Combatting internet time shifters

Michael Schapira
The Hebrew University of Jerusalem

Combatting internet time shifters

The *Network Time Protocol (NTP)* synchronises computer systems across the internet (typically within a few milliseconds of Coordinated Universal Time) and is ubiquitously deployed. Many internet applications, including TLS certificates, internet naming/addressing (via DNS and DNSSEC), internet routing security mechanisms (namely, RPKI), Kerberos, financial services, and many others, crucially rely on NTP for both correctness and safety.

How does NTP work?

NTP is based on a client-server architecture.

NTP clients

As illustrated in Figure 1(a), an NTP *client* periodically queries a set of *time servers*. The client exchanges messages with these time servers to learn the current clock readings at the time servers and to estimate the network delay with respect to each time server (Figure 1(b)). Based on the estimated delays and reported clock readings, the client computes the 'offset' for each time server, i.e. the estimated difference in time between the client's local clock and the time server's local clock. To update its local time, the client feeds the resulting offsets into an algorithm that discards outliers and computes, from the 'surviving' offsets, a new time to update the local clock to.

NTP servers and the NTP server pool

The NTP Pool Project centralises access to a pool of thousands of volunteered NTP servers across different countries and organisational domains. The NTP server pool assigns time servers to NTP clients based on client geolocation and balances load across its servers. By default, the NTP Pool Project is used by most, if not all, open-source OS (operating system) distributions—including all major Linux distributions—router vendors, home automation systems, security cameras, household appliances, and more.

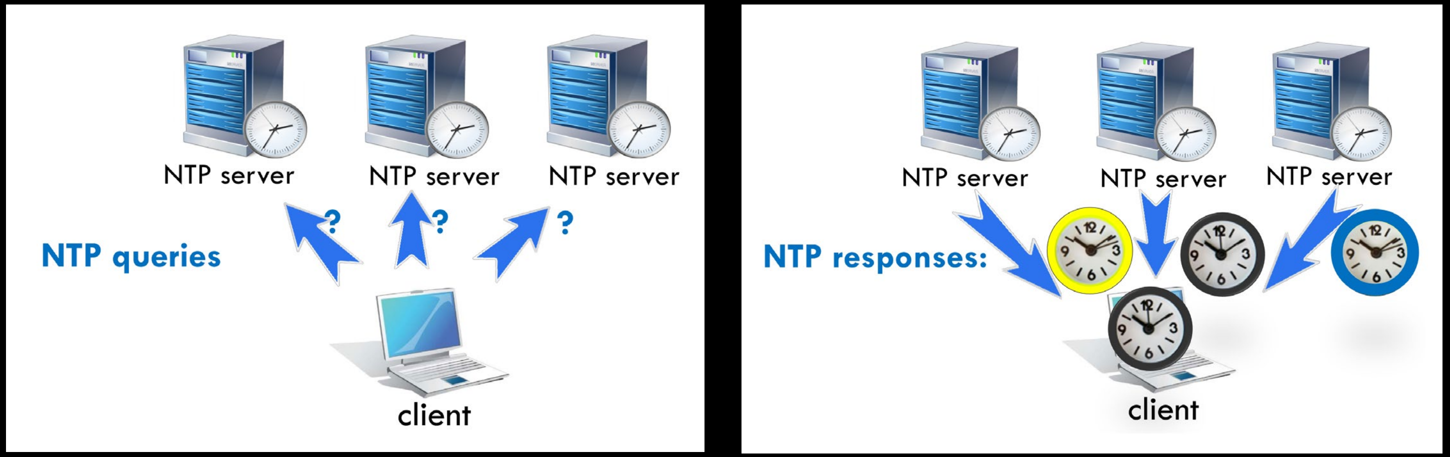

Figure 1a and 1b: An NTP client (a) exchanges messages with time servers and then (b) receives current clock readings at the time servers and estimates the network delay for each time server.

