Increased domain security through application of local security and monitoring

J.P. Morrissey, P.W. Sanders and C.T. Stockel

Network Research Group, Faculty of Technology, University of Plymouth, Plymouth.
E-mail: joem@aldiscon.ie

Abstract: *This paper presents a model for increasing security within a security domain through the use of localised security services and continuous monitoring. The model divides security services between three logical structures Local Security Units, Local Security Servers and Domain Management Centres. The localisation of security allows the functional divisions within organisations to implement modified security dependent upon their individual needs.*

Keywords: computer security, monitoring, local security, CISS

1. Introduction

As networks and the consequential connection of computers with their stored data becomes increasingly common, the security of that data becomes ever more important. Current techniques in security have proved themselves to be ineffective when confronted by credible hackers (Guardian, 1993), (Sunday Times, 1993), (Audit Commission, 1990). As more and more people gain access to networks and the computers connected to them (as current policy within the EC and USA indicates they will) there will be more people tempted to pursue illegal activities. Many may wish to instigate malicious damage while others may only inadvertently cause damage. Whatever the reason, the action is illegal and security upon the computer systems must be capable of dealing with these attempts.

To combat the increased security risks which exist when placing sensitive information upon open computer networks with every increasing numbers of users, new methods for security must be found. This paper uses CISS (Comprehensive Integrated Security System) (Muftic et al., 1993) as its basic security provider. The International Standards Organisation (ISO) have been setting standards for open interconnection of systems, and the CISS model is an attempt to provide the required security as indicated by the ISO within relevant publications (International Standards Organisation, 1986, 1988). CISS is also being developed to provide security for openly distributed environments. CISS is an adaptable security system capable of being added to existing computer systems to bring their current level of security up to a level acceptable by their owners.

2. Comprehensive integrated security system

The CISS architecture was developed to protect computer systems within an open environment against threats of a potentially malicious nature, some of these threats are detailed in Table 1.

To combat these threats, CISS makes available services that can be used to provide the necessary security measures. The security services are provided through the correct ordering and application of the security mechanisms shown in Table 2. The Services can be requested by users, or forced upon users by the security administrator's configuration of CISS according to the security policy in place.

Because a security system is a very complex and potentially large software construction problem CISS has been divided into ten functional elements known as agents, whose interaction is shown in Figure 1. The agents are responsible for the co-ordination, supply and control of security services to users/processes.

2. 1. Security management information base

The security management information base (SMIB) is the CISS knowledge store. It contains information on the configuration of CISS and will specify the selection of mechanisms, services, protocols and the subsequent limitations upon the variables they use as set by security policy. The SMIB contains information on authorised users, authentication data, user entity capabilities and privileges, etc.

2. 2 CISS agents

The ten CISS agents are:

(1) *User Agent (UA).* This is the interface of CISS

Table 1: *Threats.*

Threat	*Description*
Masqueraders	When a legal entity (user/program) impersonates another.
Illegal associations	Where an illegal entity forms an association with a legal entity that violates the authentication and authorisation policies in place.
Non-authorised access	Where an intruder/user gains access to resources that are required for proper operation.
Denial of service	Where a legal entity is denied access to resources that are required for proper operation.
Repudiation	The false denial of a legal entity that has provided a service or resource.
Leakage of information	Loss of confidentiality, anonymity and misappropriation.
Traffic analysis	Deduction of information from characteristics of data transfer.
Invalid message sequencing	Prevention of re-submission of data beyond its valid lifetime and replay attacks.
Data modification	Malicious or accidental modification of data.
Deduction of information	Collection of data from summaries in a distributed database.
Illegal modification of programs	Malicious or accidental modification of software.

