Summary: Network Time Protocol (NTP) is an established internet protocol that has been widely used for clock synchronization between computers and time-critical devices over packet-switched networks. It is a bidirectional protocol that allows network nodes to send timing information in a hierarchical structure. In the US Department of Energy's national energy delivery mission, NTP has been widely deployed for keeping the power grid devices in sync for safe and efficient electricity delivery. ORNL's Center for Alternative Synchronization and Timing (CAST) aims to provide a terrestrial timing infrastructure alternative for next-generation power delivery by using an effective combination of NTP with high-precision Precision Time Protocol (PTP). In CAST, whereas PTP carries the bulk of long-distance terrestrial synchronization transmission, both NTP and PTP can be effective means for enterprise internal time synchronization subscriptions. This bulletin examines various tools and methods that can be used to monitor multiple aspects of NTP operation and its performance.

The Center for Alternative Synchronization and Timing (CAST) views Precision Time Protocol (PTP) and Network Time Protocol (NTP) as critical elements of the grid, from generation to the grid edge. This tech bulletin discusses the operation of NTP. The document shows not only commercial but also open-source applications. In some cases, the commercial applications may already have existing feature sets to secure and monitor NTP streams. From a security standpoint, using subscription NTP is risky unless the open bidirectional ports on local firewalls and routers are properly monitored and controlled.

NTP MONITORING CONTEXT

Modern power-delivery infrastructures involve a multitude of subsystems, controllers, relays, and sensors, all requiring highly precise timing synchronization under the operational technology (OT) context for safe and efficient operation. For years, the default time synchronization protocol has been NTP, which uses stratum levels for its subscription architecture. Stratum-1 timeservers are the primary network time standard; lower-level stratum timeservers receive their timing data from upper-level stratum servers and distribute it accordingly. NTP stratums are limited to 16, and all NTP packet communication must traverse User Datagram Protocol (UDP) port 123.

NTP can achieve an accuracy level in the millisecond range, but certain factors, such as the requirement of privileged bidirectional firewall UDP port 123 access, could affect this accuracy and overall protocol security. For example, physical distance, asymmetric networks, and network congestion can all add to timing service delivery delays, which could then lead to phase errors on distributed devices and induce network synchronization control problems. Various timing discrepancies could also be created, hindering NTP operations. For example, time offset between the NTP client and NTP server may be too large, making

synchronization take too long. If a server were isolated for too long and not connected through a tiered stratum to sync, then the server time might have drifted and would no longer match an authoritative time (e.g., National Institute of Standards and Technology [NIST]). Jitter could be generated from network congestion and cause delays. Network irregularities such as congestion, asymmetric network delays, or software failures can sometimes cause an NTP host to be marked as a "false-ticker." Consequently, a host could also declare "no NTP peers" if all its defined peers are marked as false-tickers, thus losing the ability to synchronize [1]. All these anomalies suggest that NTP monitoring should be a critical and integral part of healthy timing infrastructure operations to prevent faults, delays, and malicious attacks.

NTP monitoring can be segmented into three general areas: NTP server monitoring, NTP traffic monitoring, and monitoring data analytics.

- NTP server monitoring primarily focuses on the availability, dependency, and operational status of the server. This type of monitoring is particularly applicable in a closed-loop subscription architecture, in which NTP servers are located and operated within the enterprise, with all the timing subscriptions internally provisioned. NTP server monitoring can provide a window to trigger NTP server adjustment/maintenance/repair actions when precursors of NTP operational anomalies are observed. However, for the case of open-loop architecture, in which most subscriptions are connected to external NTP servers, the purpose of server monitoring is largely restricted to server availability and accuracy.
- NTP traffic monitoring provides a holistic perspective of the NTP subscription network operation status in the context of overall combined timing traffic flow. Thus, it provides a realistic performance index for system-wide tuning, troubleshooting, and repair. All NTP traffic goes through UDP port 123 and is traceable via port and/or known NTP server IP addresses. Traffic monitoring can also be helpful in the open-loop architecture where all NTP connections must go through firewall bi-directional access rules to communicate with outside NTP servers. These situations create a potential choke point (and an attack surface) depending on traffic volume, hence the criticality of traffic monitoring.
- NTP monitoring data analytics can be done either online or offline with the help of advanced data analytic tools—potentially with the aid of AI and ML—to detect operational anomalies such as significant NTP traffic pattern and volume deviations, incorrect packet sequences, or the infamous "1900" erroneous data problem [2]. Different approaches to NTP monitoring data analytics could be taken, including identifying malicious or benign fault/attack signatures or detecting operational anomalies that deviate from normal operational profiles. Many techniques, such as ML or statistical analysis could be used to establish such normal NTP operational profiles and to recognize significant irregularities.

