Key Issues in the Management and Integrity of Electronic Information in the Commonwealth

A submission by AUUG Inc. to the Commonwealth Parliament Joint Committee of Public Accounts and Audit

December 2002

Contributors:
Anthony David
Roger Denholm <mailto:denrog@canb.auug.org.au>
Graham Menhennitt <mailto:gmenhennitt@optushome.com.au>
Michael Paddon <michael@paddon.org>
Greg Rose <ggr@qualcomm.com>

1. Introduction

AUUG Inc. is Australia's peak open source and open systems user group. Formed in 1975, we represent a critical mass of knowledge in the local computing industry, in particular in the area of enterprise class systems, networks and security.

This paper represents a broad range of views drawn from within AUUG regarding the management and integrity of electronic information in general, and with a particular focus to the needs of the Commonwealth. That being said, it is by no means a complete blueprint to such an enterprise. Rather, we present in depth commentary on several areas that our members through especially apropos:

- Information security: The protection, integrity and access control of our information assets protect us from both accidental and malicious misuse or alteration.
- Data storage: The way we physically store our data has a direct effect on in accessibility and longevity.
- Applications: The applications we choose to use determine who may view and manipulate information. Especially in the public arena, equality of access is key. Furthermore, the obsolescence of applications can be a key barrier to archival storage.
- Policies and procedures: The polices and procedures that direct our management of information assets have at least as much effect on the outcome as technology.

Each one of these areas is explored in more detail below. Should more information be required, please contact AUUG at auug@auug.org.au.

Finally, AUUG practices what it preaches. This document is formatted as a standards compliant XHTML file, capable of being viewed by nearly all web browsers. Furthermore, as it is in a human readable format, it will still be easily understood a hundred years from now.

2. Information Security

2.1 Privacy and Confidentiality

In any situation where a large database holds sensitive personal information, the primary concern (after, of course, accuracy of the information itself) is that the information must be private and confidential. These are slightly different attributes. When the database is a multi-agency government information base, these concerns become paramount; some people would say that the citizens are better served by not having the database at all, than having one that is insecure or open for abuse.

2.1.1 Privacy

Privacy of the information in the database simply means that the information should be protected from unauthorised access. But there are different parties who might attempt unauthorised access:

- Outsiders, who should be denied all access.
- Insiders, who have no authorisation for access, but nevertheless are "inside the firewall" and might have more opportunity for access.
- Authorised insiders, who have access to some of the data but not all of it. For example, a worker with authorisation to access tax records should not, in general, be allowed to access health records.

One of the advantages of departmental level databases is that there will be relatively little opportunity for abuse by authorised insiders. Conversely, though, there will also be little benefit to be gained from automatically correlating information between the databases ("data mining").

Whenever information from multiple sources is to be combined in shared or centralised databases, it is important that access controls be stringent, enforced and audited.

2.1.2 Confidentiality

Confidentiality can be seen as a requirement that different items of information of a sensitive nature should not be linked together unnecessarily. For example, while health data about AIDS patients might be analysed to look for trends in health care, the identity of individual patients is not relevant to this study; conversely, a study about delays in health fund payments to patients might need to look at the names, locations, and ethnic backgrounds of patients but should not be related to their ailments.

The requirement for confidentiality of data extends especially to access to an individual's records. Where possible, information intended for demographic use should be de-identified in the database, so that abuse of confidentiality is simply infeasible.

2.1.3 Auditability

It is not good enough to put privacy and confidentiality safeguards in place and assume that they work. Experience shows that mistakes and intentional abuses will occur anyway. It is vital that strong and independent mechanisms to gather access records exist. These access records will form an audit trail to enable technical or judicial correction when these situations occur.

2.1.4 Encryption and Integrity Protection

Historically, it has often been assumed that data can be stored on a computer system, and that the computer system itself should enforce the security. This is not sufficient any longer. Cryptographic protection of the stored data should be employed, both to prevent unauthorised access to the raw data itself, and to ensure the integrity of stored data. Cryptographic keys for these purposes need to be carefully managed, probably using hardware key mechanisms, and with a key management design that ensures access to critical data will be available in the event of any realistic failure of hardware

components; that is, an appropriately secure key recovery mechanism is required.

2.2 Data Integrity

Data integrity is the requirement that the data present in the database is, and remains, accurate. There are many examples of incorrect data leading to injustices in the U.S. Social Security system, for example.

