c:29:e0:ef:22), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Destination: Broadcast (ff:ff:ff:ff:ff:ff)
> Source: VMware_e0:ef:22 (00:0c:29:e0:ef:22)
  Type: Link Layer Topology Discovery (LLTD) (0x88d9)
Link Layer Topology Discovery
  Version: 1
  Type of Service: Quick discovery (0x01)
  Reserved: 0x00
  Discovery function: Discover (0x00)
> Base header
  Generation Number: 0x465c
  Number of Stations: 0
```

Each "LLTD capable" device on the link is supposed to respond with a LLTD Hello message.

LLTD Hello messages are sent to the ethernet broadcast as well and will contain useful and will contain useful information.

```
Ethernet II, Src: WistronI_31:52:6a (f4:a8:0d:31:52:6a), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Destination: Broadcast (ff:ff:ff:ff:ff:ff)
> Source: WistronI_31:52:6a (f4:a8:0d:31:52:6a)
  Type: Link Layer Topology Discovery (LLTD) (0x88d9)
Link Layer Topology Discovery
  Version: 1
  Type of Service: Quick discovery (0x01)
  Reserved: 0x00
  Discovery function: Hello (0x01)
> Base header
  Generation Number: 0x0000
  Current Mapper Address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Current Apparent Address: 00:00:00_00:00:00 (00:00:00:00:00:00)
> TLVs
```

By processing the TLV list section we can gather a lot of information about the responder, such as:

- IPv4 Address
- IPv6 Address
- Host Name
- Ethernet Address

```
TLVs
> TLV Item (Host ID)
> TLV Item (Characteristics)
> Characteristics
> TLV Item (Physical Medium)
> TLV Item (IPv4 Address)
> TLV Item (IPv6 Address)
> TLV Item (Performance Counter Frequency)
> TLV Item (Link Speed)
> TLV Item (Machine Name)
> TLV Item (QoS Characteristics)
> TLV Item (End of Property List)
```

Predicted Behavior

Because MS-LLTD is a Microsoft-defined protocol, we can assume that most of Microsoft's operating systems will respond to these requests. Furthermore, other vendors may implement this standard in their operating systems.

Observations

- Windows hosts with the LLTDIO service enabled reply to "Quick Discovery" requests
- Linux and MacOS do not respond to any types of requests
- Some IOTs respond to "Quick Discovery" requests, such as printers and Smart TVs

Protocol 2: MDNS

mDNS or "Multicast Domain Name Service", a protocol that was proposed ~20 years ago and came to life 13 years later, enables hosts to perform DNS-like operations on a local network with no need for a serialized unicast DNS server.

The mDNS protocol is a zero-configuration service and is based on the existing DNS protocol. As in the DNS protocol, a host may use mDNS to query IP addresses (A, AAAA records), service pointers (SRV records), and others.

Procedure of Usage

As defined in RFC 6762, queries will be sent to IP Multicast (224.0.0.251) on UDP port 5353. Just like when querying DNS, a host that wants to resolve information will send an mDNS query to the multicast with a set of questions.

Reverse Lookup

We figured we could try to use the same methods we would use in a DNS reconnaissance situation. Therefore, we decided to try to perform a reverse-lookup. If any device on the local link has a PTR record for this IP address, it will answer with the host's name.

```
Internet Protocol Version 4, Src: 10.0.10.19, Dst: 224.0.0.251
User Datagram Protocol, Src Port: 54967, Dst Port: 5353
Multicast Domain Name System (query)
   Transaction ID: 0x0000
 > Flags: 0x0100 Standard query
   Questions: 1
   Answer RRs: 0
   Authority RRs: 0
   Additional RRs: 0
 ⌄ Queries
   > 2.10.0.10.in-addr.arpa: type ANY, class IN, "QU" question
```

Not so surprisingly, we found out that we were correct! Devices that implemented the mDNS service started to answer our queries. Sometimes they even provide additional data, like their IPv6 address.