Table 2: *Security mechanisms.*

	Security Mechanism	*Description*
En	Encipherment	Application of cryptographic algorithms for confidentiality.
DS	Digital Signature	Use of cryptographic techniques to provide proof of origin.
AC	Access Control	Controlling the access to resources upon a network.
DI	Data Integrity	Algorithms to check the integrity of communicated data.
AE	Authentication Exchange	Techniques by which the identify of an entity can be confirmed.
TP	Traffic Padding	Provide protection against traffic analysis.
RC	Routing Control	Control the path along which data is communicated.
Nt	Notarisation	Provision of proofs.

through which the users can access its security mechanisms and services directly via a request. Dependent upon the security policy some security services will be provided irrespective of the users.

(2) *Security Administration Agent (SAA).* This is the second interface to CISS and is solely used by the security administrator. The division of functionality between the UA and the SAA has come about due to the necessity of making the UA a multi-user agent producing complex code. The SAA produces simpler code for a single user agent.

(3) *Operational Environment Agent (OPENA).* This is the final interface through which CISS communicates with the outside world. Its main function is interfacing the operating system (OS) and applications to the SSA for the provision of security services. The OPENA is made up in part by the Application Programming Interface (API). The interface allows applications to access CISS services giving CISS greater flexibility for implementation within existing systems. The OPENA interacts with the AA and IDCA (see below) for communication with entities external to the local security domain.

(4) *Security Services Agent (SSA).* The SSA is the central agent through which all services are requested. The SSA selects the security mechanisms that comprise a requested security service. With assistance from the security mechanism agent it implements the service.

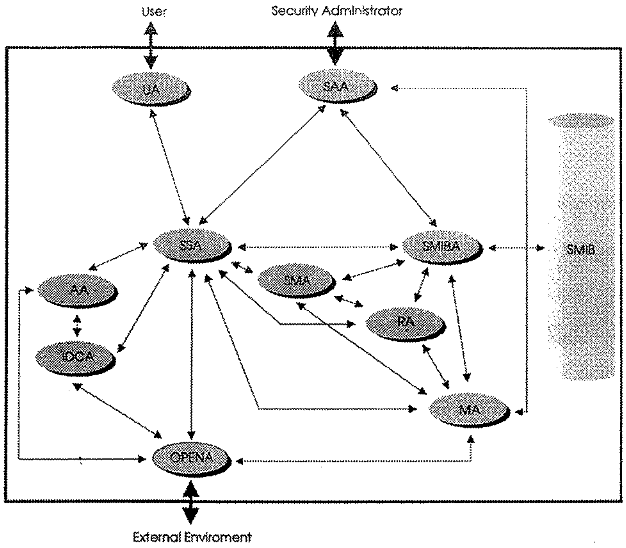


Figure 1: *Agent interaction.*

The SSA must restrict the use of security services by user entities to those that they are entitled to use, either through their privilege or the capability of the system they are using.

(5) *Security Mechanisms Agent (SMA).* The SMA interfaces the security mechanisms to the SSA for the provision of security services. Therefore, it accepts control and data commands from the SSA. The SMIBA provides information on the parameters that a mechanism can accept; for example, depending on the security policy in place, an RSA key length of 512 may be too small or may be excessively safe.

(6) *Security Management Information Base Agent (SMIBA).* The SMIBA is responsible for interfacing the SMIB to the other agents within CISS. It is the only agent with direct access to the SMIB.

(7) *Association Agent (AA).* The AA is responsible for the associations between agents within the local security domain. It must make sure that associations are initiated with the correct security and that security is maintained throughout the association.

(8) *Inter Domain Communication Agent (IDCA).* The IDCA is responsible for communication with entities external to the local security domain. Therefore, it must negotiate the required parameters for a secure connection with an external entity. Such negotiations may include the type of cryptographic algorithm to be used, the hashing algorithm, size of key. The IDCA interacts with the AA and the OPENA for the provision of secure inter domain communications.

(9) *Monitoring Agent (MA).* The monitoring agent monitors the activity of CISS, primarily the actions of the SSA, and logs the events within the SMIB. The historical events that are stored within the SMIB are only accessible by the security administrator for the preparation of audit logs.

