Time Insecurity and the Network Time Problem~Part 1

Author - Paul M. Wright for UKCERT™ first edition 21 March 2005, second edition 2005-09-29, third edition published October 30th 2005-10-30

1. Executive Summary

This paper is about the urgent need for increased time accuracy and synchronisation in computer networks in order to reduce vulnerability to an electronic attack and to enable subsequent identification and prosecution of illegal attackers after a hacking incident. The following paper highlights time problems in computer hardware, network security and proposed changes to international standards [1]. Some time attacks will be summarised and defence via network time synchronisation will be explored. Future trends will be predicted and recommendations proposed before concluding that increased computer time synchronisation is good for computer security and would be hampered by a forced change in the process of time keeping[2] which seeks to divorce the millennia old marriage between the movement of our planet around the sun and the measurement of human time[3]. This large and complex issue will be approached from the viewpoint of the computer security industry which is partly dominated by the IBM clone PC and Microsoft™ Windows client operating system.

2. IBM clone hardware time inaccuracy

If you are viewing this using a Windows Operating system on a PC it will be interesting to start with an experiment. Click on the clock in the bottom right corner of the screen and count along with the seconds and you will notice that the seconds are not seconds at all. This is the tip of the iceberg of the Network Time Problem as your PC not only cannot keep seconds to a regular pulse but is also very inaccurate at keeping time in general. In fact according to NIST[4] which provides, standard Internet time signals, all IBM clone machines are inaccurate to an average of plus or minus 10 seconds each day. The time inaccuracy is due to the low quality BIOS clock that most PC's come with as standard. In

[1] http://www.ras.org.uk/index.php?option=com_content&task=view&id=830&Itemid=2
[2] http://www.ucolick.org/~sla/leapsecs/nc1985wp7a.html
[3] http://www.itu.int/events/eventdetails.asp?lang=en&eventid=7289
[4] http://tf.nist.gov/timefreq/service/pdf/computertime.pdf

fact the original IBM Personal Computers with MS-DOS did not come with a clock built in at all and the time had to be set manually each time the machine was started. The fact that software running on the PC is now controlling many employees personal and business lives is of great concern.

3. Software time inaccuracy

PC software relies on the BIOS clock when the machine is switched off and then synchronises with this clock when it is switched on. Due to the BIOS clock inaccuracy, software companies have not been very diligent in the timekeeping of the software that runs on an IBM clone PC and related computers. For instance the author has discovered a design flaw in Oracles database logging system that makes it report TIMESTAMPs inaccurately. This has been recognised by Oracle and will be discussed in the Oracle Database Forensics section later in this paper. The problems of hardware and software keeping good time have partly been addressed in the form of networked time protocols, the most popular of these being Network Time Protocol (NTP).

4. Network Time Protocol

The NTP protocol (RFC 1305) works over UDP port 123 and is currently at version 4 which has been stable since the early 1990s. NTP uses a networked time signal that originally comes from a stratum 1 server which should be a very accurate time source reference. Time then filters down from stratum 1 to lower stratum 2, then 3, 4 up to a potential limit of stratum 16 which is rarely used. The system can be reciprocal and works on an algorithm that allows an average time to be calculated from different sources but essentially relies on a trust relationship between the receiver of the time signal and the sender.

5. Problems with NTP

1. Firewall administration's understandable reticence to open UDP port 123 on the perimeter to a public NTP server on the Internet.
2. Network administration's understandable reticence to trust the network time of an external time source.
3. The possibility that the source of the time signal could be spoofed, particularly as communication is over UDP, resulting in an incorrect time being utilised.
4. The possibility that UDP port 123 could be subjected to a DoS attack, therefore preventing time synchronisation.
5. The possibility of a remote exploit that could give external access to the internal NTP server.[5]
6. NTP version 3 and SNTP have no built in security. Version 4 can optionally be secured but the balance is that encrypting traffic and or verifying checksums is going to slow down the transfer of packets therefore making the system inaccurate. Therefore most NTP systems are not secured.