```
Internet Protocol Version 4, Src: 10.0.10.2, Dst: 10.0.10.19
User Datagram Protocol, Src Port: 5353, Dst Port: 54967
Multicast Domain Name System (response)
   Transaction ID: 0x0000
 > Flags: 0x8400 Standard query response, No error
   Questions: 1
   Answer RRs: 1
   Authority RRs: 0
   Additional RRs: 0
 > Queries
 v Answers
   v 2.10.0.10.in-addr.arpa: type PTR, class IN, LGwebOSTV.local
        Name: 2.10.0.10.in-addr.arpa
        Type: PTR (domain name PoinTeR) (12)
        .000 0000 0000 0001 = Class: IN (0x0001)
        0... .... .... .... = Cache flush: False
        Time to live: 10 (10 seconds)
        Data length: 17
        Domain Name: LGwebOSTV.local
```

Observations

- Windows OSs do not respond to PTR queries. However, mDNS has started making its way into Windows 10 and newer versions.
- We were able to trigger an A record response. But in most cases, the host name wasn't known to us before the Network Discovery process.
- Apple devices respond to PTR queries and provide the following information: hostname, ipv4 and ipv6. To elaborate further, the Airplay discovery mechanism is based on mDNS.
- Not all Linux distributions support mDNS.
- IOTs may also respond to PTR queries.

Undiscovered

While observing mDNS traffic on Wireshark, we saw some mDNS queries over IPv6 from Windows hosts. The queries were immediately responded to from the same host that issued them and provided the A, AAAA records.

We were not able to trigger Windows hosts to respond to any type of mMDNS query without knowing the host name. However, we believe this will be discovered in future research as Microsoft keeps integrating mDNS to its operating systems.

Background: MITM Attacks

The third protocol, ICMPv6, can be leveraged for actual attacks. Below, I demonstrate how to leverage the ICMPv6 protocol to execute a MITM attack and also provide mitigation tips.

Therefore, let's start with a bit of background on MITM attacks.

MITM Attack Techniques

According to MITRE, here are the known MITM attack techniques:

- Arp poisoning: The most popular MITM attack technique. In ARP poisoning, an attacker transmits false ARP responses to the victim, stating the mac address is related to the desired IP address the victim is trying to reach.
- Rogue DHCP – An attacker will respond to DHCP requests and offer its own network configuration. This will lead to either DNS / Default GW hijacking. We covered this in our previous DHCP 101.
- ICMP Redirect – An attacker may send victims false ICMP redirect messages noting there is a better route to a specific destination, which is through the attacker.

Learn more about these techniques from the MITRE website.

Protocol 3: The ICMPv6 Protocol

ICMPv6 is the implementation of ICMP on the IPv6 protocol. It is similar to ICMP and contains most message types: Echo, Redirect, Destination Unreachable.

ICMPv6 introduces new features that try to solve different problems in IPv4.

One of them is the ICMPv6 extension called "Neighbor Discovery Protocol" or NDP.
We will do a deep-dive into that later in the article.

Before I show how to gain information about ICMPv6, a bit of background on IPv6.

Link-Local, SLAAC:

IPv6 has three different types of unicast addresses:

- **Link Local** – The equivalent to IPv4 automatic private IP addressing (169.254.0.0/16).
  It will be set automatically under the prefix of FE80::/10.
  These addresses are used only for the local network and are not forwarded by routers.
- **Site Local** – The equivalent of IPv4 internal reserved IP ranges (192.168.0.0/16, etc.).
  It will be set to the prefix FEC0::/10.
  These addresses are supposed to be used in internal IPv6 networks, but are deprecated.
- **Global** – The public IPv6 that is routed across the internet.
  It will be set to the prefix 2000::/3.

The SLAAC (Stateless Address Auto-Configuration) mechanism enables each device to assign a link-local & global IPv6 address to itself, with no need for a serialized mechanism for IP assignment.

This mechanism uses the device's mac address to generate a "unique" link-local address. After verifying this address is unique on the connected link, the device is assigned with the link-local address. Once the device has a link-local address it can start requesting global configurations from the router or the DHCP server.

The ND Protocol

The "Neighbor Discovery" protocol is the equivalent of ARP for IPv6 host discovery. It operates on the link-layer and is an extension of the ICMPv6 Protocol.