There are many possible sources of incorrect information. Most are unintentional, data entry errors, misunderstood or inappropriate forms, and so on. Some inaccuracies can be introduced through incorrectly cross-linking data, such as attributing debt incurred by one resident of a house to another resident. In relatively rare cases, records can be intentionally tampered with for malicious reasons.

As above, an extensive audit trail is required to identify the source of erroneous information so that corrective or preventative action can be taken in future.

More importantly, though, mechanisms must exist that allow individuals to identify such erroneous information in the first place. Secondly, once identified, it must be possible for the individuals to submit correct information, which will be used to update the database after it has been appropriately verified. The current wave of "identity theft" makes this both very difficult and vitally important.

On a more mundane level, integrity of data also implies continued accessibility. Backing up data, and regularly ensuring that it can be recovered, should not be underemphasised.

2.3 Online Security

The security of information assets maintained on online systems poses several unique challenges.

1. Authentication of parties accessing data and services remotely.
2. Security and integrity of data as it transits networks.
3. Transitivity of access rights and capabilities.
4. Unexpected synergies between diverse systems.
5. Denial of service.

2.3.1 Authentication

Authentication is the foundation of all secure online systems. Not surprisingly, inadequate or misapplied authentication is a key element in many breaches of security. For instance, a well publicised privacy breach in June 2000 involving the Commonwealth's GST Assist website was the consequence of weak authentication. In this case, the designer of the system decided to only require the user to present a valid ABN to retrieve business details. Given that an ABN is easy to guess, this was a bad idea to start with. However, in the context of an online system it was a disaster since it is straightforward to write a program to try every possible ABN in sequence. This kind of "brute force" attack is only one of several ways by which online authentication systems may be subverted.

System designers have been using passwords and passphrases as authenticators for many years. Consequently, users are familiar and comfortable with this type of mechanism, and their use has proliferated in the online world. Unfortunately, passwords are highly inappropriate in a networked environment due to a broad range of attacks they are subject to. For instance:

- Guessing: Most people are not good at remembering long, nonsensical strings, and tend to choose short, easy-to-remember passwords instead. As a consequence, these tend to be easy to guess, either directly (by knowing something about the person) or indirectly (by trying a

dictionary of common words and phrases).
- Sniffing: Passwords are often transmitted across networks unencrypted, and may be captured in transit by an attacker. Unfortunately many common web and email applications transmit unprotected passwords, leading to key weaknesses in organisational security infrastructure.
- Replay: Even if a password is encrypted in transit, it may be captured in and replayed at a later time. This highlights the need for each authentication exchange to be uniquely different.
- Social Engineering: If users know their passwords, then it is likely in a large population that some will unintentionally divulge it.
- Server Compromise: If the passwords are stored in a central database, then the compromise of this database subverts the entire system and all users. Even if the database is encrypted or otherwise specially protected, it is well worth the attacker's time and effort to compromise. Good authentication systems should minimise such points of catastrophic failure.

The best sort of authentication system is one which has effectively unguessable passwords which the users themselves don't need to remember, is proof against sniffing and replay, and where there is no central database to attack. Authentication based around *public key* encryption exhibit these features, and are widely available. Public key authentication revolves around a long, random sequence of numbers that never leaves the possession of the user, known as the *private key*. For each private key, there is a corresponding *public key* that the user can publish with impunity. The public key can be used to craft a unique *challenge* that only the holder of the private key can correctly respond to.

Public key technology is widely available and AUUG highly recommends the use of this style of authentication technology for access to any valuable or sensitive online information asset or service. One common and widely trusted public key system is known as "RSA". RSA supports keys of various lengths, measured in "bits"; the longer the key, the longer it takes for an attacker to try all possibilities. AUUG recommends that if RSA is used then keys be at least 1024 bits in length, preferably longer.

Because public key systems can provide strong authentication, they can also be used to create *digital signatures* that can be applied to documents with much the same utility as a traditional signature. This facility is particularly useful when transitioning established policies and procedures that require signatures to an online system, and is far more secure than the commonly observed 4 digit PIN.

An extension of public key authentication is the notion of a *certificate*, issued by a *certification authority*. A certificate is simply a users public key that has been digitally signed by a trusted agency to notarise that the user is who they purport to be. Digital certificates make the roll out of strongly authenticated online services easier since the association of public keys to specific users is a key issue. The same certificate can be used for arbitrarily many online services.

