sciendo International Conference KNOWLEDGE-BASED ORGANIZATION
Vol. XXIX No 3 2023

# STUDY OF METHODS AND TECHNIQUES FOR MANIPULATING THE TIME SYNCHRONIZATION COMPONENT OF NTP SERVERS IN COMPUTER NETWORKS

**Alexandru-Gabriel ROMANIUC*, Vlad-Cosmin VASILE**, Elena-Monica BORDA***

***Technical University of Cluj-Napoca, Romania**
****"Nicolae Bălcescu" Land Forces Academy, Sibiu, Romania**
**romaniuc_alexandrugabriel@yahoo.com, vladut.vasile@yahoo.com, monica.borda@com.utcluj.ro**

***Abstract:*** *In computer networks as well as in many other domains, time synchronization is carried out by some dedicated devices called time servers that uses network protocols, such as NTP (Network Time Protocol), PTP (Precision Time Protocol) or SNTP (Simple Network Time Protocol). Many domains and applications, such as, computer networks, security protocols, telecommunications, network services, and many others, depend on accurate time synchronization. According to the CVE (Common Vulnerabilities and Exposures) databases, between 2001-2022, more than 150 time synchronization protocols vulnerabilities have been reported. One of the most dangerous threats is the manipulation of the time synchronization information. Thus, the existence of such a vulnerability can generate serious security problems. Considering these aspects, in this article we aim to use some methods and techniques for the manipulation of the time component and use them to test some synchronization systems. This will require real-life scenarios of the use of time synchronization systems in which some methods and techniques are applied to exploit their vulnerabilities. Following these scenarios, we will be able to assess the risk to which these systems are exposed.*

**Keywords: NTP servers, time synchronization, network protocols, computer networks**

## 1. Introduction

Time synchronization is a very important process that requires highly accurate time reference signal sources. Globally, the main time reference sources are GNSS systems (Global Navigation Satellite Systems), atomic clocks, telecommunications systems or international standards. These sources represent the basic level of synchronization called Stratum 0 level. Time servers that synchronize with Stratum 0 level are called primary time servers or Stratum 1 level and are considered to be the most accurate [1]. In general, the higher the stratum level, the greater the distance from the base time source and the less accurate the timing can be. Time servers in a higher Stratum have many intermediate levels of servers to go through, which amplifies fluctuations and time delays. The hierarchy of these systems can be seen in Figure 1. In this article, we will use GNSS time source from Stratum 0 level and Stratum 1 time servers. Time sources of GNSS systems provide very accurate time information that can be used to synchronize network devices, computer applications, network protocols and more. Synchronization of the time component through GNSS systems is achieved by high-precision atomic clocks that can be found in the satellites of each constellation.

DOI: 10.2478/kbo-2023-0078

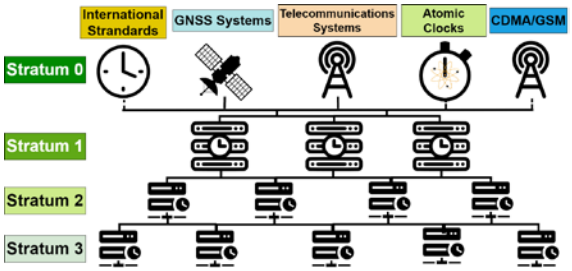

*Figure 1: Stratum hierarchy of time servers and sources*

Certain time servers have incorporated a GNSS sensor used to capture the radio signals transmitted by these systems. Thus, the sensor uses these signals to calculate its position and synchronize its clock with the satellite time. After the server synchronizes, it will be able to provide synchronization to all devices from the network.

Time synchronisation is achieved by using NTP, PTP or SNTP network protocols [1]. Some recent studies show that these protocols are vulnerable to amplification or spoofing attacks [2], [3], [4]. Thus, the time component can be manipulated through various methods, either by modifying the timing packets in the network or by compromising the time server. This further increases the risk of serious security problems.

