NTP Over IPv6: A New Look at an Old Protocol

Author: Jacob Magdziarz, jacob.magdziarz@student.sans.edu
Advisor: Johannes Ullrich

Accepted: 20 August 2023

Abstract

The Network Time Protocol (NTP), described in RFC 5905, is an important and often hidden component of modern computer systems, providing accurate time synchronization for a wide range of applications (Mills, 2010, p. 1). With the increasing adoption of Internet Protocol version 6 (IPv6), there is a growing need to understand the security implications of using NTP over IPv6. The results of this research indicate that public NTP servers, even within well-known public pools, should not be implicitly trusted. The use of untrusted NTP servers could lead to information disclosure or targeted attacks attempting to disrupt accurate timekeeping in an environment. However, there are already methods to better secure NTP within an environment and reduce the risk exposure of NTP.

1. Introduction

The Network Time Protocol (NTP) is a critical component of modern computer systems, providing accurate time synchronization for a wide range of applications. There are several existing attacks against the NTP protocol and different implementations of NTP servers (Malhotra, 2015, p. 1). With the increasing adoption of Internet Protocol version 6 (IPv6), there is a growing need to understand the security implications of using NTP over this protocol. IPv6 provides several advantages over IPv4, including increased address space and improved packet handling, but it also introduces new security challenges that must be carefully considered (CISA, 2022). This research examines the security implications of using NTP over IPv6, including potential vulnerabilities and attacks that may be leveraged against NTP servers and clients. We also explore strategies for mitigating these risks, such as secure configuration options and using new strategies for authenticating NTP. By addressing these issues, we aim to provide a comprehensive understanding of the security implications of using NTP over IPv6 and to enable organizations to make informed decisions when deploying NTP in IPv6 environments.

1.1. IPv6

The Internet Protocol version 6 (IPv6) was designed to replace the existing IPv4 protocol, which was first deployed in the early 1980s. IPv6 was designed to overcome the limitations of IPv4, which include a limited number of available addresses and a lack of support for new features. IPv6 also sought to address some of the security concerns that were not considered when developing the IPv4 protocol (CISA, 2022).

Despite being developed in the late 1990s the adoption of IPv6 has been slow. There have been efforts by some organizations, including the United States government, to increase the adoption of IPv6 on sensitive systems and networks, citing enhanced security as one of the primary drivers (CISA, 2022).

While IPv6 will have many advantages, misconfigurations of IPv6 can introduce new security concerns and a misunderstanding of the protocol can cause adopters to assume a “security by obscurity” approach when assigning and using public IPv6 addresses.

1.2. NTP

The Network Time Protocol (NTP) was first introduced in 1985 to synchronize the clocks of computer systems on the ARPANET, the precursor to the modern Internet. Since the 1988 publication of the NTPv1 protocol in RFC 958, three new versions have been released (Mills, 1985; Stenn, 2016). During that time several vulnerabilities have been discovered in the NTP protocol, including CVE-2017-6460 which can result in remote code execution on misconfigured NTP servers. Replay and spoofing attacks also exist due to a lack of default authentication. Amplification DoS attacks have made up the majority of misuse of the NTP protocol and gotten the most attention (Malhotra, 2015, p. 16).

NTP has been used as a method of discovering active public IPv6 addresses by the internet scanning service Shodan, but the prevalence of that activity by other actors is not well-researched (Ulrich, 2016).

2. Research Method

To gain a more complete understanding of the NTP ecosystem measurements were taken from both an NTP client and an NTP server. The client's role was to reach out to public IPv6 NTP servers, collect NTP responses, and monitor all incoming and outgoing traffic. The server was registered with pool.ntp.org and monitored all incoming and outgoing traffic. Information from Shodan and NTP Pool Project was used to identify public NTP servers.

2.1. Discovery

This research did not involve any active scanning or outbound connections other than legitimate NTP traffic. To gather addresses of servers providing NTP over IPv6 two data sources were used: Shodan and NTP Pool Project.

Shodan was queried for any servers with an IPv6 address that appeared to be running NTP or had port 123 open using the following query: The results were downloaded as JSON and used in further scripts.

NTP Pool Project maintains a list of servers that can be used by clients or other servers that are broken down into stratum. Stratum 1 servers are a primary time source and are not getting time information from any other servers. The stratum of other servers is determined by the server they receive time from; a server using a stratum 1 time source would be a stratum 2 server. A script was used to scrape all server information that was available on the server listings. NTP Pool Project also makes its pool monitoring information available, including both active and inactive servers. Another script was used to query for monitoring data available from any of the IPv6 addresses that had been discovered.