(10) *Recovery Agent (RA).* The recovery agent is responsible for the recovery of CISS when faults occur either at component level or procedural level, caused either maliciously or accidentally.

3. Local security

The objective of the 3-Level (3-L) architecture is to provide management of security services within the boundaries of a security domain in a distributed environment by selective deployment of the CISS agents. Some of the security services that must be co-ordinated over the entire domain are described below.

- *Logging and auditing security relevant events.* Audit information is analysed here for potential privilege abuse and intrusion detection.
- *Interdomain communications.* When a user wishes to perform some function manipulating an entity within a different security domain, the management structure of each domain must negotiate what services are required for secure communication over the public unsecured network.
- *Generation, storage and distribution of cryptographic keys.* To maintain confidentiality within the domain it is necessary to generate keys to a set policy and then use secure protocols for their storage and distribution.
- *Domain notary service.* The SMC must act as a Trusted Third Party (TTP) (CCITT, 1989) for the users hosted upon it. This means that it must provide the functions, registration, notarisation, certification services and public key verification along with public key distribution for all users.
- *Trusted entry point.* Firewalls (Avolio, 1994), are used for access control to local area networks through the Internet. This method of access control should be implemented in CISS via a single point of entry and exit to the security domain, allowing the implementation for efficient security logging and access control.
- *Domain Communications.* If a user wishes to perform a remote operation upon a machine that is within the same security domain, the security management structure must make sure that all security requirements are met according to the policy laid out within the domain.
- *Access Control.* Through the use of the security domain's SMIB the access rights and privileges of individuals may be determined so that unauthorised manipulation of domain entities can be secured against.
- *Authentication.* Every person that wishes to use CISS functionality must first be authenticated through information stored within the security domain's SMIB.

The 3-L architecture is comprised of a domain management centre (DMC), multiple local security servers (LSSs) and a local security unit (LSU) upon each user terminal.

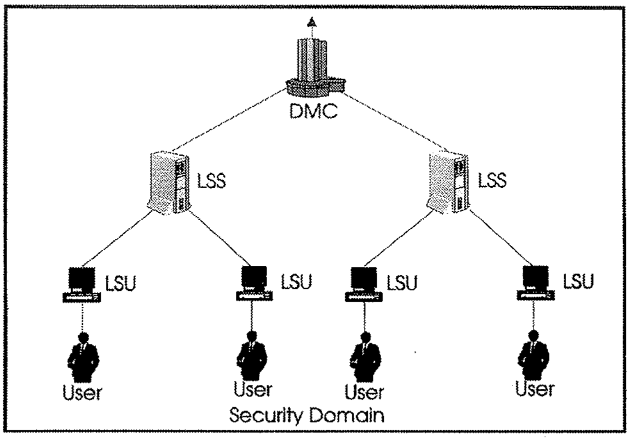


Figure 2: *Three level architecture.*

Multiple LSUs would be hosted upon a single LSS, the number of LSSs within an organisation would be dependent upon its size and its internal policy for computer asset organisation. The three levels interact within the boundaries of the security domain to meet the requirements of the organisation's security policy for activities within the security domain and external to it. The structure of the interaction is shown in Figure 2.

Each higher level within the 3-L architecture is aware of the capabilities of the immediate lower level units that are connected to it, information concerning the hosted security modules would be kept within the SMIBs of the LSSs and DMCs. This knowledge allows the upper levels to determine whether services requested by lower level units meet with the required security laid down by the domain security policy and therefore whether the operation can go ahead. If the operation requires more security than is available, the operation is prevented by the higher level entity and the reason is logged with the requesting unit.

Each of the level entities that make up the three level architecture are built from the agents that are available within CISS. The CISS agent list for each level is given in the table below.

Some of the co-ordination duties are common amongst the three levels and some are specific to a certain level, the table below sums up the division of labour for the 3-L architecture.