While certification brings many benefits, a key weakness is the certification authority itself. Clearly it is a weak point as a compromise may allow attackers to forge certificates. This, however, is not a catastrophic failure as *certificate revocation* mechanisms may be used to recover from such a scenario. Just as important as technical considerations, however, is the issue that the goals (and even operational jurisdiction) of a commercial certification authority may not be entirely aligned with those of the Commonwealth.

The existing Commonwealth Gatekeeper Framework addresses many of these concerns effectively. AUUG therefore recommends that the Commonwealth mandate the use of Gatekeeper compliant PKI infrastructure for the authentication of users of all sensitive online systems.

One arguable weakness in the Gatekeeper accreditation of commercial certification authorities is the inability to enforce alignment with a commercial entity's goals and the broad public interest. While commercial providers are a key part of the online community, they could be balanced by the creation of one or more not-for-profit competitors bound explicitly to public interest goals.

Other, exotic, authentication mechanisms exist. However few are as suited to the online environment as public keys. Biometric authenticators raise privacy concerns, pose expense and difficulty in wide deployment and are impossible to revoke if compromised. Hardware tokens are popular in some segments of industry, but are many orders of magnitude more expensive to issue than public keys, and arguably much less secure. In addition, proprietary authentication solutions may make the Commonwealth's online assets beholden to a third party supplier, which is an unattractive state of affairs at best.

In practice, it is straightforward to use certificates for authentication for basic web and email services, by insisting that they be accessed via the standardised TLS protocol (formerly known as SSL). Most online services can be protected using the same technology. AUUG recommends the use of standard security protocols in all cases in order to maximise availability and interoperability.

Regardless of the authentication mechanism chosen, all online systems must be constructed with the idea of "defence in depth". Once a strong mechanism like certification is used for authentication, the weakest link becomes the user, and the machine on which they store their private keys. Given that people are fallible, there will always be the occasional compromise of a single user in any system, however the system as a whole should continue being quite secure.

2.3.2 Security and Integrity of Data

Data transiting a network is subject to being copied and modified en route. The larger the network, the more difficult it is to secure physically from malicious users. In particular, the Internet must be treated as a totally insecure and hostile environment.

Data may be protected from both of these attacks by a secure protocol that encrypts the data and provides integrity checks. Standardised protocols that perform these functions are readily available, and can be applied to most online services. For example, the abovementioned TLS provides not only strong authentication, but for encryption and integrity services as well. It is supported by most web browsers and many email clients, and is a natural selection for wide scale online services.

In some cases, it is desirable to provide secure access to an entire network as an online service. While this is useful for applications such as telecommuting, AUUG recommends that such practices be constrained as much as possible, due to the difficulty of securing such wide spectrum access. In cases where this is necessary, standard protocols such as IPSEC are recommended.

AUUG strongly recommends that all online access to sensitive materials be via a protocol at least as secure as TLS. Furthermore, we recommend only the use of open, standard protocols due to the immense amount of public review and scrutiny that they have received.

2.3.3 Transitivity of Access Rights and Capabilities

Designers of online systems need to be aware of the issue of transitivity of access rights and capabilities. Attackers commonly choose the weakest server to subvert, which gives them all of the rights and capabilities belonging to that host. They may then use these additional rights to mount a successful attack on a much more secure machine, and so forth. This "stepping stone effect" has been used to compromise extremely well protected central servers and databases.

It is common to see a group of online servers protected by a firewall, with only the web server accessible from the outside. This may lead to a false sense of security, since the web server is often the easiest machine to compromise. Once that has been broken, the firewall is all but useless in preventing additional incursion.

AUUG recommends that highly sensitive data be strictly partitioned from all online systems. Where

it must be available to online systems, a regime of strict review, change control and audit is highly desirable.

2.3.4 Unexpected Synergies Between Systems

One of the effects of the deployment of diverse online services is the possibility of creating new capabilities unintentionally. For instance, the ability to mine diverse databases and correlate heretofore uncorrelated information may allow users to exceed their authorised capabilities in unexpected ways.

There is no technological fix to this problem. AUUG recommends, however, that appropriate reviews be made of each new database to be made available online (or each significant change to an existing online database) to see if it may be subject to such unintended usage. Furthermore, data mining requests may be logged and reviewed regularly for unusual or suspicious patterns.