2.2. NTP Pool Server

A virtual server, referred to as the NTP Pool node, was created via a DigitalOcean Droplet and configured with a new IPv6 address. It was running Ubuntu 22.04 (LTS) x64 and NTP was installed and configured following the guidance from NTP Pool Project (Configuration, 2023). Once the NTP server was running correctly and accessible over IPv6, it was given a public DNS record and registered as an NTP server with NTP Pool Project. Anyone can register a server by providing an IP address or hostname. At the time the server was registered it did not appear possible to register a server by providing only an IPv6 address, however, it was possible by providing a hostname that resolved to an IPv6 address.

2.3. NTP Client

A virtual server, referred to as the client node, was created via a DigitalOcean Droplet and configured with a new IPv6 address. It was running Ubuntu 22.04 (LTS) x64 and configured to capture all IPv6 traffic. The client machine was used to run Python scripts sending NTP requests to the discovered IPv6 servers and to collect the NTP responses. Traffic sent from the machine's IPv6 address was limited as much as possible to avoid it being exposed other than through sending NTP requests.

2.4. Packet Capture and Analysis

A tmux session was used to run tcpdump and capture all IPv6 traffic. The following command was used to collect the traffic and rotate the capture file every 12 hours: `tcpdump -i eth0 ip6 -w ipv6_%Y-%m-%d_%H.%M.%S.pcap -G 43200 --print -B 4096`

Full packet captures of IPv4 traffic were also collected, excluding port 22 which was used for administration and data retrieval. This was the most straightforward option for excluding my own traffic from the data and was not believed to impact the quality of the data collected as the focus of the research was IPv6. The files from each system were later combined and analyzed together.

3. Findings and Discussion

3.1. State of the NTP Ecosystem

In June 2023 the NTP Pool Project listed 4416 active servers in the pool, of which 1627 were available over IPv6.

IPv4

There are 3045 active servers in this zone.

3042 (+3) active 1 day ago
3062 (−17) active 7 days ago
3061 (−16) active 14 days ago
3035 (+10) active 60 days ago
3093 (−48) active 180 days ago
3170 (−125) active 1 year ago
3008 (+37) active 3 years ago
2988 (+57) active 6 years ago

IPv6

There are 1627 active servers in this zone.

1628 (−1) active 1 day ago
1630 (−3) active 7 days ago
1623 (+4) active 14 days ago
1625 (+2) active 60 days ago
1607 (+20) active 180 days ago
1524 (+103) active 1 year ago
1388 (+239) active 3 years ago
1152 (+475) active 6 years ago

Global — pool.ntp.org (4416)

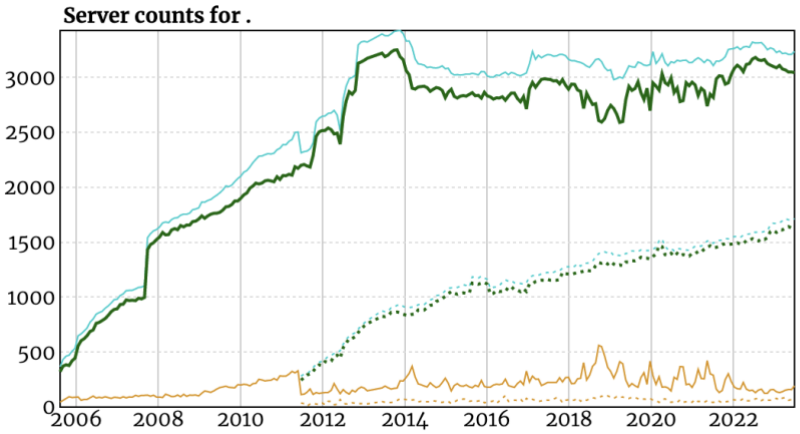

Figure 1. Chart showing servers active in pool.ntp.org over 6 years

The Shodan results included 34,310 servers with IPv6 addresses that also had port 123 open. This represented a sharp decrease from the peak of over 60,000 servers in late 2022, although no specific reasons were identified that would have caused this decrease.