ND Protocol Types

1. Router Solicitation – Used by devices to identify routers on the link.
2. Router Advertisement – Used by routers to advertise themselves periodically or in response to a solicitation message.
3. Neighbor Solicitation – Used by devices to identify the ethernet address of neighbors (the equivalent of an ARP Request).
4. Neighbor Advertisement – Used by devices to respond to solicitation messages (the equivalent of an ARP Response).
5. Redirect – Used by routers to inform hosts of a better route to specific destinations.

Security & Operational Advantages

The ND protocol was designed to solve some long-standing problems:

Discovery Through Network Scanning

Traditional scanning techniques, like ICMP ping or TCP SYN discovery, are executed per-host. IPv6 introduces large-scale address ranges (default prefix of link local is /64) therefore, using these traditional discovery methods would take us an unscalable amount of time.

This means that with ND, attackers cannot base their discovery techniques on scanning anymore. Without doubt, this will toughen the first phases of an attack.

Neighbor Reachability

NDP enables declaring a host as "Unreachable". This will trigger a deletion of the cached entry from the OS.

Before declaring a host is unreachable, a node will send a solicitation message to the desired address. If no response has been received, eventually the destination host will be declared as unreachable.

This helps operating systems to maintain a true valid cache of neighbor hosts, unlike IPv4 which bases its neighbor discovery on ARP and has no unreachability mechanism.

Security Disadvantages

Of course, with every new protocol, attackers will always find a way to use it to their own advantage. Here are some ways they can do it:

Address Starvation – DAD

During the process of SLAAC, before setting an address, a host will evaluate if this address is already configured on the link. This is done by sending a neighbor solicitation message to the destination address. The process is called "Duplicate Address Detection" (DAD).

An attacker could answer every neighbor solicitation message and cause the victim to think the address is already in use. The operating system of the victim will eventually stop trying to configure an IP address and will wait for manual configuration. This may deny the computer from accessing any resource over IPv6.

Unverified Router Advertisement

The router advertisement message, which we will dive into in the next section, is typically a message containing information about the router and the local network configurations.
It includes thes:

- MTU
- Prefix
- Indications of DHCPv6 services

These messages are sent periodically to the "All-Node" multicast. Nodes are expected to accept and configure these settings.

This creates a situation where attackers may spoof router advertisement messages and influence network configuration of nodes connected to the link.

Deep Dive: Router Advertisement

Packet Structure

```
Internet Control Message Protocol v6
  Type: Router Advertisement (134)
  Code: 0
  Checksum: 0xd911 [correct]
  [Checksum Status: Good]
  Cur hop limit: 255
v Flags: 0x40, Other configuration, Prf (Default
    0... .... = Managed address configuration: N
    .1.. .... = Other configuration: Set
    ..0. .... = Home Agent: Not set
    ...0 0... = Prf (Default Router Preference):
    .... .0.. = Proxy: Not set
    .... ..0. = Reserved: 0
  Router lifetime (s): 1800
  Reachable time (ms): 0
  Retrans timer (ms): 0
```

The router advertisement packet contains the same fields as any other ICMPv6 packet:

- Type
- Code
- Checksum
- Current hop limit

We are going to focus on the following fields:

Flags
The RFC states 2-bit flags that indicate whether DHCPv6 information is available.

Router lifetime
The lifetime of the associated router, which-indicates how long the router should appear in the default router list.

Reachable time
How long a node should save a cached neighbor entry.

Retrans timer
How long a node should wait before transmitting a new Neighbor solicitation message.

Packet Options
Router advertisement messages may contain a list of NDP Options, such as:

- Prefix Information
- MTU
- Router Information
- Source Link-Layer Address
- DNS

Nodes receiving a router advertisement message should act upon those options.
If needed, they should add or modify configurations.

ICMPv6 DNS Hijacking

Let's see how we can hijack a victim's default DNS resolution address. We will spoof a router advertisement message containing the 'Prefix Information' and 'RDNSS' options. The victim will then append the new IPv6 DNS to the DNS list and will auto-configure a new global IPv6 address.

