Survivable, Real Time Network Services

Defense Advanced Research Projects Agency
Contract F30602-98-1-0225, DARPA Order G409/J175

Quarterly Progress Report
1 January 2001 - 31 March 2001

David L. Mills
Electrical Engineering Department
University of Delaware

1. Introduction

This report covers the work done in support of the DARPA Information Technology Office program in computer networking. Contributors to this effort include Prof. David L. Mills and graduate Tamal Basu. Graduate student Qiong Li has completed his dissertation and been granted the PhD degree. The project continues previous research in network time synchronization technology jointly funded by DARPA and US Navy. The technology makes use of the Network Time Protocol (NTP), widely used in the Internet, together with engineered modifications designed to improve accuracy in high speed networks. Specific applications benefiting from this research include multicast topologies, multimedia, real-time conferencing, cryptographic systems, and management of distributed, real-time systems.

This quarterly report is submitted in traditional report form on paper. As the transition to web-based information dissemination of research results continues, almost all status information and progress reporting is now on the web, either on pages belonging to the principal investigator or to his students. Accordingly, this and future progress reports will contain primarily schedule and milestone data; current status and research results are reported on web pages at www.eecis.udel.edu/~mills in the form of papers, technical reports and specific briefings.

2. Autonomous Authentication

The current Autokey implementation uses the rsaref2.0 software distribution formerly distributed by RSA Laboratories, Inc. This distribution was intended for evaluation only and not for support of a generic protocol. However, code with equivalent functionality has been available from other sources both in the US and other countries. It would be most convenient to include the rsaref2.0 software in the NTP software distribution, avoiding the hassle to find an appropriate alternative elsewhere. A request to do this was forwarded to RSA Laboratories and is in process by their legal department. Nothing has been heard for the last two months. Accordingly, this plan is probably dead.

Alternately, and probably the best solution, is to adapt the cryptographic routine interface to use the Open SSL tools, which are widely available. We have obtained a copy of the public source code and begun a project to adapt the NTP cryptographic algorithm to use it.

The current status of the Autokey protocol and implementation is available via the web at www.eecis.udel.edu/~mills/autokey.htm.

3. Autonomous Configuration

Work continues on the development and test of the Manycast autonomous configuration scheme. In this scheme a multicast client sends an ordinary request packet to a multicast group address and potential multicast servers in TTL range have the opportunity to respond with an ordinary unicast message. Each received message causes the client to mobilize an ephemeral association, which then proceeds as if the server were originally explicitly configured.

An important feature of the Autokey protocol is that automatic refreshment of the cryptographic basis values is an intrinsic function of the protocol. In particular, the key list is refreshed about once per hour and the agreement values about once per day. Unfortunately, while the key list refreshment is generally transparent to clients, the agreement refreshment is not. In the current design a agreement refreshment causes all cryptographic associations to be broken and subsequently reformed. One of the reasons why this is done is to protect against an intruder seeking to deny service by breaking an association and falsely making the client believe the agreement values had changed and then attempt to rebuild the association.

We are exploring possible protocol changes that would ameliorate the client protocol dance while still maintaining sufficient vigilance against such a hacker attack.

The current status of the Manycast mode and implementation is available via the web at www.eecis.udel.edu/~mills/autocfg.htm.

4. The Kiss-of-Death Packet

The load on our most visible primary (stratum 1) time server rackety.udel.edu continues to escalate. This machine sits lonely in the equipment rack and generally requires no special care. Recently, there have been indications the machine was nearing an overload state. There were occasional time offset spikes visible from other machines and the jitter reported on the radio clocks gradually became significantly worse. Inspection of the usage counters showed an alarming increase in the user population double what it was a year ago. Since the rules of engagement posted in the list of public time servers at www.ntp.org require advance notice of use and no requests have been received over the year, the natural conclusion is that folks are not respecting those conditions.

An inspection of the monitoring data suggested that two-thirds of the received packets were for historic NTP versions, not only NTPv3 but even earlier versions, including NTPv1 which was discontinued almost a decade ago. In fact, the code servicing NTPv1 packets was itself incorrect for some NTPv1 modes. Accordingly, all support for NTPv1 has been removed from the current distribution. And, in order to throttle the load on rackety.udel.edu, the access control function has been set to disregard all NTP versions except NTPv4. This was not a popular move, but disgruntled users were told we run two other public primary servers, both TrueTime NTS-100/200 devices.

The experience suggested the need for some kind of active response to unwanted traffic. Previously, unwanted traffic filtered out by the access control function was simply discarded. In order to further discourage such traffic, the server code was modified to optionally return a special “kiss-of-death” packet marked in such a way that the NTP client association will demobilize the association and send an appropriate message to the system log. While this behavior will happen

for NTP server code dating from the modification, previous NTP server code will simply disregard the kiss-of-death packet.