If an external attacker can spoof a signal from the time server that the company uses then they could send an incorrect time signal. The usual mechanism for NTP server identification is via hostname through the DNS system. The reason for this is that the supplier of time may change their IP address. So the first step is for the attacker to

[5] http://www.ciac.org/ciac/bulletins/l-071.shtml

identify the NTP server for the organisation. This can be done using the *ntptrace* command as below which shows a stratum 1 server.

```
root@localhost:~$  ntptrace ntp.cis.xxxxx.ac.uk
ntp.cis.xxxxx.ac.uk: stratum 2, offset 0.001117, synch distance 0.018009
ntp2-rz.rrze.xxxxxxx.de: stratum 1, offset 0.000000, synch distance 0.000000,
refid 'GPS'
```

However most NTP servers no longer allow this functionality, which can be confirmed by going through a list of public time servers and trying the ntptrace command. This is important in a commercial situation where the established practice has been to synchronise to three stratum 2 NTP servers and take the average. If they are all running from the same stratum 1 server source upstream then there is no "strength in variety" and the average of downstream servers will be meaningless. Hence the need for some kind of human communication between the NTP server provider and the receiver to ascertain the upstream source is different from the others. Either that or synchronise directly to three stratum 1 servers. One problem with this is that in the UK there are only two official, publicly accessible stratum 1 servers available according to http://ntp.isc.org/bin/view/Servers/StratumOneTimeServers.

There are, however, many unofficially recognised stratum 1 servers which leads us to the main Achilles heel of the system. Anyone is able create a top level Stratum 1 server using tools such as XNTP, available from http://www.five-ten-sg.com/. XNTP runs on Windows very easily as shown below via the ntptrace command on a stratum 1 server created by the author in a few minutes.

```
C:\Documents and Settings\Administrator.SERVER.000>ntptrace 127.0.0.1
localhost: stratum 1, offset 0.000000, synch distance 10.86559, refid 'LOCL'
```

The low barrier to setting up a stratum 1 NTP server has caused problems for organisations wishing to have synchronisation. First of all it is relatively easy to setup a spoofing NTP server and since the protocol is UDP, no three way handshake is required to confirm the sending IP address. Crafting a packet that sends the incorrect time to an SNTP client is trivial. GUI based packet crafters such as NetDude[6] by Christian Kreibich and spoofed packet sending tools like TCPReplay[7] allow for easy creation of an NTP packet that has an incorrect time and spoofs the source IP address of a real and trusted NTP server.

This problem is exacerbated by the fact that Windows clients use Simple Network Protocol or SNTP based on the older NTP version 3 which only has the option of symmetric key cryptography and so faces the practical problem of secure key distribution. Windows time service may be a possible future target for attackers but at this time it is worth outlining the reasons why an attacker may maliciously wish to alter the time of a computer or networked system.

[6] http://netdude.sourceforge.net/
[7] http://sourceforge.net/projects/tcpreplay/

6. Why is time accuracy and synchronisation important?

The main reason it is important for network administrators to keep well synchronised time on their computer networks is that it will enable the admin to monitor events that occur in real-time and after an incident. Despite the work of leading security researches such as David and Mark Litchfield[8] who report new Zero day exploits to Vendors confidentially before knowledge of these exploits become public, there is still a great threat from less scrupulous attackers. The inability to combat an unknown future zero day necessitates disciplined network logging and therefore synchronisation. It can also be preventative. As an attacker may be able to ascertain the level of time synchronisation of a network using the ping –s command and therefore deduce whether there is likely to be effective log correlation in place. This timestamp is measured as time from midnight and can be converted to human time.

```
C:\Documents and Settings\Paul>ping -s 1 127.0.0.1
Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
   Timestamp: 127.0.0.1 : 82226980
```

GMT/UTC time, for many end users, is increasingly being reported and controlled by their computer and organisations are at risk of an attacker changing the times on their networks in many ways which are variably sensitive, as I shall exemplify.

There are three main factors that I have identified for measuring the time security requirement of computer networked services. Firstly how internally synchronised does the time need to be, secondly how close does the internal synchronised time need to be to GMT/UTC external standards and thirdly how safe the time is i.e. how safe the time should be from attack? I will call these “internal” and “external” synchronisation accuracy and “time integrity” respectively. Values assigned to these labels are subjective and variable to individual circumstances so the judgements given below are general guidelines.

1. **Collaboration software-** Microsoft Outlook, Exchange, Project Manager, Oracles Collaboration Suite and Lotus Notes collaboration software are utilised to arrange meetings and book resources. External and internal sensitivity should be to the second to allow one to sort messages in the order that they were sent, so that meaning can be inferred.