This hijacking method was first published in 2011 by the InfoSec Institute, who called it the "SLAAC Attack". A few years later, in 2018, an open-source tool called mitm6 was released publicly for pen testers.

The DNS Hijacking Attack: Step-by-step
In this example I'm going to show a step-by-step example of this DNS hijacking technique, which takes advantage of the ICMPv6 protocol by using the NDP extension.

Our lab setup:

- Attacker – Ubuntu Server 20.04 VMrunning both DNS & HTTP servers
  IPv4 - 10.0.10.1/24
  IPv6 Link Local - ff80::20c:29ff:fee0:ef22
  IPv6 Global - 2001:44b8:cc00:8d63:b518:131:1111
- Victim – MacBook Air Monterey
  IPv4 - 10.0.10.34/24
  IPv6 Link Local - fe80::83d:c7c:b25d:55a7
  IPv6 Global - N/A

For this technique to work, both the attacker and the victim need to connect to the same switch.

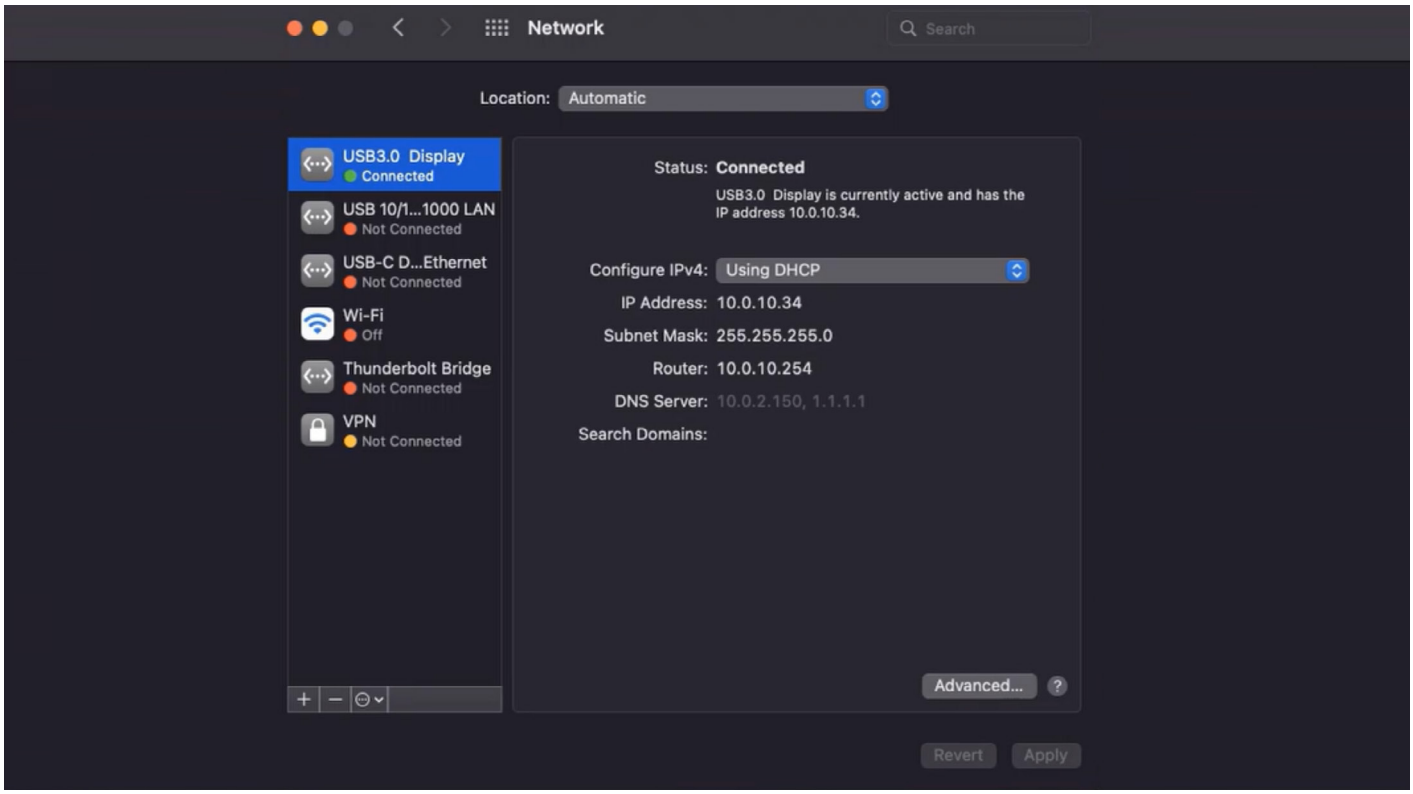
Like in most attacks, we figured we should start with Discovery.