5. Infrastructure

There have been a continuous stream of contributions to the NTP software distribution from the developer's group. These include refinements to the clock filter and clock discipline algorithm, new reference clock drivers and The latest version of NTP Version 4 has been deployed in the DCNet research network, EECIS campus network and CAIRN research network.

A security problem with NTP was reported to the CERT in late March. While it was easily fixed, the issue caused a great deal of misconception and rumor mongering in various user communities. The problem was a hole in the message processing code in the ntpq utility program which invited a determined hacker to insert bogus code beyond the end of the otherwise valid message. If the message is stored in a temporary buffer on the stack and the stack grows downward, the bogus code could overwrite the return address and other temporaries on the stack and potentially gain control of the processor. Some users were petrified with fear that their herds of NTP servers were being attacked, resulting in massive break-ins and uniterruptable power source shutdowns.

The defect was quickly fixed and the distribution uploaded to the NTP web site www.ntp.org. An advisory was sent to the CERT and redistributed to the newsgroups and interest lists. Apparently, most folks don't get their security information from these sources, since irate messages were arriving at a rate over 50 messages per day to this investigator's mailbox a week after the advisory. However, there was one important issue that was often overlooked in the wash of messages across the net. There are a number of users, including large corporations, that continue to use NTPv3, which is no longer maintained by the NTP developer's group. The ntpq program itself was written by Dennis Fergusson early in the last decade and essentially unchanged since then. Thus, the defect has been in both NTPv3 since then and in NTPv4 since its departure from the NTPv3 code base. While the patch supplied to the CERT works in either version, a patched NTPv3 version is not available from the NTP web site.

A technical vulnerability analysis was supplied to the CERT shortly after the incident. The original report stipulated the possibility that a hacker could gain root privilege through the defect and provided a test program which allegedly did just that, presumably by causing the processor to execute code on the stack. However, the buffer was in fact in file scope and not on the stack. While there remains a dim possibility that the global variables allocated just after the message buffer could be compromised, the likelihood that this could lead to a virus infection is very small. In particular, a design which might result in something worse than a coredump would have to be specific to each architecture, operating system, compiler and library. In addition, it was found the test program report of alleged success was unreliable at best, since it reported success even when the NTP daemon was not running. While a determined attempt to reproduce the incident in every architecture, operating system and version available to this investigator resulted in no adverse effects whatsoever, other testers did report a core dump in some configurations.

The lessons learned from the incident were not surprising, even though answering mail and testing configurations consumed almost every waking minute for over two weeks. One problem was that the volunteer distribution maintainer was out of town during and shortly after the incident, so the patch and updated distribution were not properly numbered. In addition, the patched distribu-

tion had a number of minor bugs that were later fixed in the repository at the NTP web site. As of this writing the repository version has not been officially uploaded to the NTP web site.

In other news, we have rescued a number of orphaned computer systems originally headed for the scrap heap. These hulks have been instantiated with current and historic operating system versions to be used in NTP validation tests. The complete suite of architectures and operating systems includes three Alpha systems running Tru64 (OSF/1 derivative), an Alpha system running Debian Linux, Several UltraSPARC systems running Solaris 2.8, two SPARC IPC systems running SunOS 4.1.3 and a third running Solaris 2.7, three Pentium systems running FreeBSD 4.1, one HP 9000 system running HP-UX and one Digital RISC system running Ultrix 4.1.Volunteer Harlan Stenn has organized the autoconfigure and build scripts which tests the several combinations of compilation options.

6. Future Plans

Our plans for the next quarter include further work on the NTP book mentioned in the last report. Mr. Basu is to continue work on the simulation project preparing for his graduation in late May.

We plan to continue developing the cryptographic algorithm interface to use the Open SSL tools widely available on the Internet. This will both avoid the distribution problems with the existing rsaref2.0 software as well as provide support for additional algorithms, including SHA and others. We also plan to develop support for certificate retrieval and transport.

We plan to continue work on the Autoconfigure scheme, in particular to develop means to avoid the agreement refreshment problem mentioned earlier. Other improvements expected include a rewrite of the configuration code and name resolution code.

7. Publications

All publications, including journal articles, symposium papers, technical reports and memoranda are now on the web at www.eecis.udel.edu/~mills. Links to the several publication lists are available on that page, as well as links to all project descriptions, status reports and briefings. All publications are available in PostScript and PDF formats. Briefings are available in HTML, PostScript, PDF and PowerPoint. The project descriptions are cross-indexed so that the various interrelationships are clearly evident. Also included are the documentation pages for various public software distributions. Links to other related projects at Delaware and elsewhere are also included on the various pages. Hopefully, the organization of these pages, which amount to a total of about 300 megabytes of information pages and reference documents, will allow quick access to the latest results and project status in a timely way.

Following is a retrospective list of papers and reports supported wholly or in part on this project and the immediately preceding project “Scalable, High Speed, Internet Time Synchronization,” DARPA Order D012. The complete text of all papers and reports, as well as project briefings, status reports and supporting materials is at www.eecis.udel.edu/~mills.

