2009-03-09

Right on Time? The Security Implications of the Humble Computer Clock

Garfinkel, Simson

CSO
SECURITY AND RISK

From: www.csoonline.com

Right on Time? The Security Implications of the Humble Computer Clock

If your company's computer clocks aren't in sync, forensics, backups, and much more can suffer. Simson Garfinkel on getting the time right.

by Simson Garfinkel, CSO

March 02, 2009

Is the clock on every computer system in your organization set to the correct time? If your answer is no, you're not alone. According to a 2007 study by Florian Buchholz and Brett Tjaden, both professors at James Madison University in Virginia, more than a quarter of the Web servers on the Internet have their clocks off by more than 10 seconds. Making sure that computers are set with the correct time is one of those seemingly petty technical things that can unfortunately have big, negative consequences if not done properly. That's because assumptions about time and its flow permeate modern computer systems—including software, hardware and networking. This is true of desktop systems, servers, mobile devices and even embedded systems like HVAC, alarm systems and electronic doorknobs.

Buchholz and Tjaden studied Web servers because they are particularly amenable to analysis: Every time you request a page from a modern Web server, the server sends back an HTTP header called "date" which indicates the time-of-day for the server's clock. But unless your organization has made an effort to keep time in a precise and accurate way, the chances are very good that you're doing a bad job.

Does Your Server Have the Time?

Having the correct time is important for security because system clocks are used for a lot more than just displaying the current time on your organization's homepage. Web servers write the time of every page downloaded into their log files: If the time-of-day clock is wrong, the log files are also wrong. (See also Another Look at Log Files.) This can be a problem if you are trying to figure out an attack that originated inside your organization. You may want to correlate file accesses with whether or not a suspect was seen at a desk, in a meeting or was known to be out of the office. But it's also important for attacks that originate outside your organization. That's because many attackers will mount their attacks from dynamically assigned IP addresses; if your time is off by a few minutes (or more), it may be nearly impossible to figure out where the attack came from or who was responsible.

Log files are just the beginning. Every time a file is modified, accessed or has its metadata changed, modern computer systems will update the file's so-called "MAC times." Forensic tools like EnCase, FTK and Sleuth Kit have the ability to read all of the MAC times within a computer system and sort them to create a single time line. Incident response teams will typically use these time lines to figure out which files an intruder browsed or modified. (See also Richard Bejtlich's Incident Detection, Response and Forensics.)

Because clocks are so often set incorrectly, some forensic tools will allow the security practitioner to enter a time offset or "delta" when a log file is constructed. But these tools assume that a computer's time offset is constant—that if the computer was 30 minutes slow today, it was also 30 minutes slow three months ago. Unfortunately, that assumption isn't valid.

During their six-month study of more than 8,000 Web servers, Buchholz and Tjaden found that systems with the wrong time frequently drifted—or jumped—in unpredictable ways. Some systems would get steadily slower or

faster, and then jump back to the correct time. Other systems were rock solid in the rate that time passed, but they were off from the correct time by minutes, hours, days or even years. Some systems followed the wrong rules for Daylight Savings Time. And some servers appeared to have multiple wrong times—that is, one query to the server would return one time offset, and other query would return a completely different time offset, and then subsequent queries would alternate between the two. (The authors hypothesized that these situations happened when two or more physical machines with different time offsets were hiding behind a single IP address through some kind of load-balancing arrangement.) You can read the entire article at www.dfrws.org/2007/proceedings/p31-buchholz.pdf.

The system's clock is used by many other processes and systems on a typical server. Since many tasks on a server are keyed to the time of day, a server whose time is wrong or erratic may not perform automatic routine maintenance like accounting, scheduled cleaning of temporary files or rebuilding of system databases. Backups may not be performed or they may be inaccurate. Security patches may not be properly applied, automatic update scripts may not properly run. If the time is wrong, the entire server is potentially suspect.

Client Time

Getting the time right on your clients is important too—and not just so that security patches get properly installed. The SSL security protocol, the basis of secure Web browser and mail downloading, requires that your client knows the correct time. That's because SSL is based on X.509 public key cryptography certificates, and every SSL certificate has two time and date stamps inside—when the certificate starts being valid and when the certificate expiries.

Time shows up in many other desktop applications. Many calendar programs display the current date in a different color and have a button that moves the calendar's display to "today." Many mail clients will change the way that date of incoming mail is displayed depending on whether the message was received today, yesterday or some other day in the past. These features won't work properly if time isn't set right.

Many e-mail clients automatically classify messages "from the future" as spam; spammers may set the date header in their message to be hours, days or weeks in the future in order to make their spam messages appear at the top of the user's mailbox. So if your computer's clock is in the past, it may think that every message it receives is spam.

Many standalone devices also have a built-in clock, which are frequently wrong as well. Setback thermostats that have the wrong time may make you inappropriately hot or cold. Recently, I had to consider the log of an electronic door lock to figure out who had entered a room—and when. The log took a lot longer to parse than it should have because it was off by an hour and 15 minutes and because it didn't adjust itself automatically when Daylight Savings Time rolled around.

How to Get the Time?

Getting and keeping accurate time actually requires two independent operations. First, the computer needs to have some way of knowing a precise point or mark in time; second, the computer needs to be able to adjust the frequency or drift of its clock to stay in sync with some presumably more accurate external source. Fortunately, having accurate time is so important that Windows, MacOS, Linux, PalmOS and practically every other modern operating system has built-in support for the Internet Network Time Protocol (NTP), which performs both of these features automatically.