Sending reverse lookups to the multicast was a good start.

```
Internet Protocol Version 4, Src: 10.0.10.1, Dst: 224.0.0.1
User Datagram Protocol, Src Port: 5353, Dst Port: 5353
Multicast Domain Name System (query)
   Transaction ID: 0x0000
 > Flags: 0x0100 Standard query
   Questions: 1
   Answer RRs: 0
   Authority RRs: 0
   Additional RRs: 0
 v Queries
    > 34.10.0.10.in-addr.arpa: type ANY, class IN, "QM" question
```

After a few seconds, an answer was sent back, including the AAAA record.
Now we have a potential victim.

```
Internet Protocol Version 4, Src: 10.0.10.34, Dst: 224.0.0.1
User Datagram Protocol, Src Port: 5353, Dst Port: 5353
Multicast Domain Name System (query)
   Transaction ID: 0x0000
 > Flags: 0x0000 Standard query
   Questions: 0
   Answer RRs: 2
   Authority RRs: 0
   Additional RRs: 0
 v Answers
    > Lab-Macbook: type A, class IN, addr 10.0.10.34
    > Lab-Macbook: type AAAA, class IN, addr fe80::83d:c7c:b25d:55a7
```

We switched to the victim's network configuration before taking any active poisoning steps. We can see that the victim's DNS servers list doesn't include an IPv6 address.



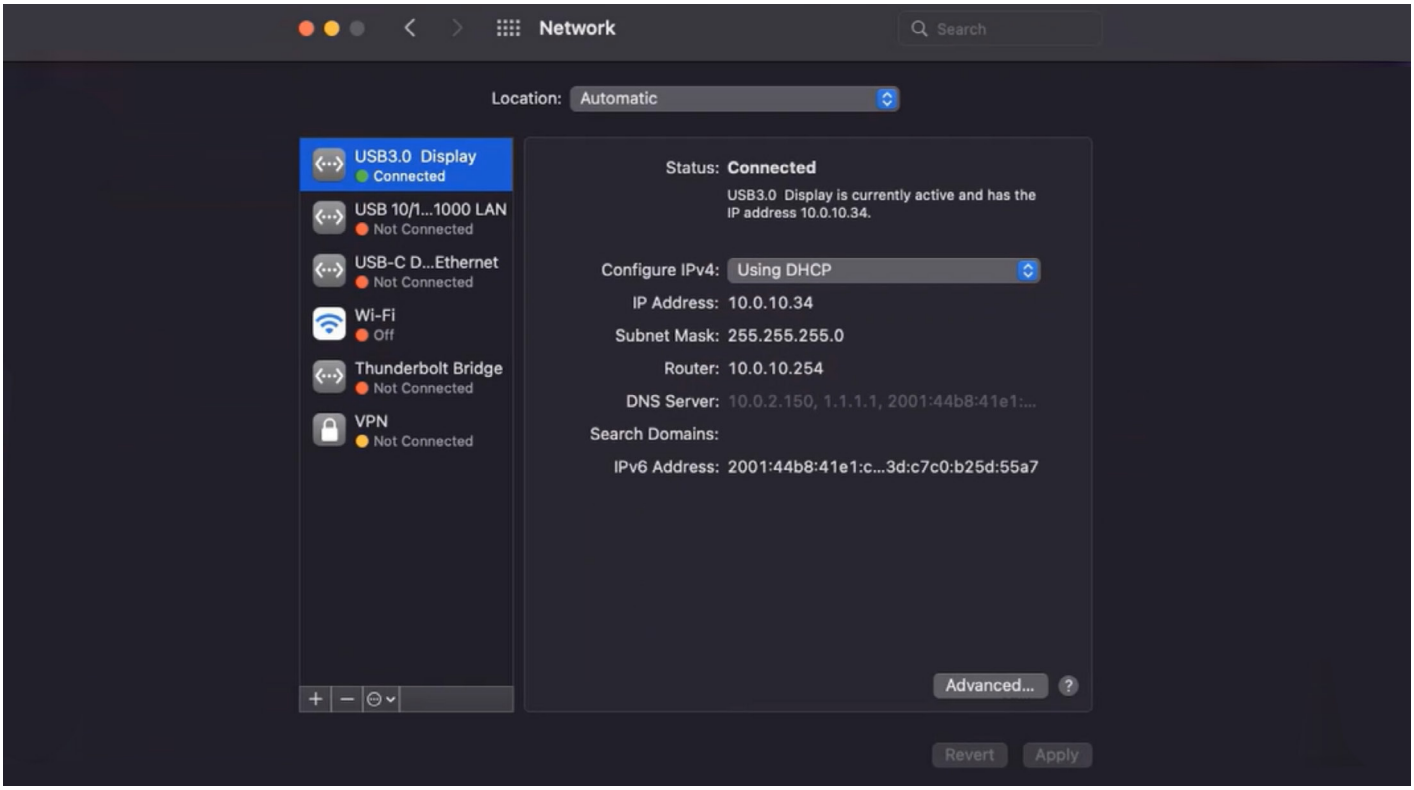