Considering these things, we intend to analyze several scenarios in which some time synchronization systems will be targets of time manipulation attacks. In these scenarios, we will use a hardware time server solution and a software time server solution. Both systems are Stratum 1 level with synchronization from GNSS systems. The purpose of these scenarios is to assess the impact that these attacks generate and the security risks that the server's clients are exposed to.

## 2. Time Synchronization Protocols

The most widely used time synchronisation protocol is the NTP protocol. This protocol synchronises the network with a common time reference that is retrieved from sources such as atomic clocks or GNSS systems. The protocol is not secure or encrypted by default so data sent between the NTP server and clients can be intercepted and modified. Within a computer network, this protocol uses network packets to transmit timing data to all the equipments and computer systems that need this information. There are 3 types of packets carried between NTP servers and clients: NTP Request Packets: are sent by the client to the NTP server to request synchronisation information; NTP Response Packets: are sent by the NTP server after receiving NTP query packets from an NTP client system; NTP Control Packets: are used to send commands and receive reports between time servers [5].

Because the information in these packets is not encrypted or secure, it can be captured and modified. In our scenarios, the systems under-test will be configured to use this protocol.

A more complex, secure and accurate alternative is the PTP time synchronization protocol. It was developed to replace the NTP protocol and it offers greater accuracy. This protocol is harder to implement but offers increased security.

## 3. Time Servers under Test

The time servers that will be subjects in our scenarios are two systems. The first one is a system from the manufacturer Microsemi model SyncServer S600. The second one is a software time server solution installed on the Raspberry Pi 4 system on which a GNSS module from the manufacturer U-blox model EVK-F9P has been installed and configured. All these solutions are Stratum 1 level and the main source of synchronisation is from GNSS systems.

### 3.1. Microsemi SyncServer S600

Syncserver S600 is a hardware solution manufactured by Microsemi company and is characterized by very high performance and reliability for time synchronization services in high-performance computer networks. For network synchronization, this server uses a system based on NTP and

PTP protocols, which provides a timing accuracy of up to 30 nanoseconds. In terms of main constellations supported by the server, it is compatible with multiple constellations such as Galileo, GLONASS, GPS, QZSS and Beidou. Figure 2 shows the server connected to a LAN (Local Area Network) used in our scenarios.

*Figure 2: Syncserver S600 (up), Cisco Switch (middle) and Cisco Router (down)*

### 3.2. Raspberry Pi 4 time server with EVK-F9P GNSS module

The Raspberry Pi 4 is a small but relatively powerful processing unit capable of running demanding processes in terms of resources and architecture. Thanks to its connectivity, the Raspberry Pi system can be configured to provide network services such as: NTP server, DNS server, VPN server and more. Turning the Raspberry Pi into an accurate time server can be done by installing and configuring a network service called ntp daemon. This service can be configured to obtain time synchronization from other time servers or from a GNSS sensor installed on the Raspberry Pi. Thus, a GNSS module called EVK-F9P will be installed and configured on the Raspberry Pi which will turn the system into a Stratum 1 level time server. Figure 3 shows the Raspberry Pi time server connected with the EVK-F9P.

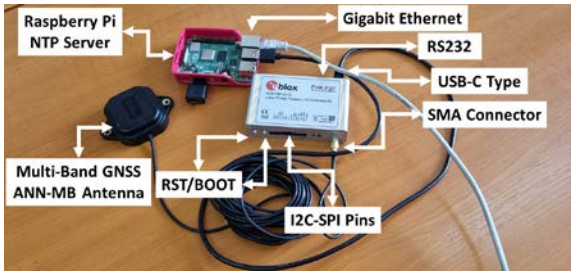


*Figure 3: Raspberry Pi with EVK-F9P module*

The EVK-F9P evaluation module is a High Precision GNSS device that provides multi-band and multi-constellation positioning and timing. The module is compatible with the main global navigation systems, namely GPS, Galileo, GLONASS, Beidou and QZSS. The module is also compatible with the following GNSS bands: B1I, B2I, E1B/C, E5b, L1C/A, L2OF.