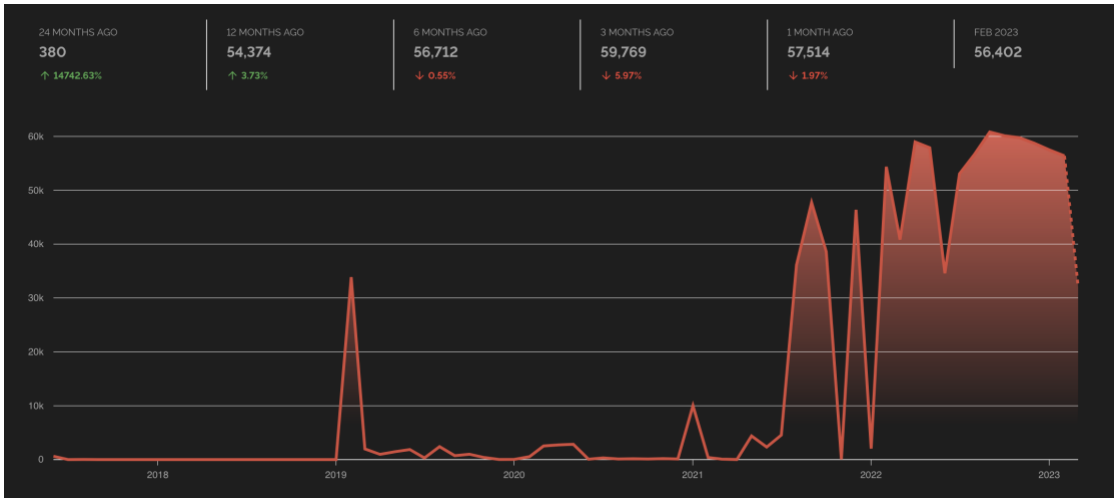

Figure 2. Chart of NTP servers in Shodan since 2018

Geographically, the majority of IPv6 NTP servers were using IP addresses associated with Germany, followed up by the United States and the UK.

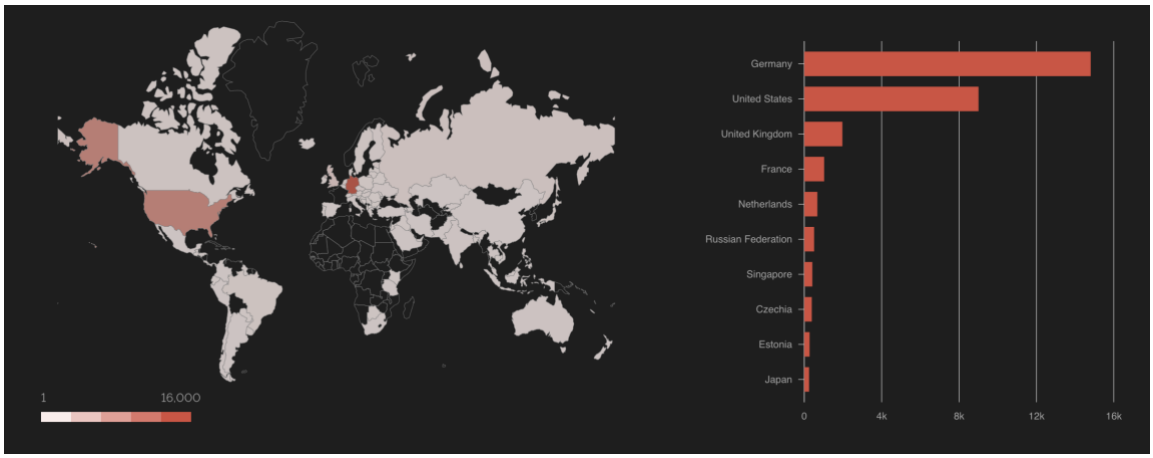

Figure 3. Chart showing geographic distribution of discovered NTP servers

3.2. NTP Pool Analysis

Data was collected by the server joined to the NTP pool February 21st – May 1st, which included 18 days of full packet captures before the server was registered with pool.ntp.org to establish an activity baseline. Within a few hours of joining the pool, the server began to receive requests from the monitoring servers used by NTP Pool Project to score servers in the pool. Only servers that meet a score threshold of 10 points are given NTP requests from the pool. Servers start with a score of -5 which increases each time a

monitoring server is able to connect and gets an accurate time response. The research server achieved the score threshold in about 12 hours and began receiving NTP requests from the pool.