2.3.5 Denial of Service Attacks

One of the major problems facing online systems, particularly those accessible from the Internet, is the "denial of service" attack. In general, these attacks work by overwhelming a server with bogus requests, reducing its capacity for bona fide users.

There is no simple remedy for these attacks. While strong authentication can distinguish bona fide requests from spurious ones, often the authentication mechanism itself is the point of attack. Sometimes the best that can be done is to choke demand so that at least some users continue to receive service.

Denial of service attacks are often made difficult to trace by the perpetrators providing fake source address information. This is possible in many cases because sites providing access to the Internet do not provide any verification of the origin address of the Internet traffic. One countermeasure that can be deployed is "ingress filtering". This is method of rejecting all Internet traffic entering the Internet via an ISP that does not have a source address of one of the hosts the ISP is providing service for. AUUG recommends that the Commonwealth investigate how ISPs can be encouraged to apply effective ingress filtering in the form of a code of practice, or even a regulation. To maximise the effectiveness of such steps, international cooperation from other jurisdictions is required, clearly making this an area where the Commonwealth can add value.

AUUG recommends that all at risk online systems be reviewed for resilience to denial of service attacks specifically, and for countermeasures to be tailored for each system. In particular, constant monitoring and proactive response by support staff are key ingredients to weathering this kind of storm.

3. Data Storage

3.1 Backup Media

Backup is an essential part of electronic data management. As a senior HP engineer states in his email signature there are only 2 kinds of computer user:

1. Those who have lost data, and
2. those who are going to.

Backup systems minimise the impact of data loss from most causes. It cannot protect against willful

or unintended data entry omission or long term data removal or corruption.

Issues that should be considered when choosing backup media and hardware:

Purpose of backup and storage	Unless the data storage requirements are considered from the daily, weekly, monthly operation needs, annual budgetary cycle, archival, monetary, census and legal views an optimal backup regimen cannot be developed. These will influence the choice of media for backup as well as what is backed up.
Medium	Various tape and disk formats.
Hardware	Purchase cost, availability, maintainability, maintenance costs, longevity.
Media	Unit cost, stability, re-usability, longevity, licensing, availability.
Software	Licensing, cost, availability, support, portability.
Staff	Ability to use information on media.
Long term storage	Cost, security.
Disposal	Cost, security.

The most common medium for operational backup is magnetic tape, now usually in a cartridge form for use in high density drives and robots, used in medium to large organisations. Typical modern examples are DLT (capacity up to 80 GB) and DDS (40 GB). All tapes can hold more depending on the compressibility of the data. Some data, such as new databases and spreadsheets can compress 3 times, other media such as images may not compress much if at all as it is already compressed.

There are newer variations on DLT such as AITor LTO which allow 200GB on a cartridge and very fast access to files on the tape by use of embedded microchips in the cartridge. There are a few other systems in use but these are reducing in number. They used optical media like CDROM or propriety tape hardware and tapes. An extreme example is at AUSLIG where a now defunct product had hardware which stored 2 TB on large reels of tape.

Tape lifespans are limited but DLT manufacturer Quantum claims an archive life of 20 years. See the MEDIA FAQS website for details.

DDS tapes are not considered reliable after a year in storage. DLT tapes can be used 1000 times, DDS tapes re-used 100 times. This has a big bearing on which tape format is suitable for a given application.

With the rise of more visual means of displaying data the available disk capacity will be needed. Organisations are also keeping more data and for longer as operating systems, applications and historical data grow. CASA for instance, keeps a copy of every web page served due to a court decision against it some years ago. As many of these are pages generated by databases and programs the number of web pages stored over a year is significant.

CDROM is a popular medium for medium term storage of archival data or irregularly used data. CDROM specifications conform to a patented standard held by Phillips which ensured portability and economies of scale. It has failed to be a significant backup media due to slowness of writing to disk.

This cannot be said of tapes or the current pretender to succeed CD, the DVD. DVDs have partly stabilised on 2 contending media formats, which is slowing its introduction for backup purposes. There are also other developing DVD technologies with much higher data storage capacities. For instance, Hitachi demonstrated a 120 GB disk 18 months ago. Even so, CD and DVD formats cannot

be said to store much compared to tapes, with capacities of 4.7 GB and 700MB respectively. DVD and CD media currently claim to have an archive life of up to 20 years.