NTP SERVER MONITORING

Many aspects of human society operations are based on accurate timing and correct chronological event ordering. Precise timestamps are crucial in scenarios such as code execution or debugging sequencing in software engineering, remote device synchronization for power delivery infrastructure, forensic events correlation in criminal proceedings, and dispute resolution of time-critical business transactions. In most cases, timestamps are sourced from the underlying computer systems, which use NTP. Consequently, NTP server monitoring becomes a critical function to ensure high availability and integrity of timing data.

A wide selection of tools can be used for NTP server monitoring. They can be largely divided into two categories: commercial products and public-domain (or open-source) utilities. Several of these public-domain utilities come with different NTP reference implementations. Commercial products are mostly general-purpose, multifunctional enterprise infrastructure monitoring tools that can, when properly configured, also monitor NTP servers. Among a wide selection of commercial and public tools, the following section explores a limited subset to exemplify the different facets of NTP server monitoring.

Public-domain and open-source tools

At the operating system level, various NTP implementations exist, and they come with monitoring and logging utilities useful for NTP monitoring.

Ntpd is an open-source operating system daemon implementation of NTP. The ntpd daemon serves to maintain computer system time in sync with standard time provided by the NTP server. Ntpd comes with a filegen utility to define and enable logging of ntpd operation parameters' statistics. These parameters include clock, crypto, clock discipline loop, peer status, protocol events, timestamp, system, and process time. The logs generated contain crucial data for NTP monitoring. Closely related to ntpd, NTPsec[1] is a 2016 fork implementation of the original NTP to address several security compromises in 2014. NTPsec includes a collection of public utilities useful for NTP monitoring, logging, and management, and it removes some insecure features and obsolete Linux hardware supports.

Many of the commercial NTP monitoring offerings are built on top of these primitive utilities. Among them, ntpq is a utility program that queries NTP servers for operation status.[2] Its protocol uses the Mode 6 (report request) control message, originally defined in NTPV3 (RFC 1305 Appendix B) but also applies in NTPV4, to query NTP servers about their ntpd system daemon operation and performance status. Ntpq runs in both interactive mode and command-line mode. When an ntpq request is sent, it will time out with an error if a reply is not received from the server within a predefined time.

[1] https://www.ntpsec.org/
[2] https://support.ntp.org/Support/MonitoringAndControllingNTP

Ntpmon is another public real-time status monitor utility for NTP. It presents similar information as ntpq. It has a multiwindow display that includes a status summary bar. Ntpmon starts with per-second updates and then adaptively adjusts the rate to twice the shortest pooling rate, providing timely status updates. It collects and displays information such as status, NTP server name/IP, association ID, stratum, server type, last packet time, poll interval, reach shift register, roundtrip delay, offset, and jitter.

Beyond ntpd and NTPsec implementations, open-source infrastructure monitoring tools such as Munin[3] can also be applied for NTP monitoring. Munin is an open-source software application for general system, network, and infrastructure monitoring. Implemented in Perl and capable of producing graphical outputs accessible online, it has more than 500 plugins for diverse infrastructure monitoring applications. Some of these are pertinent to NTP monitoring, and they can watch over NTP parameters such as drift values, kernel's phase-locked loop frequency tolerance for the NTP status, queries handling, or peers offset values.

For Windows systems, Microsoft's W32Time time service synchronizes all computer times via Active Directory Domain service. The w32tm command is used to configure its settings. The same command is also used to configure the W32Time time service monitoring options. Windows registry can also be updated to enable enhanced W32Time service logging.