Adobe Stock © jakkaje8082

NTP is vulnerable to time-shifting attacks

Similarly to other internet infrastructure components (for example, TCP/IP, BGP, DNS), NTP was designed without security in mind. NTP's design thus reflects the need to achieve correctness in the presence of inaccurate clocks ('falsetickers'), assumed to be fairly rare, as opposed to designated attacks by powerful and strategic adversaries. Consequently, NTP is alarmingly vulnerable to attacks.

We consider *time-shifting attacks*, in which an attacker shifts the local time at the NTP client forward/backwards. Recall that the local time at a client is determined based on the clock readings received from the time servers the client interacts with and the delay with respect to these time servers, as estimated by the client. By reporting false clock readings at time servers or affecting the experienced delay between the client and the time servers, an attacker can induce wrong decisions at the NTP client. In particular, if the attacker has sufficient presence in the set of time servers with which an NTP client communicates, it can stealthily shift time at the client by repeatedly pushing the client further away from the actual time when queried by the client. Such attacks can be launched, for example, by a man-in-the-middle (MitM) attacker capable of intercepting and tampering with NTP messages between the client and (a significant subset of the) time servers, or by an attacker in direct control of (a subset of) the NTP servers themselves. See Figure 2.

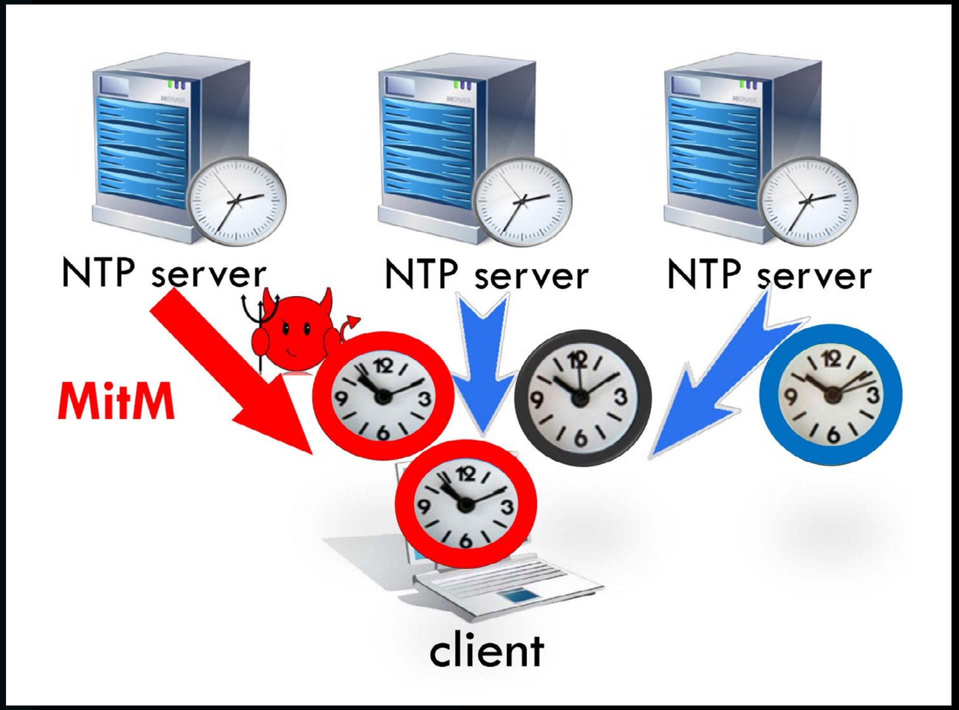

Figure 2: An illustration of a man-in-the-middle (MitM) attack.

Recently introduced patches to NTP's implementation eliminate/mitigate some *off-path* attacks and implementation flaws. Yet, MitM attackers—let alone attackers in direct control of time servers—are often deemed too strong to protect against. The cure to some of NTP's ailments may lie in encrypting NTP traffic between clients and servers. However, even ubiquitous encryption and authentication is insufficient for fully protecting NTP time synchronisation from a MitM attacker capable merely of delaying and replaying packets and, of course, from malicious time servers.

> "MitM attackers—let alone attackers in direct control of time servers—are often deemed too strong to protect against."

Just how vulnerable is NTP to malicious timeservers?