3.2 Disaster Recovery

Despite being the most complex machines created by humans, computers are astonishing reliable when configured correctly with well tested software. Hardware has improved to the point where good quality consumer hardware has reliability and performance only available on very expensive dedicated hardware 5 years ago. Large scale hardware failure is rare and usually due to change control systems failing in the manufacturing stages.

One wishes some common operating systems and software was as good as the common hardware it runs on. Regardless, the moving parts of any system are most likely to fail. This includes the long term data storage device, the hard disk. Human and procedural error may also delete data. However, copying data back onto a running system is simple to compared to rebuilding a destroyed system.

To recover from a total system failure, such as the destruction of a computer system by fire or disaster the following issues must be addressed:

- Location for new equipment
- Power quality and availability
- Communications for networks into site and around recovery site
- Site security
- Access for staff and equipment delivery
- Environment including temperature and humidity control
- Availability of replacement hardware, software and licenses
- Current data archive
- Trained staff to install, configure hardware, software and data
- Trained staff to use applications

Simply having a backup tape with the last application data will not recover a business. Depending on application and budgets, disaster recovery may use all or some of the above points. A small business would have much of a problem relocating to new premises after a fire unless they had unusual requirements. Planning for a major disaster in a datacentre will require considerably more effort and testing that the plan actually works. The cost of failure would destroy the business.

For large organisations, redundancy in plant and equipment can be a deliberate design decision to ensure alternate premises are available if one site becomes unusable. Applications and customers will be ranked in business/contractual/monetary importance so that critical systems can be loaded onto machines with unused capacity or warm spares at a remote location. Modern disk hardware is making this process potentially transparent by having mirrored disk storage. The mirrors are at separate locations. Loss of one site allows failover to the other site. Some sites replicate transaction logs. A good example of doing this with open source is the use of rsync by GM to keep a terabyte database replicated remotely.

The most important part is the availability of readable complete backups of the data used by the organisation. This is usually done with off-site storage of backup tapes. Software is usually replaceable and large commercial vendors keep records of a customers licensed software. This allows licences to be re-issued.

Obtaining hardware is the most time consuming issue. PCs and small servers are an off the shelf item. Large servers and mainframe class machines can take months to deliver. Configuring and installing them can take a week. Installing the software may also take days. All this assumes a site is available. Old hardware adds to the levels of spares required. Currently HP/Compaq are finding it

hard to obtain Vax components after Sept 11 2001 as the loss of so many servers used available spares for a server no longer being manufactured.

The commercial response to this is the availability of Disaster Recovery firms who, for an annual fee, keep a warm site ready for setup for a customer. This enables one DR site to be shared, giving major cost savings.

4. Applications

All data that is maintained by the government needs to be manipulated by computer software. The software falls into a number of categories:

- operating systems - the low level software that interacts with the computer hardware and manages the higher level software
- communications and networking - for transmission of data between computers
- systems software e.g. file copying, backup
- high level applications:
    - generic e.g. word processing
    - custom e.g. tax submission software for Australian businesses

There is often considerable blurring of the distinctions between these categories.

Within any of the above categories, there are a number of parameters that determine the suitability of a particular software package for use in any environment:

- security - is the data sufficiently protected against unauthorised access from both within the government network and via external communications connections?
- verifiable - can the purchaser verify that the software does what it should do, and does nothing that it shouldn't do e.g. "back doors"?
- interoperability - the software should be able to interact easily with other software for transfer and sharing of data
- vendor independence - if the vendor can no longer provide or support the software, existing data can be accessed by software from an alternate vendor
- price - obviously the software purchaser wants to minimise their expenditure:
    - initial costs - the cost of buying or licensing the software initially
    - ongoing maintenance costs - what does the purchaser need to pay to have the inevitable bugs fixed?
    - upgrade costs - do new versions of the software require a completely new purchase or is there an incremental upgrade price?
    - support costs e.g. training of users, system administration
- performance - does the software produce the desired results within an acceptable time?
- suitability - does the software perform the tasks that the user requires it to?
- robustness/reliability - does the software crash often?
- ability to customise - can the purchaser adapt the software to their needs and how much does it cost to do this?
- scalability - if the number of users expands, will the software still perform acceptably (perhaps with some increased hardware outlay)?
- maintainability - can the purchaser modify or fix the software themselves or are they at the mercy of the supplier?