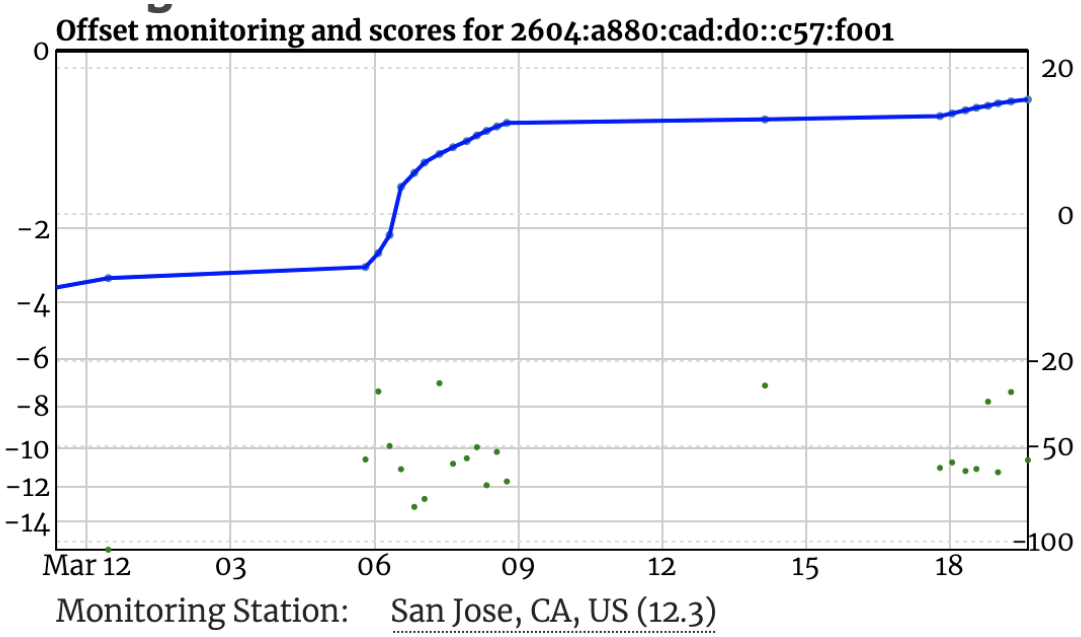

Figure 4. Graph showing NTP Pool score over March 12, 2023

Servers in pool.ntp.org can configure the amount of bandwidth they have available and the pool will adjust the amount of traffic sent based on that setting. The research server was configured for 512 Kbit, the lowest value, to help ensure there would be minimal packet loss at peak traffic. Over the 3 months of data collection, a total of 245,172 NTP requests were received from 21,832 unique IPv6 addresses.

3.3. NTP Query Modes

NTP operates in several modes, which determine the role and behavior of an NTP server or client (Haberman, 2022, Section 1). For the purposes of this research, four of the modes are significant:

- Mode 1 (Symmetric Active): In Mode 1, two NTP nodes exchange time information with each other. Both nodes act as peers and actively participate in time synchronization.

- Mode 3 (Client): In Mode 3, an NTP client requests time information from one or more NTP servers. The client then adjusts its local clock based on the received time.
- Mode 4 (Server): Mode 4 is used by NTP servers to respond to time requests from clients.
- Mode 6 (Control): Mode 6 is used for monitoring and controlling NTP servers and clients. It is intended to allow administrators to retrieve operational data and statistics from NTP devices, however, it has been abused by attackers for DoS and other attacks (Malhotra, 2015, p. 16).

The majority of NTP packets received by the pool server, almost 91%, were Mode 3 queries, which is to be expected for an NTP server. 2,207 packets were responses to time synchronization requests made by the research server, and no gratuitous responses were observed.

A further 48 mode 6 requests were received, all of which were from IPv6 addresses in the 2001:470:1:332::0/64 or 2001:470:1:c84::0/64 ranges and appeared to be probes looking for vulnerable NTP servers. Hosts from these two IP blocks were doing port scans throughout the data collection period, discovering the NTP server only a couple of days after it was given a public IPv6 address, and well before it was joined to the NTP pool. These IP ranges are further analyzed in Section 3.5.

3.4. Reference IDs

One interesting piece of information was the “Reference ID” field included in many of the NTP requests and responses. The primary use of the Reference ID (refid) is to prevent timing loops by letting other NTP nodes that query a server know which time source was used by the server. If two servers, A and B, are exchanging time information and server B follows server A as its time source, server B’s “refid” will be calculated from Server A’s IP address. That way, if server A queries B it would see its own “refid” listed in server B’s response and know that it cannot use server B as a time source because it would cause a timing loop (Stenn, 2016. pp. 2-3).