Our focus is on investigating and addressing NTP's vulnerability to strategic attacks by malicious time servers. The natural starting point for this investigation is the NTP Pool Project. As evidenced by the many millions of systems that rely on the NTP server pool for time synchronisation, the NTP Pool Project successfully facilitates accurate time synchronisation at scale. However, as our results demonstrate, the pool's mechanisms for assigning time servers to NTP clients are vulnerable to hazardous attacks. We considered two attack strategies: (1) compromising existing NTP servers and (2) injecting new time servers into the NTP server pool, and show that both are alarmingly effective.

An attacker in control over fairly few existing NTP servers in a country/continent can impact time at NTP clients across the entire country/continent. An NTP client that uses the NTP server pool is periodically assigned time servers to sync to by the pool. We showed (Perry *et al.*, 2021) that even an attacker in control of fairly few servers in the pool can inflict significant harm. The root cause for this vulnerability is that the pool's mechanism for assigning time servers to clients is oblivious to inter-server dependencies. Local time at most NTP servers is derived from interaction with *other* time servers. This implies that an attacker in control of a few NTP servers with great influence over other time servers can potentially influence time at a client *indirectly* by manipulating time at other time servers (with whom the client communicates). We showed, through extensive empirical analyses, how this can be leveraged by an attacker for shifting time at country/state-scale, or even continent-scale, adversely impacting the performance or security of various applications. In particular, an attacker in control of merely 10s of time servers in Europe or the US (out of thousands of time servers in Europe and many hundreds in the US) can shift time forward/backwards by hours at many clients across the entire continent/country, impacting various applications of interest. We observed, however, that more effective and also simpler to launch attacks are feasible by injecting new time servers into the NTP server pool.

Influencing time computation at clients via the injection of new time servers into the NTP server pool is effective and simple. Entering a new time server into the NTP server pool is remarkably easy. Moreover, she/he entering the new time server, and operating that time server, is trusted to provide truthful information about the time server. We showed how, through the proper configuration of parameters, an attacker that enters a time server into the pool can increase the number of clients its time server is assigned to by the pool by three orders of magnitude (compared to the default time server configuration). As our empirical analyses established (Perry *et al.*, 2021), this translates to hundreds of thousands of clients per hour trying to sync with that time server. We showed how this state of affairs can be leveraged by an attacker for launching large-scale *opportunistic* attacks (as with taking over existing NTP servers) and for launching strategic and stealthy attacks that target specific NTP clients. In addition, this attack strategy is not limited in terms of the extent to which time can be shifted and can be employed to shift time at clients by days, weeks, months, and beyond, impacting a much broader range of internet services.

Why is NTP so insecure?

There are two crucial aspects of NTP that make it particularly susceptible to time-shifting attacks.

1. **The mechanisms used by the NTP server pool to determine which time servers a particular NTP client should sync with.** As discussed previously, the NTP pool's mechanisms for determining which time servers in the pool to assign to a client ignore potential dependencies between the assigned time servers, which enables an attacker in direct control of one time server assigned to the client to influence the local times at other time servers assigned to the client. In addition, the ease at which an attacker can inject disproportionately influential time servers into the pool enables an attacker to influence time at many NTP clients that query the pool for time servers to sync with.
2. **The client-side scheme for inferring the current time from time server-provided responses is highly susceptible to manipulation.** As previously explained, NTP was not designed with strategic attackers in mind, but to provide accurate time in the presence of time servers with false times, assumed to be fairly rare and non-strategic. Consequently, the algorithm executed at the NTP client to detect which time servers (from those assigned to the NTP client by the server pool) provide false time reports, and to infer the correct time from the information reported by the remaining time servers, is highly vulnerable to powerful, strategic attackers.