In almost every case, the purchaser will need to make trade-offs between these factors in order to arrive at an optimum solution. However, AUUG believes that it is of paramount importance that the government maximises the first four of these parameters when it controls data that has been

entrusted to it by the public. The government needs to guarantee long term reliable access to this data without requiring the purchase of specific software. They also need to ensure that the data is not susceptible to unauthorised access; including access by the software vendor.

4.1 Interoperability and Vendor Independence

Interoperability is the ability to access data using software from more than one source. This is achieved by ensuring that the data is stored and transmitted using formats and protocols that are standardised and well documented. Standards may be formal (i.e. specified by a standards body like Standards Australia) or informal (e.g. publicly documented by a software vendor like the Java programming language). Using standards avoids problems with data stored in proprietary formats being inaccessible due to patents, trade secrets, or just lack of good documentation. Other sections of this document discuss standard storage formats in more detail. Similar standards should also apply to communication protocols.

As an example, consider the federal government web site. The web site might have been developed using Microsoft FrontPage, published by the Apache web server, and hosted on IBM hardware that is connected to the Telstra communications network. An Australian citizen sitting at their Macintosh computer in their lounge room, can access this web site via the Optus network and view it in the Mozilla web browser. Here we have hardware, software and communications from a number of vendors all interoperating. This is only achievable by using standard protocols (TCP/IP, HTTP), standard hardware interfaces (Ethernet, RS232), and standard data format (HTML).

On the contrary, a document created in a proprietary application such as Microsoft Word and saved in the native format is only accessible using that particular application. In fact, it is possible that it is only accessible using the exact version of the application that was used to create it. Often, the creator of the document will use such an application since it is what is most familiar to them, or even just because it is installed by default on their computer. However, effectively forcing all other users of the document to use the same application (and perhaps version) is not acceptable for a government. Maximising ease of use for a small group of content creators at the cost of accessibility to a large population of consumers is a poor tradeoff.

Vendor independence is another advantage of using standard formats and protocols. Software vendors may go out of business, may increase prices to an unacceptable level, or may decide that it is no longer in their business plan to support the software. If the software has used standard formats for the data, it should be possible to find another vendor who can access that data. At the worst, custom software could be developed to read the existing data. Using proprietary formats, the vendor achieves a lock-in - only that vendor can access the data without considerable effort.

4.2 Software Development and Distribution Paradigms

In order to explain the differences between a number of different software development and distribution paradigms, it is necessary to understand a little of the (most commonly used) software development process. Software is created by programmers who describe the steps to be performed by a program using a high level, source code language (e.g. C, Java, or COBOL). This will usually have an English-like appearance and the program will be understandable by other programmers who are familiar with that language. This source code is then passed through a program known as a compiler to produce the binary code (or machine code) of the program. The binary code is what is actually executed by the computer that sits on the user's desk. In order to modify the program's behaviour - to enhance it or perhaps to fix a bug, the programmer edits the source code and runs the compiler to produce new binary code. Binary code is difficult to understand, and making modifications to the program using only the binary code is generally impractical.

The most common software distribution paradigm is proprietary closed-source software. This is

developed by a commercial entity and sold or licensed in a binary-only package. This means that the executable programs are installed on the purchaser's computers and the source code is kept secret by the vendor. Although there is no direct link between the two, closed source usually goes hand-in-hand with proprietary data storage and communication formats. Not having the source code or documentation of the data formats means that the data is (practically) inaccessible to any software other than the vendor's proprietary programs. Also, not having access to the source code means that it cannot be verified not to contain back doors (covert communication with unauthorised third parties), and it cannot be modified or fixed by the user if bugs are found.

At the other end of the spectrum is the concept of open source software. Here, the software is provided to the user both as a binary program and also as source code. The user is able read and modify the source code at their discretion. Such software may be developed by a company or is often developed by programmers as a hobby. Many open source projects are developed by professional programmers in their spare time. With access to the source code, several important factors become available: interoperability, the ability to verify the software's operation, and maintenance. Even if the purchaser does not have the ability or resources to perform these, they can always contract a supplier to do it for them - something that is not possible with closed source. There is an argument that having the source code available makes software more reliable and secure since there are "many eyes" available to find problems. With closed source, security bugs can be concealed by vendors rather than fixed.