Table 4: *Agents used within 3-L.*

	DMC	*LSS*	*LSU*
UA	O	O	X
SAA	X	X	O
SSA	X	X	X
SMA	X	X	X
SMIBA	X	X	O
OPENA	X	X	X
AA	X	X	O
IDCA	X	O	O
MA	X	X/O	X/O
RA	X	X/O	O
ADDITIONAL	X	X	X

X-Present, O-Absent

The 3-L architecture relies heavily upon the use of public-key (Nechvatal, 1991) and secret-key cryptosystems (Schneier, 1994); the public-key cryptographic techniques are used for the authentication, integrity and confidentiality requirements; while the secret-key cryptosystems are used to accomplish very fast bulk encryption solely for the purpose of confidentiality. Every user, LSU, LSS and DMC has their own public cryptographic key pair used for authentication and certified by an LSS, DMC or external TTP. The use of the LSU, LSS and DMC key pairs are for security service management only, while the user's public-key pair is used to attain personal liability, accountability, etc.

3. 1. Domain management centre

The domain management centre is the highest authority within the three level architecture and is comprised of most agents within the basic CISS model, see Table 4. It acts as the single entry/exit point to entities on opposite sides of a security boundary. The DMC is the only level entity within a security domain that has the direct use of the IDCA. Within the DMC's SMIB each LSS has its own

Table 3: *Sample service structure.*

Service	*En*	*DS*	*AC1*	*DI*	*AE*	*TP*	*RC*	*Nt*
Peer Entity Authentication	X	X	O	O	X	O	O	O
Data origin Authentication	X	X	O	O	O	O	O	O
Access Control Service	O	O	X	O	O	O	O	O
Data Confidentiality	X	O	O	O	O	O	O	O
Non-repudiation Origin	O	X	O	X	O	O	O	X
Non-repudiation delivery	O	X	O	X	O	O	O	X

X indicates a mechanism is present and O indicates its absence.

Table 5: *3-L division of labour.*

Security Service	*DMC*	*LSS*	*LSU*
Logging and Auditing Security Relevant Events	X	X	X
Generation, Storage and Distribution of Cryptographic Keys	X	X	X
Domain Notary Service	O	X	O
Inter-Domain Communications	X	O	O
Trusted Entry Point	X	O	O
Domain Communications	O	X	O
Access Control	X	X	O
Authentication	O	X	X

X-Provided, O-Not Provided

security profile in which its inter domain communication properties are listed, i.e. the mechanisms and services to be used during contact with entities outside the local security domain.

For example, a business wishes to place its accounting machines upon a LAN so that its accountants may access them from remote sites across the domain. However, the administration is worried that the placement of the machines upon the LAN will put them at risk from attackers outside the organisation's security domain. Therefore, a rule is entered into the DMC SMIB that no external associations are allowed with the accounting machines. As the DMC is a required route for external connections to the accounting machines greater access control is attained.

The DMC is responsible for the implementation of the security policy laid out for a domain. To accomplish this the DMC is capable of modifying the SAA or rather the SMIB entries that determine the security mechanism used by various protocols within the 3-L architecture by each of the levels, such as digital signature systems or cryptosystems used for confidentiality.

The co-ordination and synchronisation of management keys within the security domain is the primary responsibility of the DMC. The DMC ensures synchronisation by regular distribution of LSS public-key certificates to all LSSs. The certificates can be generated and distributed by the DMC during off-peak hours when system usage is low. The timing of the generation of public-keys and their associated certificates will be set within the domain's security policy; for example it might be decided that generation and distribution of certificates will be performed weekly on Sunday nights, which in all likelihood is the quietest night in most organisations.

3. 2. *Local security server*

The local security server provides the framework for secure communications across the LAN and has a part in all of the LSU's activities. The LSS can be thought of as a file server and any references to files will be examined by the LSS. It enforces the protocols to be used as indicated by the DMC and subsequently entered into the LSS's SMIB. The DMC provides the LSS with notary services and trusted key certificates. It is envisaged that physical access to the DMC machine will be restricted and the attainment of system privileges be restricted to minimal personnel. The DMC should be solely controlled by the security section of the organisation.