2. **Expired software licenses -** Often when the year is moved back a license will be made to last longer but if it is moved forward, a yearly license may run out and is often difficult to re-activate without contacting the software manufacturer. Software licence keys could benefit from being able to verify time more accurately than the BIOS clock as this may be set backwards. Also if the BIOS clock is changed it should be able to compensate and not lock out. The accuracy of internal and external synchronisation is not the most important point here, but integrity of the time is a high factor so that software is not maliciously locked out by an attacker and fraudulent use of software out of the licence period is prevented.

[8] http://www.ngssoftware.com/index.htm

3. **Certificates expiration-** Time limited certificates can have strength of encryption linked to the length of time that they will be used. Time synchronisation does not need to be highly accurate but the ability to verify and guarantee time integrity is very important. If a certificate has elapsed by a long time then an attacker has had a longer time to break its encryption.

4. **Account expiration-** Again the strength of the accounts security is partly linked to the length of time that the account lasts before the password needs to be changed. If the time can be changed then an account that should not be valid could still be used. Again accurate synchronisation is not so important but being able to guarantee time integrity is important. The strength of a password may be chosen to be appropriate for a 30 day period. But if the password is allowed to carry on for a lot longer then an attacker has longer to crack the password using software like "John"[9].

5. **Allowed logon hours-** Management of logon times relies on an accurate centralised time. For instance if an employee were to get the blame for an IT problem which was not their fault but it occurred at a time when only they had been allowed to logon then it would look like it was them. This would require internal and external accuracy to the nearest minute with high integrity, in most situations.

6. **GPS Software-** GPS relies on an accurate time signal and needs to be externally accurate to the fraction of a second. GPS provides its own time signal and therefore can be the source of good time for the rest of the network. How resistant to tampering this is has not been tested publicly as yet.

7. **Authentication-** Kerberos for instance (see later). This would require internal synchronisation to the minute at least and preferably high external accuracy too. Also high time integrity would be crucial, as an attack which made clients and servers unsynchronised, could lead to a denial of service for logons.

8. **Logging-** Which can include SYSLOG or RDBMS centralised SQL logging. General logging needs to be internally accurate to the second and externally accurate to a second if it is likely to be cross-referenced externally. If the logging host accepts high input from many logging machines into a single file or database then accuracy of TIMESTAMP recording would preferably be to the fraction of a second.

9. **Forensics and Auditing software-** Linked to logs quite closely but includes file systems and live memory and is usually sensitive to the second with a high requirement for time integrity. Forensic evidence can be made inadmissible in court on the basis that the network had incorrect time synchronisation. In the majority of cases correct internal sequencing of events is enough for evidence to be admissible but if evidence has to be correlated from an unconnected network or external source then time inaccuracy makes the evidence less credible in the eyes of the Court. Case law is limited in this field especially in the UK which has been catching up with US cyber law, however the US should not be used as a model for the UK as the US differs from the UK in that it has to contend with 9 different combinations of time zones and daylight saving time.

[9] http://www.openwall.com/john/

7. Time Attacks - a brief history

Computer based attacks with a time element have been a largely overlooked aspect of IT Security especially since the exaggerated effects of the 2000 bug. Previous to 2000 a book called Time-based Security by Winn Schwartau addressed some issues of time in IT Security but since then there has been very little[10]. A notable timing attack on RSA was published by Paul Koch in his paper entitled "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems" in 1998. Since 2000, NTP itself was the subject of an exploitable buffer overflow in 2001 as reported on BugTraq[11]. In 2002, Chris Anley's paper entitled Advanced SQL Injection In SQL Server[12] outlines how blind SQL injection can be carried out using varied timing delays. Time delay in password checking whilst logging via SSH can also be used to verify if a user account exists when PAM is being used as reported to the Debian bug list in June of this year[13].
Subtleties of time in security have been utilised in tarpit products like LaBrea which deliberately delay a network user[14]. These attacks are interesting but there have not been many reported attacks to network time synchronisation especially considering the weakness of NTP and SNTP. This is an area of future potential attacker activity as the amount of chaos caused, compared to the effort required is in the attackers favour.

The most well known time hack is the replay attack in which users credentials are captured on the network segment and then replayed by the attacker at a later time by the attacker to gain a logon. This attack partly relies on the network not being able to recognise that the time has changed relative to the timestamp of the replayed packet. An example of an authentication system that uses time to stop replay attacks is Kerberos and represents one of the most worrying potential time attacks.