Installing the EVK-F9P module on the Raspberry Pi can be done as shown in the following steps:

- the gps daemon tools will be installed to help interpret the data collected by the EVK-F9P module;

```
$ sudo apt-get install gpsd gpsd-clients python3-gps
```

- next, we will edit the gpsd daemon configuration file with the following commands: START_DAEMON="true", DEVICES="/dev/ttyACM0"(ttyACM0 is the interface assigned to the EVK-F9P module), GPSD_OPTIONS="-n".

```
$ sudo nano /etc/default/gpsd
```

- next, we will check if the gpsd service is running;

```
$ sudo systemctl status gpsd.service
```

- also, to verify that the gpsd daemon is receiving data from the EVK-F9P module, we can run the utilities using the commands: gpsmon -n, cgps -s, xgps;

Next, we will install the ntp server and we will configure it so that the synchronization data is obtained from the gpsd daemon which receives data from the EVK-F9P:

- the ntp server will be installed;

```
$ sudo apt-get install ntp ntpdate
```

- next, we will edit the configuration file of the ntp service so that it receives data from the gpsd daemon;

```
$ sudo nano /etc/ntp.conf
```

```
server 127.127.28.0 minpoll 4 maxpoll 4
fudge 127.127.28.0 time1 0.0 refid gnss
```

- ntp service will be started;

```
$ sudo systemctl start ntp.service
```

- next, we will check if the ntp service is running;

```
$ sudo systemctl status ntp.service
```

- this could be done also by using the commands: sudo watch –n 1 ntpq –p, and sudo ntpshmmon;

## 4. Time Manipulation Methods and Techniques

The scenarios we are going to study will use two methods implemented into a Python application. The first method will use the arpspoof library to impersonate the time servers so that synchronization traffic from them is redirected to the NTP Spoofer station. The second method involves the use of dsniff, scapy and netfilterqueue libraries which will modify the data from the synchronization packets coming from the time servers. The application interface is shown in Figure 4 and will be installed on the NTP Spoofer station.

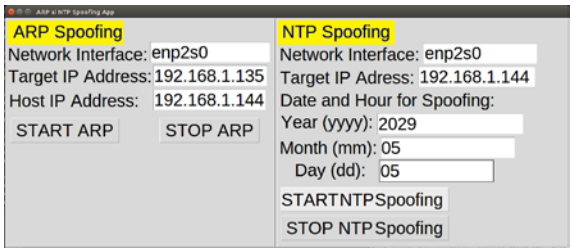


*Figure 4: Application - ARP and NTP spoofing*

The application uses the vulnerability of the Address Resolution Protocol (ARP) to modify the ARP table of the victims, a process called ARP poisoning [6]. In this way, the server's synchronization traffic is redirected and modified by the spoofing application.

## 5. Testing Scenarios and Results

### 5.1. Initial architecture configuration

The devices used in the scenarios are located in a laboratory as shown in Figure 5 and the server antennas will be located outside, on tripods. The time manipulation scenarios assume the use of a LAN architecture shown in Figure 6. Within the topology we have a user zone in which we set up three workstations with Windows 10, Windows 11 and Ubuntu 20.04 operating systems. Each of them is configured to get synchronization from SyncServer S600 and from Raspberry Pi time server. Also, the IP addresses of each device are noted in Figure 6. Within the network, we have the NTP-Spoofer station which was unauthorized infiltrated into the network. Its role is to impersonate the time servers and modify the time packets sent by them.