The LSS provides secure services such as secure file transfer, notary services, etc., to the LSUs that it hosts (Figure 3).

For secure communication outside of the security domain the LSU contacts with LSS which then contacts the DMC on the LSU's behalf.

3. 3. *Local security unit*

The local security unit is situated at the user terminal and is concerned primarily with the confidentiality of data between it and the LSS. If dumb terminals are being used on a network then due to their lack of processing power and therefore their inability to maintain confidential comunications between themselves and their host, they provide a small security risk. In the case of dumb terminals it must be assumed that the physical links between the terminal and the host are secured, either through cryptodevices (Barnes,

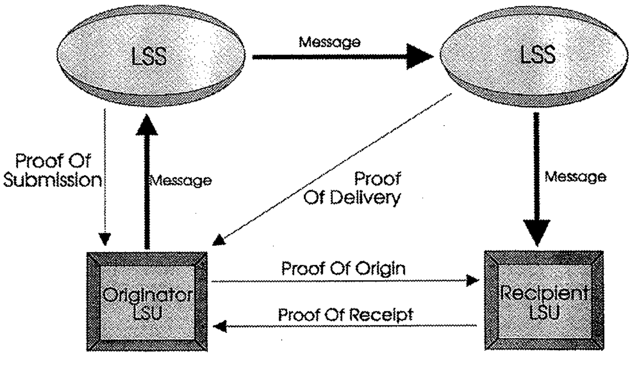


Figure 3: *Non-repudiation routes.*

1993) at either end of the connection, or physical impediments, for example concrete encasement of the line.

In the event of dumb terminals being used within the 3-L architecture the LSU will reside on the host and the security methods for confidentiality mentioned above must be relied upon for secrecy. However, with the downard spiral of the cost of processing power and the movement towards distributed processing, it can be assumed that most recently procured user terminals on LANs will not be dumb terminals, with the most common approach being the use of PCs. How much processing power these terminals have varies greatly and is dependent upon such things as the computer purchase policy of the company, or the position of the user within the company. Therefore, the architecture of a security system must be flexible enough to cope with different levels in computing technology.

The local security unit provides the processing power for most of the work in the provision of security under CISS. Within the 3-L architecture there are many LSUs within a domain. This division of processing power is a more efficient way of meeting the needs of users than through the use of a large central processing element. Therefor, it is important that the LSU works as quickly and as efficiently as possible.

The bulk of processing performed by the LSU will be by the cryptosystems within the security protocols. Public-key and secret-key systems can be very processor intensive and should therefore be provided by dedicated hardware wherever possible. Fast RSA hardware and cryptographic boards have been designed by the Network Research Group (Onions, 1995), (Morrissey, 1995) for this very purpose.

4. Expansion of the monitoring agent

Almost all security placed on computer systems are preventative, be they smart cards, passwords or biometric identification systems (Sherman, 1992). They are barriers placed in the way of an attacker attempting to gain unauthorised access to a computer system. As useful as these techniques are for access control, once an attacker has by-passed them, they become redundant. As a consequence, conventional computer security systems have repeatedly failed to deal with three major security problems (Madsen, 1992), (Clough and Mungo, 1992), (Fawcett, 1995):

(1) illegal use of a system by an unauthorised user once access controls have been compromised;

(2) abuse of privileges by an authorised user. User abuse is potentially the most dangerous security breach as regular users will have a good idea of where to direct attacks for the most damage, or where the most valuable information is kept upon a system;

(3) anomalous behaviour of system resources, through the presence of bugs or malicious software, such as logic bombs, viruses and Trojan horses.