Commercial infrastructure monitoring products

Many products in this category offer broad monitoring capability over a wide-range network of operational aspects in large enterprise infrastructures. The coverage includes cloud, servers, networks, devices, applications, databases, and security operations. Some products require an NTP-specific module to be installed and configured for different levels of NTP server monitoring. Following is an example subset of infrastructure monitoring products applied in the context of NTP server monitoring.

The product Solarwinds is equipped with a collection of ipMonitor utilities to provide information about the operational status of infrastructure components (e.g., network devices, servers, applications). Solarwinds NTP monitoring product ipMonitor can be instantiated to verify NTP service availability.[4] It opens a live connection to the targeted NTP server and waits for the service to respond. If a valid UTC time is returned within the specified maximum test duration, then the test is deemed successful, and the monitor safely disconnects from the server. If the NTP server fails to respond or responds with an error code indicating that the service is not available, then the test is deemed a failure. NTP ipMonitor reports the status to the Network Operation Center (NOC) or specific dashboard(s). Solarwinds ipMonitor also stores monitoring statistics for recent activity and historical report analysis. The data can be displayed graphically or in tabular forms.

[3] https://en.wikipedia.org/wiki/Munin_(software)
[4] https://documentation.solarwinds.com/en/success_center/ipmonitor/content/ntp.htm

Similarly, Nagios[5] is an infrastructure event monitoring platform capable of event logging, network analysis (e.g., specific performance parameters), and visual presentation (e.g., infrastructure operation status). A limited-node open-source Nagios Core version is also available. Nagios can check NTP server parameters such as clock offset, ntpd daemon output, stratum level, synchronization status against a server, server availability, and connection validity.

Another enterprise infrastructure monitoring platform example is Opsview,[6] which has a dashboard interface, can be used for network scanning and discovery, service recovery, analytics logging, and network topology analysis. Performing NTP monitoring in Opsview requires the installation of the NTP monitoring product Opspack. An NTP Opspack is essentially a configuration structure consisting of host templates, which contain targeted NTP service checks, plugins, and specific attributes that are used to monitor the NTP service and/or the status of the NTP hosting devices.

The Paessler Router Traffic Grapher (PRTG) is another enterprise network monitoring tool. It automatically discovers devices and collect data with statistics on a variety of network performance parameters: bandwidth, traffic, devices, services, applications, cloud services, databases, virtual environments, availability, uptime, and ports/IP addresses. PRTG's NTP monitoring sensor[7] can be used to continuously monitor NTP server response and collects three statistics: NTP server response time, local system times and NTP server, and local versus server time difference. It provides a real-time view of the monitoring sensor's status, NTP historical data report, and NTP sensor's live data.

One example of special-purpose NTP monitoring is IBM Cloud Application Performance Management framework's NTP monitoring utility.[8] Its primary use is to address the IBM Database 2 (DB2) mirror deployment clock synchronization issues.[9] This utility is used to troubleshoot NTP clients, check status, start/stop NTP service, and control logging activity.

Different from others in this category, Meinberg's NTP Time Server Monitor is not part of an enterprise monitoring framework.[10] It works for NTP servers hosted on Windows NT/2000 and Windows Server 2003 and later. This monitoring software provides a GUI-based user interface, allowing for configuration and control of Windows-based NTP services. The current status of the local NTP service, as well as external NTP services, can be displayed. Some statistics of monitoring data can be graphically displayed by extracting data from NTP-relevant logfiles. Data extraction can be filtered and categorized as well.

NTP TRAFFIC MONITORING

[5] https://www.nagios.com/
[6] https://www.opsview.com/
[7] https://www.paessler.com/ntp_monitor
[8] https://www.ibm.com/docs/en/capm?topic=monitors-ntp-monitor
[9] https://www.ibm.com/docs/en/capm?topic=monitors-ntp-monitor
[10] https://www.meinbergglobal.com/

NTP network traffic monitoring helps to keep track of broader NTP operation issues beyond the NTP server functions. By analyzing NTP traffic flow, symptoms of attacks such as denial-of-service (DOS) can be discovered [3]. Furthermore, depending on the enterprise NTP service architecture (Figure 1), different aspects of NTP operations and health can be supervised.