8. Authentication time security

Authentication mechanisms such as Kerberos[15], which underpin Microsoft's Active Directory, are secure largely because they control time in order to prevent the classic replay attack as described previously. Kerberos prevents a replay attack by encrypting the current timestamp into the login requests. If this is replayed it will fail as the time has changed. However Kerberos's use of time is more deeply embedded than this. The central Kerberos Authentication Server is only used sparingly so keeping it secure. It is the Kerberos Ticket Granting Ticket from this server that is used by an account to create session keys for each individual networked service interaction. The Ticket Granting Ticket is time limited, which means by the time it is broken it will have expired or, if it was broken, will be limited in the damage caused over that short time it has left before expiration. If time is not sufficiently synchronised within an organisation, Kerberos authentication will not work correctly. By default a Windows XP machine will not be able to logon via Kerberos if there is a greater than 5 minute discrepancy in time synchronisation. If time synchronisation were to be sabotaged by an attacker, Kerberos would fail, resulting in a denial of service. Practical examples of time based attacks and a more detailed history of how time has been used for attack purposes is the subject of a follow-up paper. This paper is mostly concerned with defence through synchronisation and the effect of changing time standards on that defence. Using time based records for defence is the subject of Incident Handling Forensics.

[10] http://www.amazon.com/exec/obidos/tg/detail/-/0962870048/002-1898212-6932038?v=glance
[11] http://www.securityfocus.com/bid/2540/exploit
[12] http://www.nextgenss.com/papers/more_advanced_sql_injection.pdf
[13] http://lists.debian.org/debian-ssh/2005/06/msg00101.html
[14] http://labrea.sourceforge.net/
[15] http://web.mit.edu/kerberos/www/

9. Database Forensics and time insecurity

The author's personal expertise is in Database Forensics of which he authored the first paper[16] which I will now use to exemplify the time insecurity of computer based products. Database Forensics is interesting as Microsoft are in the process of making their client file system into an SQL Server database file system (WinFS). At the same time many organisations networks are moving to thin client architectures enabled by reliable, fast gigabit Ethernet and enhanced centralised storage which means that networked databases like Oracle are increasingly being used to store client side information that was formerly on the workstation. Therefore user file data in the future is likely to be kept in a database of some sort either at the client or server ends.

Recent database hacks have included the theft of 32,000 people's security numbers, all of whom had to be contacted individually (SB 1386) and an attempt at the biggest bank robbery in British history. In order to catch the culprits, forensic incident response personnel require the ability to cross reference database logs from many systems, organisations and time zones using time as a central reference. This is incredibly difficult with the current level of timekeeping inaccuracy and synchronisation on computer networks and software.

Time accuracy is crucial for a forensic incident handler. The separate events that comprised an incident whether it be a hack, loss of data or internal accounting irregularities all need to be plotted on a timeline that can integrate information from different technology sources in order to accurately deduce a sequence of actions. Unfortunately this is often impossible due to the lack of time synchronisation. Localised sequence identifiers (incrementing numbers) are used to compensate for time inaccuracy on email servers, log hosts and databases, but when these sequence identifiers are integrated they do not interleave correctly due to the lack of a centralised sequence identifier which on a disparate network can only be time (UTC/GMT).

Many other IT security professionals such as Marcus Ranum are interested in the way that logs can be aggregated to trace a sequence of events[17]. Marcus has corroborated the fact that in most forensics cases, the external synchronisation of timestamps from IT systems can not relied upon. Therefore different log sources have to be skewed time-wise, in order to compensate for variations before aggregating them. The analyst will test the source machines time compared to the centralised log host and build the difference into their analysis of the combined logs. Most US court cases have only required that the sequence of events is correct and not necessarily externally synchronised or even internally synchronised as they can be skewed back later by the analyst. However this does rely on the time not being changed by the attacker. Once an attacker has gained root access to a machine changing the time will make the skewing process almost impossible. Therefore triggering auditing events based on a user changing the time is a good defence tactic. UK case law should develop slightly differently from US case law as the UK has only one time zone and therefore the ability to synchronise to a higher degree within its own legal and geographic barriers.