The traditional method for detecting cases 1 and 2 above is through the use of audit logs. Many existing operating systems such as VMS (Sandler, 1989) offer limited auditing facilities that can be used to detect masqueraders. However, the amount of auditing information that can be generated is vast. Therefore, the information cannot be processed by a single security administrator, and even if it was, discovery of a masquerader/intruder would have been made too late to prevent any damage from being caused. In case 3, computer viruses are the most well known and have become much more complex as their writers have employed more sophisticated techniques in the creation of malicious software. One of the main tools a security administrator has in the combat of computer viruses are virus scanners. Virus scanners operate by maintaining a list of known machine code patterns that appear within viruses and searching a system for matching patterns, if any of the patterns are found a suspected virus is reported.

Perhaps the best method of dealing with the security problems described above is through the continual surveillance of system activity (Lunt, 1990), (Mukherjee and Heberlein, 1994). If the following two premises are true then the development of a system to deal with the security problems discussed is possible.

(1) It is possible to learn the normal activity of a system, its resources, and the users on it, so that its behaviour can be predicted.

(2) A masquerader, privilege abuser, or virus exhibits anomalous activity that can be detected as not being part of the normal system behaviour and the correct countermeasures can be implemented to prevent or limit damage.

The reasoning behind the two premises are as follows.

- The behaviour of a system cannot be unpredictable for any length of time as this would provide no platform for any productive work to be done on it. User behaviour can be categorised due to two prevalent factors. First, humans form habits that are peculiar to them, for instance there are multiple ways to open a file and once a user uses one way they rarely change even though it may not be the most efficient way. The second is because of job specialisation, for example a secretary on the system will use different system resources than a programmer.
- Anomalous behaviour is any behaviour that falls outside of the norm for a specific entity. In the case of a maquerader the control behaviour pattern would be that of the user whose account is being used. It can be assumed that a masquerader will not perform the usual tasks that the user would be expected to perform; the masquerader is not there to perform as the legal user but instead there for exploration, or malicious damage. Programs should behave in a predictable manner. If they do not, for instance when infected by

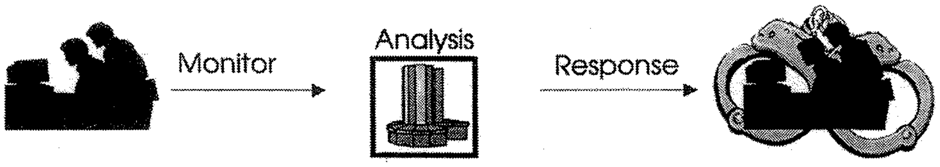


Figure 4: *Stages of intrusion detection.*

a stealth virus, they may exhibit file modifications or writes to memory that are uncharacteristic. A potential privilege abuser may show indicative signs by excessive browsing, the attainment of super user status, or the ability to assume another user's id–all of which show the potential for abuse.

A monitoring scheme has three stages, shown in Figure 4.

- *Monitor.* System variables that are capable of being used to indicate anomalous behaviour must be monitored and stored for historical analysis at some later point.
- *Analysis.* Using the measurements of the system variables, the monitoring scheme must be capable of recognising anomalous behaviour upon the computer system.
- *Response.* Depending on the conclusions reached by the analysis stage, a response must be formed and implemented, limiting the damage to data upon the computer system and the protection of services to users.

The extended monitoring agent allows the generation of a single security log that can be comprised of generic CISS events and specific events for the computing platform upon which CISS exists, i.e. the Operating System.

Through the use of security log entries a history of user activity can be built up. From this information a *profile* can be generated that describes the user/entity's normal activity on the system (see Figure 5). This profile can then be used in the detection of anomalous behaviour during any future user/entity activity.