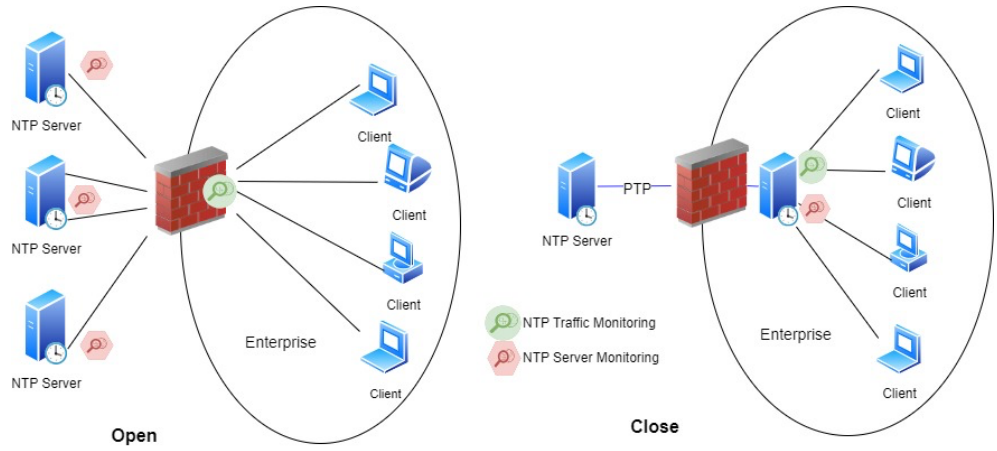

Figure 1. NTP server/traffic monitoring architectural consideration.

- **Open-loop NTP service enterprise architecture.** In an open-loop NTP service architecture, most (if not all) internal NTP clients acquire timing services and synchronize externally via public NTP servers (e.g., NIST, Google). All external subscriptions must go through enterprise firewall UDP port 123 access control rules. When scaled up, firewall hardware delay and access control software overhead create traffic congestion that could lead to time delays. Additionally, because NTP is a bidirectional protocol, all returning traffic must also go through the same privileged firewall UDP port 123, creating reverse traffic congestion with additional timing delays. As such, bidirectional traffic monitoring around enterprise firewall UDP port 123 can reveal probable cyberattacks (e.g., enterprise-level NTS DOS) as well as open-loop NTP subscriptions operational performance and deficiencies.
- **Closed-loop NTP service enterprise architecture.** Here, most, if not all, clients subscribe to timing services of NTP servers located within the enterprise. By not going through the firewall UDP port 123, the bidirectional congestion and timing delay are largely avoided. Traffic monitoring can thus be placed on internal NTP servers and clients. The server focus is similarly on UDP port 123 bidirectional traffic for operation status (e.g., activation status, server load) and security (e.g., DOS from internal rogue nodes) monitoring.

NTP traffic can be captured using a wide array of network monitoring tools such as wireshark or tcpdump when properly configured for NTP server IP and UDP port 123, or, for subscriber auditing, clients' IP addresses. Some commercial network monitoring tools also have traffic analysis plugins. However, more detailed NTP traffic analysis with GUI usually involves full-

featured data analytics tools such as RapidMiner or Splunk. The next section of this bulletin discusses high-level NTP data analytics approaches. Among the wide selection of network traffic collection and analysis tools applicable to NTP traffic monitoring, the following example subset illustrates their typical usage.

Sonicwall comes with a Packet Monitor that can be configured to filter out server IP, UDP port 123, and thus the bidirectional NTP traffic of internal NTP servers or the flows through the enterprise firewalls to external servers.[11] Traffic being monitored can be tabulated, but the tool provides no additional NTP-specific analytics capability.