The role of accurate time in forensic log analysis becomes even more interesting when using an SQL database to hold the integrated log files of separate systems. Using SQL for centralised logging makes sense as queries can be built that allow accurate analysis of integrated logs in an automated fashion. A centralised log storage/analysis database with

[16] http://www.giac.org/certified_professionals/practicals/gcfa/0159.php
[17] http://www.sans.org/sans2005/description.php?tid=57

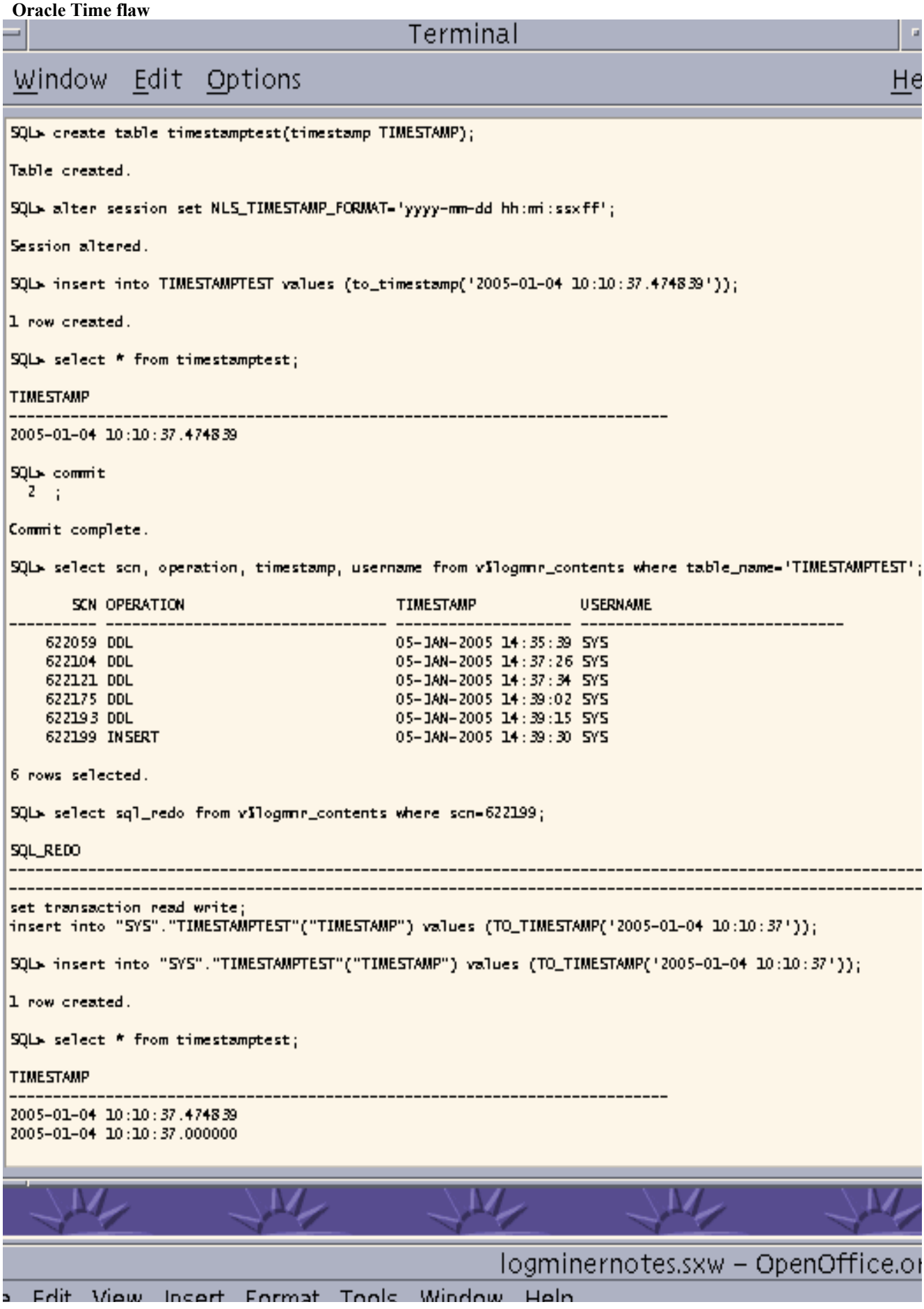
high insert performance[18] can record the log entries in the order they are inserted, which is irrespective of the time configured on the dispersed systems, because databases like Oracle add a sequential identifier to a committed record. The inserted log record can also record the database's own timestamp at the same time. The database timestamp is the baseline and unifies the various logging systems together to form the timeline of an incident. The Forensic Analyst knows the log records are sequential but when they wish to locate a record using the actual real time they cannot because the database time will often be inaccurate to the "real" GMT/UTC time due to both hardware and software deficiencies previously described. An example of this need would be when the sequence of an email and a mobile phone communication need to be ascertained. This requires a strong centralised time line across companies, technologies and maybe time zones. A databases inability to record and refer to external time accurately can partly be blamed on the underlying hardware but is also down to design flaws in the database itself. A time based design flaw in Oracle was found by the author in the LogMiner tool that is provided to analyse Oracle logs. This tool does not report TIMESTAMPs to their stated precision, loses all fractional second data of the recorded TIMESTAMP and incorrectly rounds fractions of seconds to zero. Therefore if LogMiner were being used to mine the logs contained in a centralised logging host on Oracle, all the TIMESTAMPs would be incorrectly rounded to zero. This breaks forensic rules of data integrity and time accuracy/precision.