Then, we sent the router advertisement directly to our victim, including prefix information and recursive DNS server options.

If our victim receives and verifies this information, the victim will reconfigure both of those settings based on the information we sent.

```
Internet Protocol Version 6, Src: fe80::20c:29ff:fee0:ef22, Dst: fe80::83d:c7c:b25d:55a7
Internet Control Message Protocol v6
   Type: Router Advertisement (134)
   Code: 0
   Checksum: 0xbd10 [correct]
   [Checksum Status: Good]
   Cur hop limit: 0
 > Flags: 0x08, Prf (Default Router Preference): High
   Router lifetime (s): 1800
   Reachable time (ms): 0
   Retrans timer (ms): 0
 > ICMPv6 Option (Source link-layer address : 00:0c:29:82:af:28)
 > ICMPv6 Option (MTU : 1280)
 > ICMPv6 Option (Prefix information : 2001:44b8:41e1:cc00::/64)
 > ICMPv6 Option (Recursive DNS Server 2001:44b8:41e1:cc00:8d63:b518:131:1111)
```

After the poisoning action, we can see that, without any doubt, the victim has not only configured a new IPv6 DNS server, but also a global IPv6 address with the same prefix.

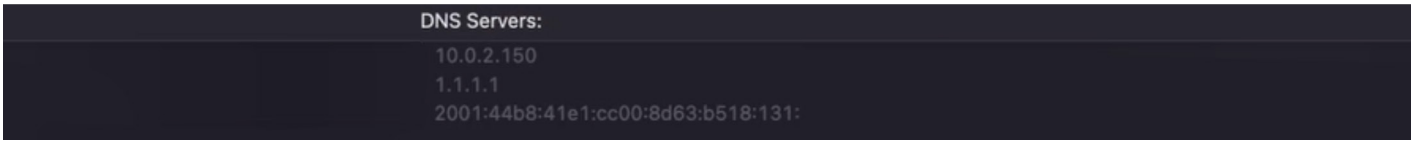
This is an indication of a successful poisoning action.