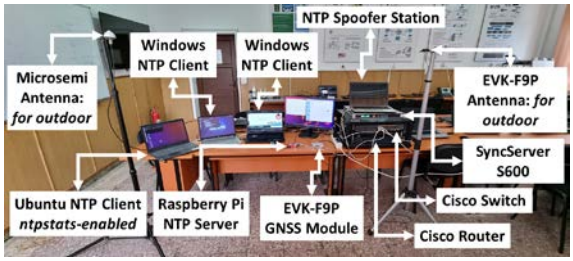


*Figure 5: Network topology for NTP packet spoofing: SyncServer S600 and Raspberry Pi in the laboratory*

In order to capture the results of the NTP manipulation traffic, ntpstats logging has been enabled on the Ubuntu station. The ntpstats logging is a process that helps monitor time servers and collect important parameters.

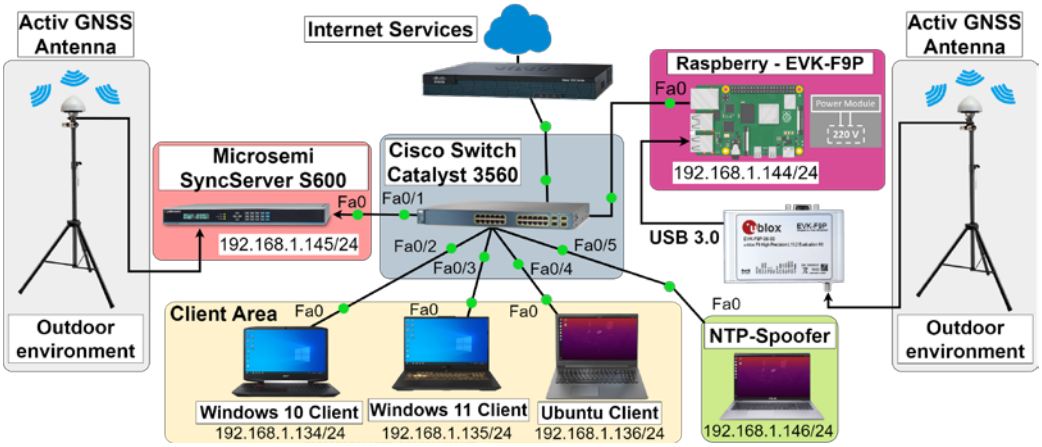


*Figure 6: Network topology for NTP packet spoofing: SyncServer S600 and Raspberry Pi*

Through NTP stats records we can collect data about offset, jitter, dispersion, delay and MJD (Modified Jilian Data). These parameters help to evaluate the time server performance.

**5.2. NTP Time Manipulation Results**

Based on the created topology, each client synchronizes with the time servers. When the NTP Spoofer application intercepts synchronization packets from the servers, it modifies them with the date set in the application, and sends them to the client that requested the synchronization.

The scenarios were made on 05-05-2023 and will last 24 hours from 00:00 to 24:00 UTC (Coordinated Universal Time). The time data manipulation will be split into two parts between 11:00-13:00. The first part will last one hour and involves testing without intervention on the ntp daemon monitoring service; and the second part will last one hour as well and involves the restart of the ntp daemon monitoring service during the attack. The restart in the second part of the attack will help the service to recalculate its parameters. The first parameter assessed is MJD Date. From Figure 7 it can be seen that for the second part of the test, the value was changed from 60069 (05-05-2023) to 62261 (05-05-2029) according to the data within the application.

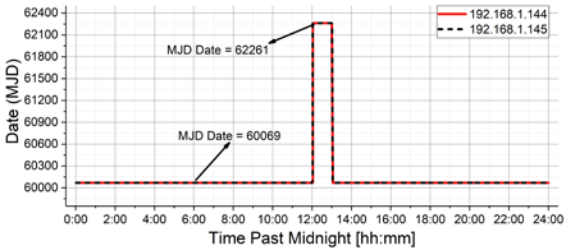


*Figure 7: MJD Date modified in second part*

The second parameter evaluated is the clock offset shown in Figure 8 for 24 hours.