Wireshark is a widely known free and open-source network protocol analyzer. It can be configured to capture NTP traffic by filtering the UDP port 123 flows. NTP traffic capture information is stored in network traffic packet capture files (pcap), which can then be further filtered, analyzed, and displayed for analysis. A large number of display filters are available for customization.[12]

Splunk is an example of general-purpose full-featured data analytics tools that can be used to analyze enterprise infrastructure operation, performance, and security management data. Such full-featured data analytics tools typically come with a large collection of graphical data presentation utilities for effective analysis. In the Splunk case, Splunk Stream utility, a data collection front end, can also be configured to capture the NTP network event data streams and directly feed into the proceeding filtering, indexing, and analysis processes. For server NTP traffic, the configuration is to capture NTP server's UDP port 123 traffic and extract metadata (e.g., statistics), which can then be aggregated to feed into the advanced signature matching or anomaly detection modules.[13]

Entuity is an enterprise network monitoring product. It is designed to monitor enterprise servers (with the hosted applications), map network topology, discover assets, manage network events, track assets, and analyze network traffic flows. It has a Network Flow Analyzer useful for NTP traffic analysis.[14] Entuity's Flow Dashboard is also useful for NTP monitoring.[15] It provides a summary of the flow information for the managed objects. The dashboard is dynamic and can be configured to monitor both closed-loop internal NTP server UDP port 123 traffic and open-loop firewall NTP-induced network congestion and delays. Its application dashboard can be configured to monitor internal NTP server performance, with latency thresholds for alerting.[16] Finally, the Event Management System can detect and provide notifications of predefined system changes in NTP service delivery,

[11] https://www.sonicwall.com/support/knowledge-base/how-to-capture-incoming-and-utgoing-ntp-traffic-using-the-packet-monitor/170505420483668/
[12] https://www.wireshark.org/docs/dfref/n/ntp.html
[13] https://docs.splunk.com/Documentation/StreamApp/8.1.3/DeployStreamApp/AboutSplunkStream
[14] https://www.parkplacetechnologies.com/entuity/network-flow-analyzer/
[15] https://support.entuity.com/hc/en-us/articles/360001702994-Flow-dashboard
[16] https://support.entuity.com/hc/en-us/articles/360001699793-Applications-dashboard

such as a “Port Utilization High” event for internal NTP server UDP port 123 or open-loop congestion (in either direction) for external NTP subscriptions.

ESET is primarily an enterprise security monitoring tool and service-offering product. It tracks a wide array of security threats across the enterprise network and analyzes/filters significant events to generate alerts. Especially for NTP monitoring, ESET’s inspect server rules can be customized to identify behaviors via received network events and metadata (e.g., statistics). Although ESET’s inspect server rules[17] are primarily for security detection purposes, they can be also adapted to monitor NTP server UDP port 123 traffic behavior and firewall NTP traffic anomalies via various severity degrees (colors) and severity scores.

DATA ANALYTICS FOR NTP MONITORING

In general, three major approaches are available for analyzing network traffic logs for intrusion and operation anomaly detection. The following examples illustrate how those methods can be applied to NTP traffic and log analysis.

Signature-based detection

This method analyzes NTP traffic/log and compares each packet/message against a database of predefined patterns, or signatures, of known attacks or faults. A match triggers an alert, indicating a potential intrusion or fault, and further investigation should be conducted. For example, NTP command “monlist” has a history of vulnerabilities, such as CVE-2013-5211, that allows for identification of DOS attacks [4,5]. If an NTP request with the “monlist” command is detected and the request generates a high number of responses, then the detection system should generate a warning and further analysis would be required. The drawback of this method is that it cannot detect never-before-seen intrusions or faults.

Anomaly-based detection

An anomaly-based method for NTP traffic targets deviations from typical NTP operation and communications. The approach typically starts by creating a baseline of normal NTP behaviors via statistical analysis of historical NTP traffic or server operation patterns. Many advanced statistical and AI/ML approaches exist to establish such normal profiles of NTP traffic and servers. Typical baselines could include features such as regular traffic volumes, request rate, payload size, event sequence, timing patterns, or protocol usage. By comparing real-time NTP traffic and server operation data against established baselines, NTP-relevant data (traffic and system log) can be monitored. Significant deviations from regular NTP patterns, such as unusual request rates, unexpected time offsets, or atypical response patterns, are flagged as potential security threats or anomalies. For example, NTP amplification is one form of distributed denial-of-service (DDoS) attack against NTP protocol. Attackers send many small NTP requests with a forged source IP address, causing the NTP server to respond with a high number of large-size packets to the victims, thereby overwhelming the network. With established normal profiles and thresholds of the normal

[17] https://help.eset.com/ei_rules/1.11/en-US/?rule_syntax.html

NTP server response size and frequency, such an NTP amplification would be a deviation from the normal operation profile, thus triggering an intrusion flag [6].