Taking a deeper look into the victim's network configuration, we can see a few interesting settings:

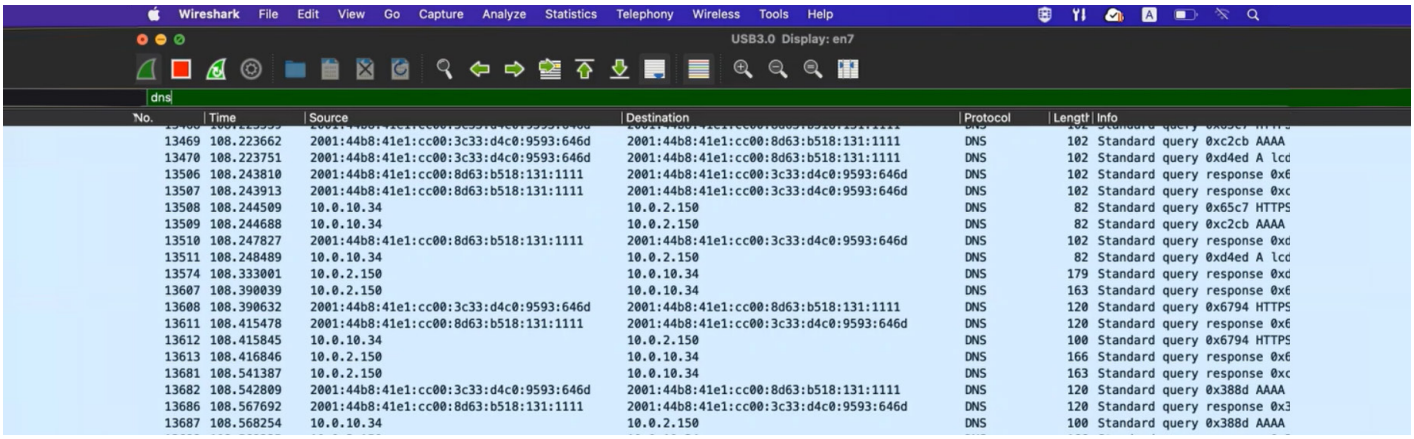
- The IPv6 Router address is configured to the attacker's link-local address.
- The victim has self-generated two global IPv6 addresses. One is based on the link-local IPv6 and the other one is the temporary address (it changes on a weekly or daily basis), which is usually used.

Switching to the DNS panel, we can see our DNS server's IPv6 address was added.



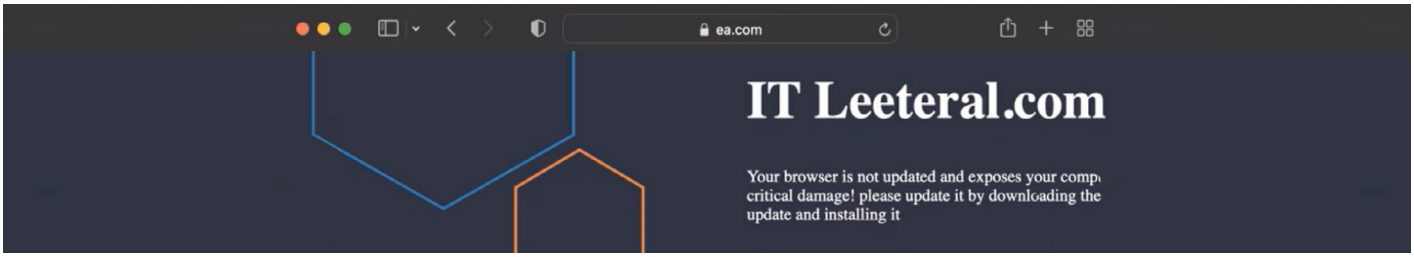
Using Wireshark, we can observe the DNS traffic from our victim's network interface.

The interesting part is that both versions of DNS (IPv4 & IPv6) are used in this case to resolve IP addresses.



When entering a legitimate URL in the Safari browser, it seems that the responses from DNS IPv6 were preferred over the responses from DNS IPv4. Our fake website is presented.

As a side note, I will mention that this type of attack can sometimes be a "race-condition" type of service. Whoever responds the fastest will take over. This could be the attacker and it could be the DNS server. In this case, we had the advantage because our attacker is one "hop" from the victim, unlike the organization's DNS server.



Why the Attack was Successful

The NDP protocol, like ARP and DHCP, has no verification of the sender's identity.
Unlike DHCP, there is no need for a "client-initiated" process. An attacker can decide to transmit router advertisements as "push" configurations and devices are supposed to accept them.