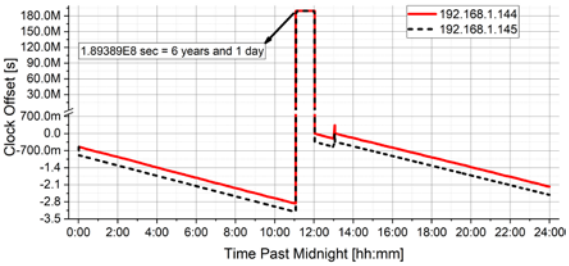


*Figure 8: Clock Offset modified in first part*

Clock offset is the time difference between the local clock and the time server clock [7]. From figure 8 it can be seen that for the first part, the value has been changed to 1.89389E8 seconds (6 years), for the entire hour of the first part.

The third parameter evaluated in Figure 9 is the roundtrip delay which is the time it takes for NTP packets to travel the distance between the server and the system [7]. For both parts this has been modified.

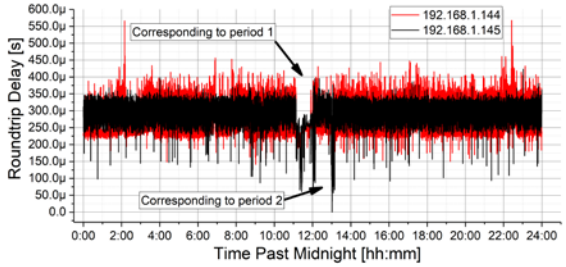


*Figure 9: Roundtrip delay changed: both parts*

The fourth parameter evaluated in Figure 10 is the dispersion, which indicates the possible variation of the time provided by the NTP server. For the second part, it has been modified due to the modification of the MJD parameter.

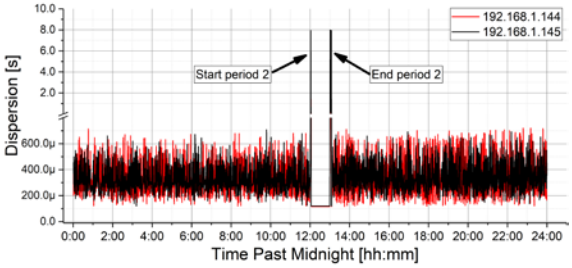


*Figure 10: Dispersion changed: second part*

The fifth parameter evaluated in Figure 11 is RMS Jitter, which indicates the variation or fluctuation in the arrival time of NTP packets from the server [7]. It can be seen that at 11:00 when the offset parameter was changed, the time difference between the NTP packet recorded at this point in time compared to the previous one was very big.

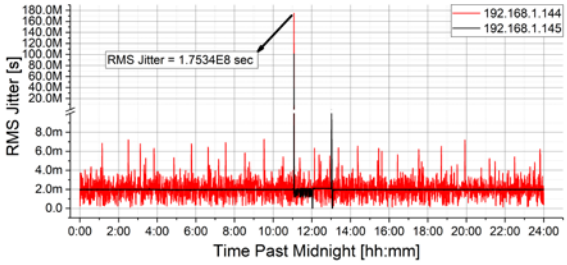


*Figure 11: RMS (Root Mean Square) Jitter*

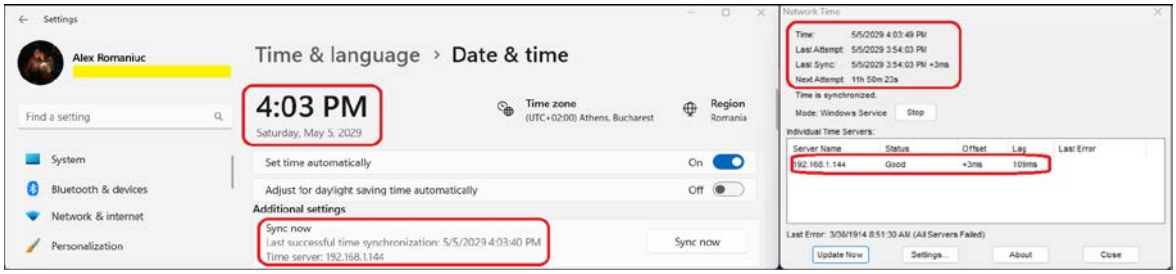


*Figure 12: Consequences of time manipulation for the Windows 11 client – Date and time.*