The problem described is shown in the screenshot below. The TIMESTAMPTEST table is created and then timestamps inserted. When these are viewed or recovered using the LogMiner tool, the reported timestamps no longer contain the fractions of a second.

Intentional blank space

[18] Need to be able to insert very quickly. Many DBs are optimised for select not insert. Enterprise DBs work well.

Oracle Time flaw



This time Bug has been raised with Oracle with reference 4137048 and is indicative of the current low level of time-keeping in even our top class enterprise products. This is further discussed in an Oracle forensics paper by the Author[19].

10.A cure for inaccurate /unsynchronised network time

The ease of setting up NTP has encouraged many organisations to provide their own NTP source internally to gain self control and avoid opening ports on the external firewall.
The major disadvantage of setting up an internal NTP source is that the internal time could drift from the GMT/UTC standard. Interestingly this is not usually the highest priority. The overriding requirement from a network authentication perspective is to synchronise the network with itself in order for Kerberos to work. Internal time keeping entails setting up a source of accurate internal time and synchronising a stratum 1 NTP server to this source without the need for any external NTP transfers. A Windows environment would normally then configure an Active Directory PDC as a stratum two NTP server feeding to the SNTP clients. From an administration perspective, cross referencing audit activity within a single organisation whose time is synchronised with itself (but not with GMT/UTC), is straightforward, however it is much more difficult between organisations or separate Strategic Business Units of large organisations which cannot use the same single source of internally generated time for instance in between two separate Active Directory Forests. Therefore large companies need to synchronise to an external time standard such as GMT/UTC. Sourcing of external time used to be done mainly through radio signals but now more commonly via the Internet from NIST [20] or through a satellite signal linked to the GPS network. The problems of Internet time synchronisation have already been discussed. In the case of GPS though, users in Europe and surrounding areas will have enhanced GPS satellite options following the launch of the Galileo GPS system[21] which will provide accurate synchronised time to Europe. A potential issue with satellite synchronisation is that an aerial/dish is needed to receive the satellite signal which is susceptible to an external physical outage when the dish breaks by malicious damage or through bad weather. This can be solved by a dedicated stratum one server that utilises CDMA signals from a mobile phone network accurate to microseconds[22]. It would be advisable to set up two or three different internal stratum 1 time sources so that they can average out between themselves and provide redundancy. These precautions would enable a large distributed organisation to secure their time with a high degree of certainty. The problem of unsynchronised times is not due to the lack of available protection mechanisms. In the authors experience many organisations are ignorant of the need for accurate and synchronised time to enable a secure network that can be accurately monitored, audited and forensically investigated. The evidence for this assertion can be seen by comparing the time on different clients, servers, databases and clocks in your organisation especially in the DMZ which is most at risk. The reasons for this are varied including user awareness which this paper seeks to address and the fact that much of the software and hardware used in the UK is designed in the US which at any time has nine different combinations of time zone and daylight saving time[23] to contend with. Web servers are thankfully standardised on W3C logging using GMT time and most other components such as firewalls tend to use UTC which is then localised by the user. It is of great benefit to society to have an agreed standard on time so that our computer networks and associated systems such as CCTV can be synchronised in a way

[19] http://www.giac.org/certified_professionals/practicals/gcfa/0159.php
[20] http://tf.nist.gov/service/its.htm
[21] http://europa.eu.int/comm/dgs/energy_transport/galileo/index_en.htm
[22] http://www.brgprecision.com/endrun.html
[23] http://www.time.gov/

that makes criminal activity more difficult to perpetrate and easier to detect. Synchronisation of computing systems, mobile phone networks and CCTV are of great concern when trying to trace the actions of terrorists potentially targeting governmental organisations and financial institutions for example. To enable global time synchronisation widely accepted standards are crucial.