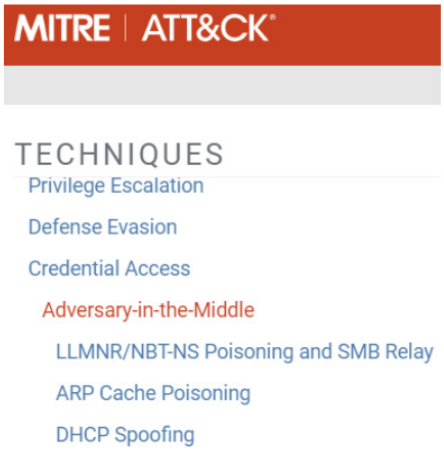
The second "problem" we take advantage of is that IPv6 is preferred by the OS over IPv4 services (in most protocols).

This means that if a computer has an IPv6 address and is accessing a resource (for example, https / dns) that supports IPv6, the IPv6 will be preferred over IPv4. A CVE regarding this issue was published in 2011, but it is classified as "DISPUTED" and hasn't received much attention.

Affected Platforms

This method affects Windows, Linux and even MacOS platforms.

MITM Mitigations



One of the most common and in-use platforms for self-validation nowadays, MITRE ATT&CK, lists various techniques for mitigating MITM attacks. This particular technique was discovered over a decade ago and a free open-source tool was published over four years ago. However, there are no references to this technique or of usages "in the wild" on the MITRE website, making it difficult to properly mitigate.

While most environments are complex, I truly recommend having a good understanding of the technologies running in your environments, so that you're able to disable any technology not in use, and apply suitable security standards for the rest.

For starters, these are a few mitigations that you can apply against MITM attacks:

Network Level

These methods are all applied at the switch level.

- **DHCP Snooping** – Defines which physical ports are trusted and are allowed to transmit DHCP offer packets.
- **Dynamic ARP Inspection** – Binds mac addresses to ipv4 addresses, which will block physical devices trying to impersonate a different ipv4 address.
- **RA Guard** specifies which port is "trusted" for transmitting router advertisement messages. This is a configuration we can set at the switch level. Most network device vendors will include this security mitigation in their arsenal.
- **Secure ND** is an extension of the ND protocol, based on new NDP options. The main difference is that it offers certificate-based authorization and a verification process. These router advertisement messages will contain additional options with certificate information.

Host Level

Disabling ICMP redirects

Final Notes

As the internet - and technology - changes and develops, it creates wonderful new capabilities, but also, new attack vectors. In my opinion, everything starts with awareness.

By continuing to research and read about the latest vulnerabilities and the latest attacks, security teams can put the right controls in place, while continuing to test and validate to make sure environments remain safe.

Want to keep learning? Read more research from Pentera Labs.

Appendix

RFC 4957 - https://datatracker.ietf.org/doc/html/rfc4957
MS-LLTD - https://winprotocoldoc.blob.core.windows.net/productionwindowsarchives/MS-LLTD/%5bMS-LLTD%5d.pdf
RFC 6762 - https://www.rfc-editor.org/rfc/rfc6762.html
RFC 4861 - https://datatracker.ietf.org/doc/html/rfc4861
RFC 2463 - https://www.rfc-editor.org/rfc/rfc2463
Airplay service discovery - https://openairplay.github.io/airplay-spec/service_discovery.html
Infosec "SLAAC Attack" - https://resources.infosecinstitute.com/topic/slaac-attack/
NDP vulnerabilities - https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8022867

About the author

Yotam Mazurik is a Senior Security Researcher at Pentera. Prior to joining Pentera, Yotam served in the Israeli Navy's cyber unit.

Reach out to us with any questions about the research at labs@pentera.io.

About Pentera

PENTERA

Pentera is the category leader for Automated Security Validation, allowing every organization to test with ease the integrity of all cybersecurity layers, unfolding true, current security exposures at any moment, at any scale. Thousands of security professionals and service providers around the world use Pentera to guide remediation and close security gaps before they are exploited.

For more info, visit: pentera.io

PENTERA | Pentera Labs™ Research Series

Death by Default: Neglected network protocols you should know