For stratum one servers, also called a reference clock because it is a primary time source referenced by higher stratum servers, the “refid” should be a unique string maintained by IANA.

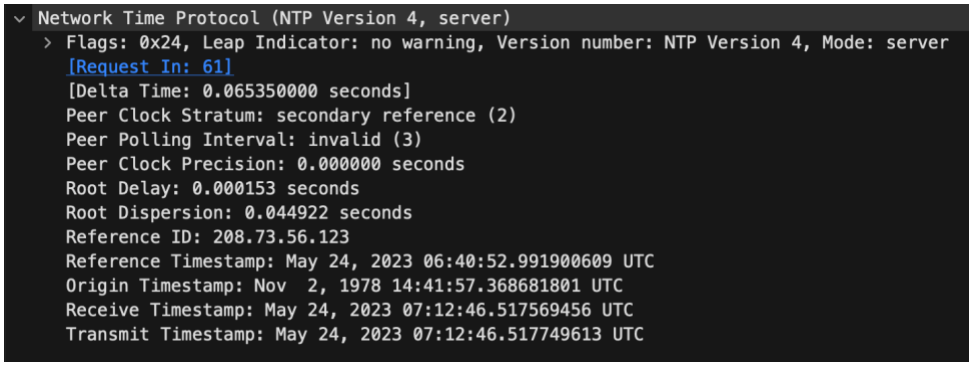

Figure 5. Example of an NTPv4 Response with Reference ID

NTP messages can also claim to be stratum zero, which causes the Reference ID to be recognized as an ASCII code. Originally called a “kiss code”, used as a way to instruct NTP clients to either rate limit their requests or stop sending them entirely (a “kiss-of-death”), they are also used for debugging or to help identify a type of server. For example, a stratum 1 sever could send a Reference ID of “GPS” in stratum zero messages to indicate it is using a GPS clock as the time source. Any Reference ID starting with the ASCII character “X” is considered experimental. (Mills, 2010. p. 22).

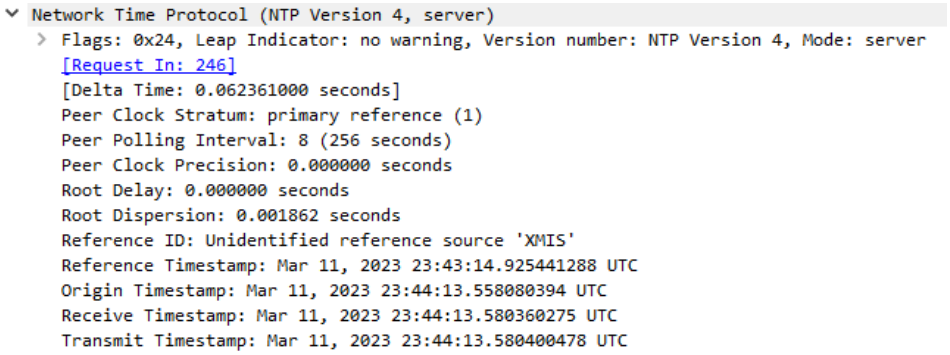

Figure 6. NTP message with the “XMIS” Reference ID

Analyzing the Reference IDs returned by the client node’s queries, 25 potential ASCII values were determined. Those values and the number of unique IPv6 addresses that returned them are shown in Figure 7.

Reference ID	# Unique IPs
INIT	297
BUNP	33
LOCL	30
GPS	24
PPS	24
STEP	13
SHM	9
VMTP	9
GPPS	3
MRS	3
NIST	2
AUTH	2
PZF	2
RATE	2
PTP	2
NICT	2
BDS	1
PTB	1
XMIS	1
ONBR	1
GOOG	1
NULL	1
MBG	1

Figure 7. ASCII Reference IDs by Unique IP

For a server using an IPv6 address, its Reference ID is supposed to be the md5 hash of the first four octets of the address (Stenn, 2016, p. 3). However, a significant number of Reference IDs for NTP servers reached over IPv6 appeared to be valid IPv4 addresses. It is worth noting that Wireshark seems to decode and display every Reference ID that is not a kiss code as an IPv4 address, even though that is not necessarily the meaning of the value. In at least some cases, according to WHOIS records, this was an IPv4 address used by IANA's 6to4 Anycast Relay service. That service is used as a bridge between IPv4 networks using the 6to4 protocol and native IPv6 networks, however, there are some security concerns with 6to4 that have still not been fully addressed, including information disclosure and potential DoS (Savola, 2004, p. 3). The security implications specific to using NTP over 6to4 networks were not evaluated by this research and could be valuable to investigate further.