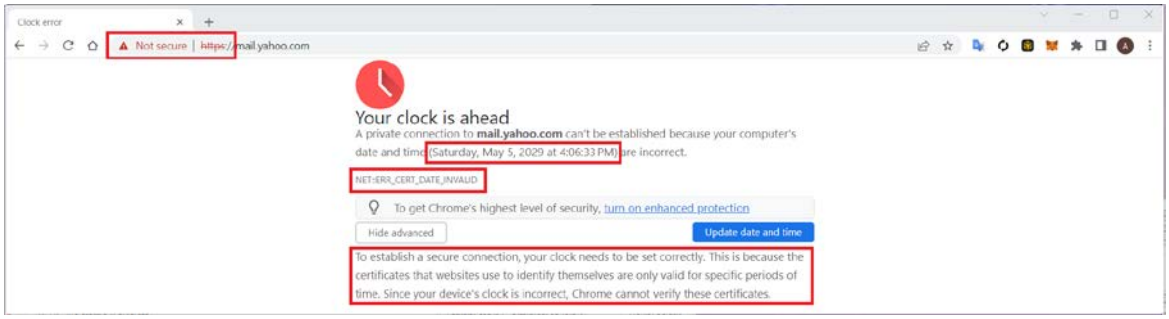


*Figure 13: Consequences of time manipulation for the Windows 11 client – Internet navigation.*

Some consequences of time manipulation were observed on the NTP client systems. For Windows 11 client, these are shown in Figure 12 and 13. The reference date and time of the system has been changed.

Also, the Internet connection is not possible because the browser thinks that the digital certificates are expired. This will not allow a secure connection to the web server.

## 6. Conclusions and Future Work

Through this paper we have shown that timing data can be manipulated with ease and network systems and equipment can be affected. The scenarios presented showed the effects generated by the vulnerabilities of the synchronization protocols. Moreover, such attacks could cause serious security problems. Because many network protocols and systems use this data, it is important to try to prevent those kinds of attacks. As a future work, we aim to test and develop security mechanisms against such attacks.

## References List


[1] Novick A. & Lombardi M. A Comparison of NTP Servers Connected to the Same Reference Clock and the Same Network. Proceedings of the ION 2017 Precise Time and Time Interval (PTTI) Meeting. 2017. 264-270. 10.33012/2017.14977.

[2] Alghamdi W. Precision time protocol attack strategies and their resistance to existing security extensions. 2021. Cybersecurity. 4. 10.1186/s42400-021-00080-y.

[3] Rangaswamy S. & Murthy S. An Overview of Network Time Protocol. 2021. High Technology Letters. 27. 10.37896/HTL27.6/3704.

[4] Widagdo G. Real-Time Early Detection NTP Amplification Attack. 2019. ACMIT Proceedings. 3. 76-84. 10.33555/acmit.v3i1.29.

[5] Ameri A. & Johnson D. Covert Channel over Network Time Protocol. 2017. 62-65. 10.1145/3058060.3058082.

[6] Nam S.Y., Jurayev S., Kim SS. et al. Mitigating ARP poisoning-based man-in-the-middle attacks in wired or wireless LAN. J Wireless Com Network 2012. 2012. 89. https://doi.org/10.1186/1687-1499-2012-89.

[7] Mills D.L. Computer Network Time Synchronization: The Network Time Protocol. Taylor & Francis. 2010. pp. 12–. ISBN 978-0-8493-5805-0.