In a number of countries throughout the world, governments are moving towards open source software or at least being encouraged to do so. Some examples are Peruvian congressman Edgar Villanueva Nuñez' Bill to mandate the use of open source software in the Peruvian public sector. Microsoft wrote to him explaining why they believed that his Bill was a bad idea and that Peru should not preclude the use of proprietary software in government. His refutation of their arguments is amusing, comprehensive, and eloquent. In the Indian state of Madhya Pradesh, the chief minister, Digvijay Singh, said "We feel that when we are putting public information out in the open, then it should not be through a proprietary software." Also in India, members of the Kochi Free Software Users's Group have written to their state government with a very similar message.

There are some sections of the information technology community that strongly advocate the use of open source software at all levels of government. There will be cases where there is no suitable open source software available to perform a required task - in these cases, the use of closed source software may be necessary. However, in applications where open source software is capable of achieving the desired result, it should be preferred over the alternative. This is the only practical way to ensure the first four parameters listed above are achieved. Proprietary software vendors will no doubt disagree with this assessment and provide many arguments as to why their software is preferable. It should be remembered that they have a vested interest in selling software to the government. We believe that the examples above - in particular the Villanueva letter - adequately refute them. In fact, this recommendation says nothing to prevent a company from selling their proprietary software to the government - the only requirements should be that the source code:

- is available for verification and auditing by the government or its agent
- is able to be compiled and run (this does not preclude licensing fees)
- can be modified by the government to fix bugs
- can be inspected to determine data and communication formats in order to develop interoperable software

As the previous suggests, there is nothing to prevent a company from charging the government a fee for the provision of open source software. However, much of the existing open source software is available for no cost. Considering the large licensing fees that many software vendors charge, this can be a big advantage. Why should the government use taxpayers' money to buy software when similar or better software is available for free?

Custom software developed specifically for the government is somewhat different. In this case, the government would either develop the software in-house or contract a software supplier to develop it. In either case, AUUG strongly recommends that standards based data formats and communications protocols are used to maximise interoperability. In the case of externally developed software, the government should specify in the terms of the contract that the source code is fully owned by the government. This avoids lock-in with the supplier - any other supplier could be used to perform later modifications to that source code.

5. Policies and Procedures

Effective policies and procedures are required to correctly implement more of the strategies and technologies addressed in this document. In many areas, most organisations either already have adequate policies or can easily turn to external help to create them. Examples are:

- Backups
- Disaster recovery
- Standard operating environments
- Change control
- Physical security

Information security, however, is worth singling out as perhaps currently the greatest policy blindspot of many organisations.

5.1 Security Policies

Historically, enterprises have secured their information assets on an ad hoc basis, generally relying on physical security to prevent compromise. No distinction has been made between information and property of a more tangible form. This model has the great advantage that physical security is generally well understood and relatively simple to implement, however it breaks down when there are non physical paths by which assets may be attacked.

The advent of the Internet has provided a multitude of such paths. It is now considered unusual for an organisation not to be connected to the ubiquitous global network. Usually these connections are made and managed according to a technical or commercial agenda, without a broad understanding that the link creates a completely new landscape of threat and risk. Sadly, the result is often massive compromise.

Writing an information security policy is hard. As a consequence it is common to find organisations that do not have one. If it does exist, often it is inappropriate, inadequate, out of date, or simply ignored in practice. Where there is no policy, strategy is diffuse and tactics have unsustainable outcomes. A policy that is not broadly understood, applied and of use in day to day risk management is even worse: it creates a false sense of security.

Furthermore, there is no such thing as a universal security policy. Each organisation is different, with unique assets and threats. Attempts to fit a ``one size fits all'' policy generally yield poor outcomes: like a poor fitting shoe, the results can be both painful and permanently damaging.

AUUG recommends that each organisational unit maintaining sensitive electronic information or services be required to document, implement and audit a tailored security policy that adequately addresses the protection of those assets. Furthermore, in order to assist the creation of such policies, the Commonwealth may wish to create guidelines and supporting material, while discouraging a "boilerplate" approach.

Good security policies should address at least the following issues:

- Threat assessment,
- Risk management,
- Asset compartmentalisation, and
- Compromise mitigation

Finally, the key to an effective security policy is adherence. Policies should be kept simple an understandable to ensure maximum compliance, and proactive education and regular auditing of the user base is essential.