3.5. Scanning Traffic

For this research, suspected scanning traffic was considered any traffic from IPv6 clients that was not using the NTP, DNS, or ICMPv6 protocols and that did not have port 80 or 443 as the source or destination. While this may have excluded some legitimate scanning traffic, it provided a good overall view of unusual traffic that would be likely to indicate scanning.

Scanning traffic was observed throughout the collection period and did not seem to be affected by the server joining the NTP Pool. The traffic came from a limited number of IPv6 ranges, shown in Figure 8, with each IP only sending a few packets. There was an average rate of one packet per second that remained consistent throughout the data collection.

The only scanning traffic observed that was not from those ranges was from 2a05:d01c:b43:8a10:c875:1594:97d2:7e9c. This single IP performed what appeared to be a “top 100” port scan lasting 7 minutes and was not seen in the logs again. No NTP traffic was sent to or from that IP.

The client node connecting to public NTP servers saw 6,372 suspected scan packets of the total 352,057 packets collected, making up 1.8% of the total traffic. Of the 34,310 servers discovered through Shodan and subsequently queried, 33,336 servers responded with NTP messages. However, none of the observed scanning traffic came from an address that was sent an NTP query by the client node. Of the suspected scan traffic, 73% came from the same subnets regularly scanning the NTP Pool node except for 2001:41d0:403:1f43::/64 which was not observed. In total, 1,093 unique IPv6 addresses sent suspected scan traffic.

IPv6 Range	NTP Pool node (prior to joining pool)	NTP Pool node (after joining pool)	Client node
2001:470:1:332::0/64	540	585	923
2001:470:1:c84::0/64	927	2311	1464
2001:41d0:403:1f43::/64	412	648	0
2a06:4880::/32	1344	3803	1988
2a05:d01c:b43:8a10:c875:1594:97d2:7e9c	0	333	0
2401:25c0:0:110:ffff:ffff::/96	0	0	294
2607:ff10:c8:594::/112	0	0	1578
Other IPv6 Addresses	0	0	304

Figure 8. Suspected scanning packets by IPv6 range

The majority of the 12,310 scanning packets were raw TCP or UDP requests, with 12,072 TCP packets and only 238 UDP packets. Protocol analysis was also performed to determine the Application Layer protocols observed. Notably, no NTP traffic was observed other than replies to the client node's requests. An additional 800 SSH packets were observed that are not shown in Figure 9.

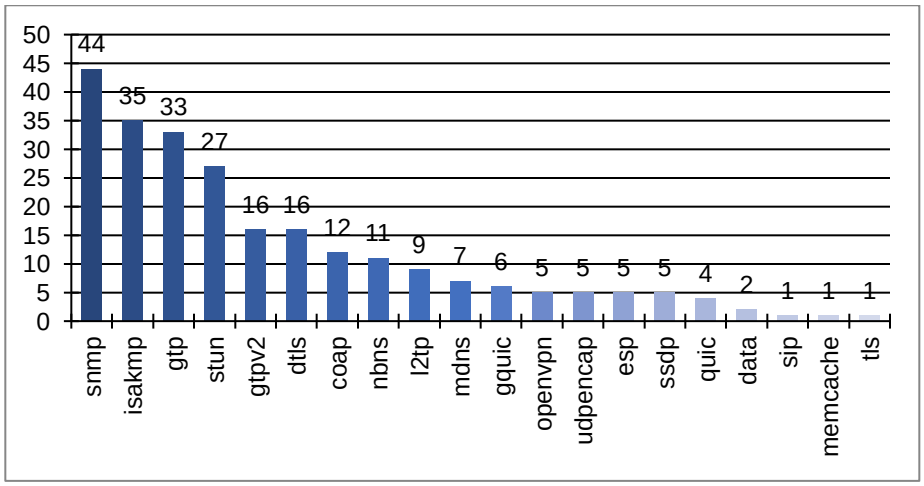

Figure 9. Bar chart of packets by Application Layer protocol

The two servers had similar exposure profiles other than the active probing done by the NTP client node versus the NTP Pool node. The NTP Pool node was also accessible for significantly longer, had its IPv6 address publicly available through the pool, and had an active DNS record. All of these factors would have seemed to make the chance of exposure for the NTP Pool node higher had the active probing from the client node not contributed to the volume.