Although anomaly detection has the potential to identify zero-day attacks, it often suffers from a high false alarm rate.

Hybrid approach

A hybrid approach integrates both anomaly-based and signature-based detection methods to overcome the limitations of each approach. It uses signature-based detection to identify known threats and uses anomaly-based detection to uncover never-before-seen attacks, thereby improving the overall detection effectiveness. These two approaches can serve as cross references to increase the detection confidence level. For example, if the "monlist" attack is detected in the NTP traffic by the signature-based detector while over-threshold NTP server responses are being flagged by the anomaly-based detector, then an NTP amplification attack can be considered more pronounced.

Each detection approach—signature-based, anomaly-based, or hybrid—has its own strengths and weaknesses. The choice of the most suitable method depends on factors such as the specific security needs of the network, available resources, and the acceptable balance between false positives and false negatives. Beyond these three conventional approaches, more advanced AI and ML knowledge-based analysis approaches are available.

CONCLUSION

NTP is a critical protocol to implement timing synchronization over a distributed network consisting of devices with tight temporal correlation. NTP monitoring, on both servers and network traffic patterns, is crucial to maintain safe and secure operation of large and distributed infrastructures that depend on precise temporal synchronization. However, different NTP subscription architectures require different monitoring approaches. Closed-loop NTP architecture requires both internal server monitoring and server traffic monitoring, whereas open-loop architecture calls for traffic monitoring surrounding firewall network flows. Analogous to intrusion detection systems, NTP monitoring approaches could be signature matching, anomaly detection, or a combination of both. Intelligent monitoring approaches utilizing AI/ML are also possible.

Additionally, NTP server monitoring largely focuses on specific NTP protocol response behavior and timing parameter measurements. By contrast, NTP network traffic monitoring typically requires only common network monitoring tools, such as Wireshark, properly configured to filter UDP port 123 traffic. As such, many system owners likely already have in their possession the required NTP server and network traffic monitoring utilities/tools embedded within their existing systems. These tools would only need proper configuration.

REFERENCES

[1] Mike Hughes, “Beware the NTP ‘false-ticker’ – to do the time warp again…”, November 21, 2012, https://techblog.smashing.net/2012/11/21/beware-the-ntp-false-ticker-or-do-the-time-warp-again/
[2] Wikipedia, “Year 1900 problem”, https://en.wikipedia.org/wiki/Year_1900_problem
[3] Coudflare, “NTP amplification DDoS attack”, https://www.cloudflare.com/learning/ddos/ntp-amplification-ddos-attack/
[4] B. A. Sassani, C. Abarro, I. Pitton, C. Young and F. Mehdipour, "Analysis of NTP DRDoS attacks' performance effects and mitigation techniques," *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, Auckland, New Zealand, 2016, pp. 421–427, https://doi.org/10.1109/PST.2016.7906966.
[5] MITRE, National Vulnerability Database (NVD), “CVE-2013-5211 Details”, last modified November 1, 2023, https://nvd.nist.gov/vuln/detail/cve-2013-5211
[6] Oliver Ripka, “Analyzing NTP Traffic with Wireshark: A Practical Guide for Network Administrators”, PacketSafari Blog, January 9, 2022, https://www.packetsafari.com/blog/2022/01/10/analyzing-ntp-traffic-wireshark/

The Center for Alternative Synchronization and Timing (CAST) at Oak Ridge National Laboratory (ORNL) performs research, development, testing, evaluation, and technical assistance to enable resilient timing and synchronization for the power grid. Working closely with power utilities, timing hardware and software vendors, network operators, and federal stakeholders, CAST helps develop and validate alternative timing architectures to augment GPS time. CAST also translates and transfers ORNL’s research and development (R&D) advances in secure timing and grid communications to power sector applications, and engages across the broader timing community to develop best practices to ensure the resilience of US critical infrastructure. CAST is sponsored by DOE’s Office of Electricity. Visit https://cast.ornl.gov for more information.