11.International Time standards

Historically time has been measured using the reference of the sun at its highest point on the zero meridian at Greenwich Royal Observatory in London. This is Greenwich Mean Time or GMT. Since the Earths rotational speed varies, GMT is not exactly consistent, hence the use of atomic time which is related to the constant frequency at which a Caesium atom vibrates and is recorded as an SI unit at the Bureau International des Poids et Mesures[24]. UTC is the central standard that connects earth-bound GMT and Atomic Time, as UTC is measured relative to atomic time but compensated to keep it in line with GMT. This compensation is in the form of leap seconds[25] which have been added on to UTC, approximately every two years since 1972 and have worked well to date. Since the seconds have been added (not taken away) we can see that the Earths rotation relative to the Sun is slightly longer than 24 UTC hours and is very gradually slowing down. Certainly current standards have been able to cope with minor fluctuations and nearly the entire UNIX epoch (1970) has been spent coping very well with the occasional adjustment to keep atomically derived UTC time in line with the actual Earth day represented by GMT.

It is interesting to note, at this stage, a proposed change in the current time standard which has been discussed on the USNO Leap Second mailing list[26] . The proposal is that the last ever leap second will be added at the end of this year (December 2005) and from then on UTC would not be compensated for until at least 2600 when there would be a 1 hour difference between our current earth time referenced via GMT and Atomic time. This 1 hour difference would, it is stated, be added at that point (2600) to bring GMT and UTC (from Atomic time) into line again. This proposal has the disadvantage of changing the current standard process for keeping UTC time in line with the earth day. This will have a negative effect on the improvements that have been made in time synchronisation. It would also result in the point of highest midday sun (currently the Greenwich Zero Meridian) moving westwards and eventually reaching the United States who are the originators of this proposed change in the standard which will be decided upon at the ITU[27] meeting November the 7th 2005[28]. This point has been recognised by eminent computer security researches such as Marcus Kuhn of Cambridge University Computer Security Group[29]. The fact that this change has been proposed without general consultation is causing concern at the Royal Astronomical Society[30]. It does to the Author of this paper as well as I can see no reason for changing the current system which works and is providing a sound basis for improved computer security via computer network synchronisation.

[24] http://www.bipm.org/en/si/
[25] http://tycho.usno.navy.mil/leapsec.html
[26] http://rom.usno.navy.mil/archives/leapsecs.html
[27] http://www.itu.int/ITU-T/tsag/index.asp
[28] http://www.itu.int/events/eventdetails.asp?lang=en&eventid=7289
[29] http://www.cl.cam.ac.uk/~mgk25/time/leap/
[30] http://www.ras.org.uk/index.php?option=com_content&task=view&id=830&Itemid=2

It should be remembered that according to Einstein's Special Theory of Relativity, time is relative to the speed at which one is travelling and it is in fact the speed of light that is constant. Einstein's theory predicted the effect of "time dilation" when travelling at different speeds and this effect has been proven[31]. Therefore using astronomical patterns such as when the sunlight is at its highest point in the sky at a point on the Earths surface is a good a way to measure and calibrate this relative unit called time. More importantly the standard relationship between GMT and UTC has worked well for 33 years and computer network systems based on this stable foundation are moving towards an adoption of accurate time synchronisation which will make our networks more secure as previously described.

So far the main concerns voiced over the proposed change to the standard have come from astronomers such as those at the Royal Society[32] who would be greatly inconvenienced by the change in relation between our time system and the earths rotation. It is true that Stonehenges pillars, Egypts pyramids, every sundial on the planet and multimillion pound telescopes around the globe will be sent out of synchronisation with the sky above. The author Paul M. Wright is an expert in IT Security being the most qualified GIAC[33] practitioner in the UK and it is the firm opinion of this paper that changing the standard would slow down the trend of time synchronisation which has been making our computer networks safer. Looking at a wider view, if the relationship between human clock time (UTC) and the Earths time (GMT) is lost then the time systems that help to protect society such as CCTV and mobile phone networks would be undermined and decrease the chance of apprehending terrorists after an incident especially when many of these systems are integrated to computer networks and their time sources.