This indicates that the active probing resulted in an increased amount of scanning. Looking at the amount of suspected scanning traffic over time, there is a peak while the NTP queries were actively running and a gradual decrease of the average until the captures were stopped.

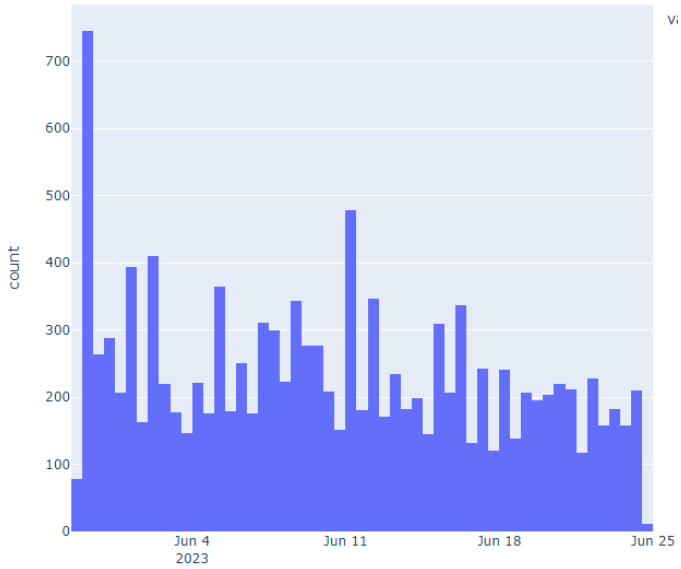

Figure 10. Bar chart showing the number of packets per day

There were also indications that more targeted scanning was triggered by the NTP client connections. While the majority of scanning traffic against both servers came from the IPv6 ranges in Figure 8, and remained fairly constant over the observation period, there were notable outliers that correlated to the active NTP probes. There were 304 suspected scanning packets that came from other IPv6 ranges, all of which were received on 2023-05-30 while the probes were actively running.

Traffic from those ranges was not observed at any other time. Those servers also never received an NTP request or any other traffic from the client node, making it difficult to determine a causal relationship between the NTP probing and the increased scanning traffic.

3.6. Security Implications

There have been some security concerns expressed by the NTP community, and an IETF draft exists that specifically seeks to address some of these concerns (Stenn, 2016). The most significant area of concern is that the Reference ID can be used to

identify IP addresses of servers that were used as a reference server. This can result in information disclosure, including internal IP addresses if the public server had used a private reference clock.

For example, server A could query server B and would receive the Reference ID of server B's time source (server C) in server B's response. If server C is an IPv4 server the Reference ID is likely already that IP address. Even if server C is over IPv6 and uses the expected hash value, it is still reasonably possible to determine the IP address by creating a dictionary of md5 hashes for public NTP servers. This information can be used to target and attempt to disrupt the time server by sending spoofed packets to server B and server C (Stenn, 2016, p. 29). This is similar to the attack described in CVE-2015-8138.

This research also indicates that there is some risk of inviting unwanted scanning traffic by reaching out to public NTP servers, however, it isn't clear that the NTP protocol was relevant. No data was collected for this research that indicates whether connecting to NTP servers invites more or less traffic than connections to another protocol or port.

4. Recommendations and Implications

4.1. Risks and Mitigations

The research shows that the main recommendation for reducing risk is to use an internal NTP server and ensure that all of the systems in the environment are configured to use it. The internal server can be configured to get time information from trusted NTP servers using a secure transport method, such as Cloudflare's NTS servers. Alternatively, a primary time source such as a GPS clock can be used internal to an environment. Regardless of the choice of time source, it is recommended to use some form of secure authentication at all levels to prevent possible manipulation of time data.

If an NTP server will be publicly accessible, care should be taken to ensure that it is not leaking information about internal or external IP address space through the Reference ID. If a NTP server is using an IPv4 time source, that IPv4 address will be sent

as the Reference ID by default. However, any manipulation of the Reference ID must be done with care as it circumvents the intended usage of preventing reference loops.

4.2. Proposals for Securing NTP

There are existing or proposed methods to make NTP more secure. The NTPv4 protocol already has support for authentication, but it is limited to using symmetric keys which can make an implementation difficult to manage at scale. Network Time Security (NTS) is an extension of the existing NTPv4 protocol that adds security features and is seeing some adoption by service providers (Lichvar, 2020).