Putting an end to internet time shifters

We presented (Deutsch *et al.*, 2018; Perry et al., 2021) a holistic approach for securing NTP from time-shifting attacks, which comprises two elements:

1. **Ananke: a root-of-trust for internet time synchronisation.** Ananke is a set of 100s of NTP servers that are carefully chosen, and periodically audited, to eliminate the possibility of inter-server dependencies and to significantly raise the bar for an attacker whose purpose is to inject malicious time servers into Ananke. (See: Perry *et al.*, 2021.)
2. **Chronos: a secure NTP client.** A Chronos NTP client periodically queries small subsets (e.g. of size 10–15) of NTP servers from a large set of time servers, consisting of 100s of time servers, to solicit timing information, and then applies a theory-informed algorithm to remove outliers and average over the remaining responses. We proved (Deutsch *et al.*, 2018) that, so long as the attacker cannot influence the local times at a large fraction of the time servers in a large time server set, this crowdsourcing scheme guarantees that the client's internal clock remains close (time-wise) to the universal time (UTC) and that the clocks of any two Chronos-clients remain close to each other. For instance, to shift time at a Chronos client by over 100ms from the Coordinated Universal Time, even a powerful attacker requires over 20 years of effort in expectation.

"We present a holistic approach for securing NTP from time-shifting attacks."

Adobe Stock © Oleg

Putting Ananke and Chronos together

Our objective is to enhance NTP's security against malicious time servers while not adversely impacting its time accuracy and precision, nor the distribution of load across time servers. Our solution is simple: each NTP client should simultaneously run *two* parallel synchronisation processes. The first synchronisation process is precisely that used by today's NTP clients to sync with pool-assigned servers in their region. This 'primary' synchronisation process is used, by default, to determine the client's local time. The second synchronisation process is run in 'watchdog mode'; the client applies Chronos' provably secure time-inference scheme to the servers in Ananke. So long as the watchdog's (Chronos') time does not deviate from the primary time by 'too much', the primary synchronisation process continues to update the local time. If, however, the results of these two time calculations are too far apart, which is indicative of an attack, the watchdog takes over and updates the local time (until such a time when the two computed times are close again).

We are currently promoting the standardisation of Chronos and Ananke at the Internet Engineering Task Force (IETF). Chronos has recently been awarded the IETF/IRTF Applied Networking Research Prize, which is "awarded for recent results in applied networking research that are relevant for transitioning into shipping Internet products and related standardisation efforts."

References

Deutsch, O., Schiff, N.R., Dolev, D. and Schapira, M. (2018) 'Preventing (Network) Time Travel with Chronos', *The Network and Distributed System Security (NDSS) Symposium 2018.* San Diego, 18–21 February. Available at: https://www.ndss-symposium.org/wp-content/uploads/2018/02/ndss2018_02A-2_Deutsch_paper.pdf

Perry, Y., Schiff, N.R. and Schapira, M. (2021) 'A Devil of a Time: How Vulnerable is NTP to Malicious Timeservers?', *The Network and Distributed System Security (NDSS) Symposium 2021.* Online, 21–25 February. Available at: https://www.ndss-symposium.org/wp-content/uploads/ndss2021_1A-2_24302_paper.pdf

PROJECT SUMMARY

Arguably, the internet's biggest security hole is the Border Gateway Protocol (BGP), which establishes routes between the organisational networks that make up the internet (e.g. Google, Facebook, Bank of England, Deutsche Telekom, AT&T). The insecurity of the internet's routing system is constantly exploited to steal, monitor, and tamper with data traffic. Yet, despite many years of Herculean efforts, internet routing security remains a distant dream.

The goal of the SIREN project is to propose and investigate novel paradigms for closing this security hole.

PROJECT LEAD

Michael Schapira is a professor of computer science and the co-leader of the Fraunhofer Cybersecurity Center at the Hebrew University. His research focuses on (Inter) network architectures and protocols. Prior to joining Hebrew U, he was a researcher at Google NYC, UC Berkeley, Yale University, and Princeton University. He is a recipient of numerous awards, including faculty research awards from Microsoft Research and Google, several Applied Networking Research Prizes from the IETF/IRTF Applied Networking Research Prizes, and an ERC Starting Grant.

CONTACT DETAILS

Michael Schapira
The Hebrew University of Jerusalem

☎ +972 2 5494570
✉ schapiram@cs.huji.ac.il
www.cs.huji.ac.il/~schapiram

FUNDING

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No.678921.