But even though support for NTP is widely deployed, many systems have it disabled. Worse, few systems will alert when NTP is turned off. And worse still, the systems won't alert when their clocks are obviously wrong, even though such an alert would be an easy thing to do. While working on this article I checked NTP on five Macs, two PCs and a Linux system: Only one Mac and one PC had NTP enabled. The other systems were off by minutes, although the Linux server was off by more than two hours. Whoops.

When it's working properly, the NTP system should perform two related functions. When the computer boots it should ask a remote "time server" for the current time and set the computer's clock accordingly. And once the system is running, NTP should periodically monitor the remote time server and gently slow down or speed up the computer's clock so that it stays accurate and so that there are no sudden jumps in apparent passage of system time. (At least this is the way that it is supposed to work in practice. Some implementations crassly reset the system's clock to the reference clock. I've seen this cause problems with applications like Palm's desktop calendar.)

Microsoft and Apple both operate their own time servers and the names of these time servers are built in to their respective operating systems. MacOS, for example, will use the sever "time.apple.com" in North America. Many universities and businesses operate their own time servers as well: Using a local server can both give you more accurate time (because there is less network delay) and can cut down on network traffic from your organization. If you want to run a local time server, you can get the time from one of the public NTP servers operated by the NTP Pool Project (www.pool.ntp.org). In January 2009, there were more than 1,734 public servers operated around the globe, mostly in Europe and the U.S. There are detailed instructions on the website for configuring most operating systems to have accurate time.

Ironically, most of the time servers on the Internet get their time from other time servers. But NTP also has support for so-called "stratum-0" time devices, which get their time reference from one of the agreed-upon time standards. These stratum-0 devices connect to stratum-1 servers on the Internet. Servers that get their time from stratum-1 servers are called stratum-2 servers, and so on. When I wrote this article, "time.apple .com" was actually four separate stratum-2 servers, which presumably connect to other stratum-1 servers inside Apple.

If you are paying attention closely, something about the previous paragraph should have troubled you—that bit about "agreed-upon time standards," with emphasis on the plural. Although it seems like there should only be one time standard, sadly there are multiple ones. The official U.S. time is operated collaboratively by the Time and Frequency division of the National Institute of Standards and Technology and the Time Service Department U.S. Naval Observatory. Both of those organizations operate their own highly accurate clocks and compare them once a week; the two clocks are typically within 20 nanoseconds of each other, which is good enough for most applications. The time is available on the Internet, the telephone system and transmitted on three radio stations (WWVB, WWV and WWVH). If you have one of those clocks that sets itself by the radio (or by an "atomic clock"), it's probably listening to WWVB. U.S. Government time is contributed to UTC time, also known as Coordinated Universal Time, GMT (Greenwich Mean Time) or Zulu time.

But there are other time systems out there. For example, there are many low-cost GPS receivers available that will provide the time to your computer. There are also cellular receivers that will pick up the time from Sprint or Verizon, since the CDMA telephone system that those companies use requires accurate time as well. Unfortunately, each of these systems is slightly out of sync with each other, but in practice, this really won't affect you most of the time. (Several years ago I noticed that Sprint's CDMA system in Boston was transmitting a time that was precisely five hours off; it looked like somebody had not properly set the time zone offset. The problem wasn't corrected for several hours.)

Leap Seconds

There is one more geeky little wrinkle in time that might affect you, though, and that is the handling of leap seconds. Recall that leap seconds are added because the Earth's rotation is slowing down due to the frictional action of the tides; the Earth hasn't had 86,400 seconds in a day (the old conventional measure) for more than a hundred years now. To deal with this unfortunate circumstance, the International Earth Rotation and Reference Systems Service, a group also known as the Time Lords, add a "leap second" every now and then to keep the meteorological day in sync with the day that our computer systems all use. We just had a leap second this past December. The standard way that computers handle leap seconds is to have the clock go to 23:59:60 GMT before they go to 00:00:00 on January 1st. (In New York City the leap second actually happened at 18:59:60 EST on December 31st.)

Leap seconds can cause problems because even though NTP and the lowest layers of most modern operating systems know that seconds sometimes go from 0 to 60 (and not their normal 0 to 59), few programmers are really up on all of the ins and outs of proper time keeping.

This past December, systems running Oracle Cluster Ready Services (CRS) clusterware crashed at 23:59:60 GMT, unable to handle the leap second that bubbled up from the operating system's underlying time service. Some Linux systems from Slackware, Debian and Red Hat also hung, apparently because of an underlying kernel bug. (This is unrelated to the bug that caused some Microsoft Zune players to crash on January 1st, 2009. That bug had to do with the fact that 2009 is not a leap year.)

Hopefully your good-natured response to this article will be to check and make sure that all of the computer systems in your organization have the correct time—and if they don't, add proper time keeping to the list of responsibilities for your security staff. Certainly having dependable time is important for good security, but it also makes other kinds of routine tasks like diagnosing e-mail delays and outages easier.

Ultimately, time is a security matter. Having correct time can be the difference between having someone convicted of a crime and having them go free. Indeed, if your system clock is wrong, you might not even know that a crime has taken place. ##

Simson Garfinkel is an associate professor at the Naval Postgraduate School in Monterey, Calif., and an associate of the School of Engineering and Applied Sciences at Harvard University. The views and opinions expressed in this document represent those of the author and do not necessarily reflect those of the U.S. Government or the Department of Defense.