Network Time Security (NTS) uses digital signatures to authenticate the time information and prevent unauthorized modification or tampering of the time data. It uses public key infrastructure (PKI) to verify the identity of the time source and ensure that the time data comes from a trusted server. NTS also provides features such as access control and encryption to ensure the confidentiality and integrity of the time data (Franke, 2020).

All of this helps to ensure that the time information is accurate and comes from a trusted source. NTS is not yet widely supported, but some trusted providers have deployed public NTP servers that support NTS including Cloudflare and Netnod (Lichvar, 2020).

5. Conclusion

The conclusion of this research is that the risks of NTP over IPv6 do not greatly differ from the known risks of using NTP over IPv4. IPv6. NTP servers do not appear to be broadly used as a mechanism for discovering IPv6 targets for scanning, although there was a notable increase in scanning traffic while actively probing public NTP servers. The volume of scanning traffic that appeared unrelated to the NTP requests was far higher.

It was very simple to join an NTP server to the public and widely-used pool.ntp.org NTP Pool. Anyone who wanted to use the pool as a way to collect potential IPv6 targets could do so by remaining a server in good standing and passively collecting data. That data could be used to target those NTP servers directly, or higher stratum

servers revealed by the information disclosure flaw in NTP Reference IDs. All of that data can be used to manipulate time information through existing attack vectors.

Ultimately, the best recommendation is to use existing mechanisms to authenticate NTP requests and responses, such as Network Time Security (NTS), and to ensure any time sources used are trusted. This is especially important if an internal NTP server is retrieving time information from a public reference clock. In order to eliminate using any external time information, a primary time source such as GPS can be used as a reference clock to internal servers. If possible, all systems in an internal environment should be configured to use an authenticated internal NTP server.

References

CISA (2022, January*). IPv6 CONSIDERATIONS FOR TIC 3.0.* Cybersecurity & Infrastructure Security Agency. Retrieved June 20, 2023 from, https://www.cisa.gov/sites/default/files/2023-02/cisa_ipv6_considerations_for_tic_3.0.pdf

Configuration recommendations for servers joining the pool. NTP Pool Project. (n.d.). Retrieved January 20, 2023, from https://www.ntppool.org/join/configuration.html

Franke, D. et al (2020, September). *Network Time Security for the Network Time Protocol*. Internet Engineering Task Force (IETF). Retrieved January 10, 2023, from https://datatracker.ietf.org/doc/html/rfc8915

Haberman, B., et al (2022, November). *RFC 9327 Control Messages Protocol for Use with Network Time Protocol Version 4*. Internet Engineering Task Force (IETF). Retrieved June 25, 2023, from https://datatracker.ietf.org/doc/rfc9327/

Lichvar, M. (2020, October 23). *Secure NTP with NTS*. Fedora Magazine. https://fedoramagazine.org/secure-ntp-with-nts/

Malhotra, A. et al. (2015, October 21). *Attacking the Network Time Protocol*. Boston University, Department of Computer Science. Retrieved January 10, 2023 from, https://eprint.iacr.org/2015/1020.pdf

Mills, D.L., et al (1985, September). *Network Time Protocol (NTP).* Internet Engineering Task Force (IETF). Retrieved June 20, 2023, from https://datatracker.ietf.org/doc/rfc958/

Mills, D.L., et al (2010, June). Network Time Protocol Version 4: Protocol and Algorithms Specification. Internet Engineering Task Force (IETF). Retrieved June 23, 2023, from https://datatracker.ietf.org/doc/html/rfc5905

Reilly, D., Ed., et al (2019, July). Network Time Protocol Best Current Practices. Internet Engineering Task Force (IETF). Retrieved January 20, 2023, from https://datatracker.ietf.org/doc/html/rfc8633

Savola, P. et. al. (2004). RFC 3964: Security Considerations for 6to4. The Internet Society. Retrieved from https://www.ietf.org/rfc/rfc3964.txt

Sheehy, J. (2015). There Is No Now. *Communications of the ACM*, *58*(5), 36–41. https://doi.org/10.1145/2733108

Stenn, H. et al (2016). Network Time Protocol REFID Updates. Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/draft-ietf-ntp-refid-updates-00

Ullrich, J. (2016, February 16). *Targeted IPv6 Scans Using pool.ntp.org* . SANS Internet Storm Center. Retrieved January 16, 2023, from https://isc.sans.edu/diary/20681