7.1 Papers

1. Levine, J., and D. Mills. Using the Network Time Protocol to transmit International Atomic Time (TAI). *Proc. Precision Time and Time Interval (PTTI) Applications and Planning Meeting* (Reston VA, November 2000).

2. Mills, D.L., and P.-H. Kamp. The nanokernel. *Proc. Precision Time and Time Interval (PTTI) Applications and Planning Meeting* (Reston VA, November 2000).

3. Mills, D.L. A brief history of NTP time: confessions of an Internet timekeeper. Submitted for publication; please do not cite or redistribute.

4. Li, Qiong, and D.L. Mills. Investigating the scaling behavior, crossover and anti-persistence of internet packet delay dynamics. *Proc. 1999 IEEE GLOBECOM 99 Symposium* (Rio de Janerio, Brazil, December 1999).

5. Mills, D.L. Cryptographic authentication for real-time network protocols. In: *AMS DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 45* (1999), 135-144.

6. Mills, D.L. Adaptive hybrid clock discipline algorithm for the Network Time Protocol. *IEEE/ACM Trans. Networking 6, 5* (October 1998), 505-514.

7. Mills, D.L., T.S. Glassey, and M.E. McNeil. Authentication scheme extensions to NTP. Internet Draft draft-mills-ntp-auth-coexist-01.txt, University of Delaware and Glassey-McNeil Technologies, September 1998, 9 pp.

8. Li, Qiong, and D.L. Mills. On the long-range dependence of packet round-trip delays in Internet. *Proc. IEEE International Conference on Communications* (Atlanta GA, June 1998), 1185-1191.

9. Mills, D.L., A. Thyagarajan and B.C. Huffman. Internet timekeeping around the globe. *Proc. Precision Time and Time Interval (PTTI) Applications and Planning Meeting* (Long Beach CA, December 1997), 365-371.

10. Mills, D.L. Authentication scheme for distributed, ubiquitous, real-time protocols. *Proc. Advanced Telecommunications/Information Distribution Research Program (ATIRP) Conference* (College Park MD, January 1997), 293-298.

11. Mills, D.L. The network computer as precision timekeeper. *Proc. Precision Time and Time Interval (PTTI) Applications and Planning Meeting* (Reston VA, December 1996), 96-108.

12. Mills, D.L. Improved algorithms for synchronizing computer network clocks. *IEEE/ACM Trans. Networks 3, 3* (June 1995), 245-254.

7.2 Technical Reports

13. Mills, D.L. Public key cryptography for the Network Time Protocol. Electrical Engineering Report 00-5-1, University of Delaware, May 2000. 23 pp.

14. Mogul, J., D. Mills, J. Brittenson, J. Stone and U. Windl. Pulse-per-second API for Unix-like operating systems, version 1. Request for Comments RFC-2783, Internet Engineering Task Force, March 2000, 31 pp.

15. Sethi, A.S., H. Gao, and D.L. Mills. Management of the Network Time Protocol (NTP) with SNMP. Computer and Information Sciences Report 98-09, University of Delaware, November 1997, 32 pp.

16. Mills, D.L. A precision radio clock for WWV transmissions. Electrical Engineering Report 97-8-1, University of Delaware, August 1997, 25 pp.

17. Mills, D.L. Clock discipline algorithms for the Network Time Protocol Version 4. Electrical Engineering Report 97-3-3, University of Delaware, March 1997, 35 pp.

18. Mills, D.L. Proposed authentication enhancements for the Network Time Protocol version 4. Electrical Engineering Report 96-10-3, University of Delaware, October 1996, 36 pp.

19. Mills, D.L. Simple network time protocol (SNTP) version 4 for IPv4, IPv6 and OSI. Network Working Group Report RFC-2030, University of Delaware, October 1996, 18 pp.

20. Mills, D.L. Simple Network Time Protocol (SNTP). Network Working Group Report RFC-1769, University of Delaware, March 1995, 14 pp.

21. Mills, D.L. Simple Network Time Protocol (SNTP). Network Working Group Report RFC-1769, University of Delaware, March 1995, 14 pp.

7.3 Internet Drafts

22. Mills, D.L. Public-Key Cryptography for the Network Time Protocol. Internet Draft draft-ietf-stime-ntpauth-00.txt, University of Delaware, June 2000, 36 pp.

23. Mogul, J., D. Mills, J. Brittenson, J. Stone and U. Windl. Pulse-Per-Second API for UNIX-like Operating Systems, Version 1.0. Internet Draft draft-mogul-pps-api-05.txt, Compaq Western Research Laboratory, August 1999, 30 pp. (obsoleted by RFC-2783)

24. Mills, D.L., T.S. Glassey, and M.E. McNeil. Authentication Scheme Extensions to NTP. Internet Draft draft-mills-ntp-auth-coexist-01.txt, University of Delaware and Glassey-McNeil Technologies, September 1998, 9 pp. (expired)