This paper has investigated the negative threat of not synchronising time on computer networks but there is also a positive promise if computers can become better at recording time in the form of securing authorship integrity of electronic Intellectual Property as I will describe.

12. The timely evolution of Intellectual property security

The speed of change regarding Internet based electronic intellectual property is awe inspiring. Extrapolating current trends brings us to the point at which an individual can write an OS kernel, a music album, a book or film and then distribute it free of charge on the Internet under their name and gain enough benefit from associated fame, recognition and paid consultancy to cover their costs and provide a good living. This is supply chain disintermediation which is in full swing currently. One barrier to this potentially fast flowing IP evolution is the inability to accurately verify original authorship of the electronic file that embodies this new creation. If some one were to copy and pretend they authored the new electronic book how does one prove it. There are web based archiving sites such as www.archive.org and googles cache can be used to show that a web page existed at a certain time but they are not widespread and detailed enough to be used to prove authorship generally. What is needed is a way to show that an electronic file was created at a certain time and by a certain person that cannot be cheated. It is incredibly difficult to be able to verify the time that an electronic file was created once it is separated from its host operating system. Manual methods such as the GIAC paper repository have

[31] http://www2.slac.stanford.edu/vvc/theory/relativity.html
[32] http://www.ras.org.uk/index.php?option=com_content&task=view&id=830&Itemid=2
[33] http://www.giac.org/

given a way to publish a paper and have the time of authorship and author identity independently verified for posterity.

Another method is to use the Timestamper[34] email service which stamps an email sent to it with an independently verifiable time and a hash that verifies the contents of the email. The email is sent back to the sending email address and can be verified back to the timestamper service at a later date. This service has run since 1995 and will accept encrypted contents for privacies sake. One could also combine an ERL which is an encrypted resource locator like a URL but the patch includes a hash of the contents at the target of the URL/ERL. http://www.cl.cam.ac.uk/ftp/users/rja14/erl3.ps.gz. The ERL gives surety of a published URLs contents.

By sending the contents of an ERL and the ERL itself to a service such as timestamper, thus recording time, location, identity and content we can begin to see that proving electronic authorship becomes more possible. This system has been used by the author for this paper. The filename of the paper includes the MD5 hash sum of the paper itself and the paper has been posted as an attachment to an email to Timestamper which includes a link to this ERL. This enables an author to trust the electronic medium ability to preserve their asset which is the reputation gained from being recognised as the true creator of that electronic Internet based IP. Authors would not require the protection of publishing via paper if this type of system can evolve. The legalities have not been tested yet and it relies on Timestamper being able to say that this ERL and file was sent by email at that standard time. The proposed changing of the standard time-keeping process detailed in the last chapter would cause a set back to the evolution of intellectual property authorship security.

13. The future

The future of time security in the UK is likely to consist of increased satellite synchronisation of time via the Galileo[35] GPS system to be launched soon[36], as well as smaller, cheaper more accurate atomic clocks[37] which can be built into PCs and mobile devices and keep very accurate time without the need to synchronise. Integration of physical security systems with computer networks will drive the need for a central timeline which will hopefully remain as UTC which is kept in line with the earths rotation by leap seconds. As long as the scientific community maintains the stable standards we have had for many years then the future of Time Insecurity on our computer networks can improve.

14. Conclusions

The conclusions of this paper are that accurate and synchronised time is required to secure a network and allow efficient forensic incident handling after an attack. To achieve this, network administrators responsible for security need to pay more attention to time synchronisation. Also hardware and software companies need to increase the quality of timekeeping in their products. Finally international time standards should not be unnecessarily changed causing the earth day to become out of synchronisation with the human time system which is embedded into our societies and cultures.

[34] http://www.itconsult.co.uk/stamper.htm
[35] http://europa.eu.int/comm/dgs/energy_transport/galileo/intro/index_en.htm
[36] http://www.telematicsjournal.com/content/topstories/990.html
[37] http://tf.nist.gov/ofm/smallclock/