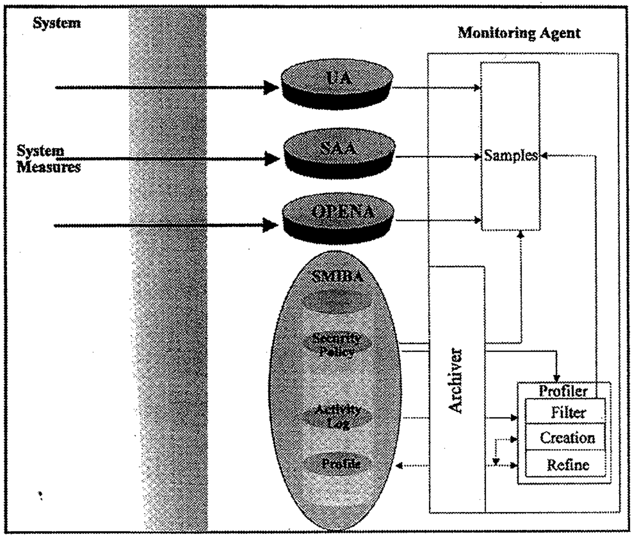


Figure 5: *Profiler.*

Profiling can be divided into three sections.

(1) *Data filtering*: through the correct data selection, the most indicative measures are used while the rest are discarded to reduce processing. This is also a function of the analysis section of the monitoring agent. All data that is thought to be relevant is collected and stored.

(2) *Profile creation*: from the filtered measures a control profile is created that indicates the boundaries of normal behaviour.

(3) *Profile refinement*: the profile is continuously refined so that it is up to date.

The basic CISS security log will be made up of two entries, the basic log entry and the activity entry. The basic log entry is made up of fields that identify the user/entity that caused an event, the time at which the event took place, the system resource acted upon, etc. An activity entry is a single user/entity action such as use of mail tools, text editors etc.

The two security log entries are used by the monitoring system to create profiles — activity profiles and behavioural profiles (Figure 6). Activity profiles describe a user/entity's normal activity entries. Behavioural profiles are created through combinations of activity profiles, for example 'session start-up' describes the first activities a user performs at the beginning of a session: login, reading mail, reading subscribed newsgroups.

Because profiles are meant to be representative of the user's normal activity it is important to keep them up to date. This can be done through the acquisition of sample activity during normal sessions. Automatic updating in this form can be accomplished daily or weekly, depending on the activity of the user. Updating a profile would only take place at the termination of a successful session. This

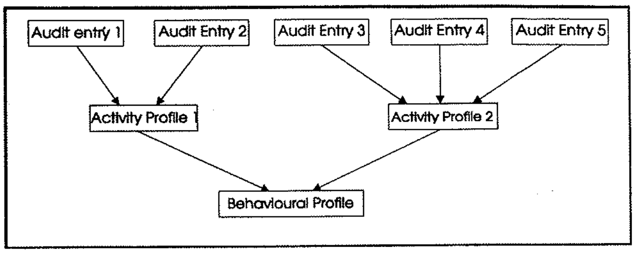


Figure 6: *Profile monitoring.*

reduces the chance of contamination of a profile by possible masquerader activity. The logical section within the monitoring agent that is responsible for profile refinement is the profile refiner.

The surreptitious updating of profiles can lead to the problem of slow change. If privilege abusers know that profiling is being used with automatic updates they may attempt to slowly modify their profiles. Thus they may not exhibit anomalous behaviour even when abusing the system. For example, a privilege abuser can gradually introduce file browsing into his normal list of activities and then slowly increase his browsing activities.

A security administrator may force an update of a profile by requesting it through the SAA. This would be required at initial set-up of the system, with the addition of a new user, or with the addition of new software.

Software profiles will be limited to activities. By attributing a piece of software with an activity profile it should be possible to detect any anomalous writes or reads that could be indications of a Trojan horse or virus. Such activity, when detected by the monitoring scheme's analyser, may demand the initiation of a virus scanner specifically targeting the suspect piece of software. If no virus is found the responder may tag the piece of software for the security administrator's attention. The security administrator can then use the activity audit entries to examine exactly what the piece of software is doing.

Anomalous activity detection of software using profiling will only work if the knowledge base holds profiles prior to infection. Any infected software introduced into the system for the first time will not be detected as exhibiting anomalous behaviour when the stealth virus delivers its payload. Since most organisations use a limited number of software packages it is possible that standard applications can be profiled and their profile introduced into the knowledge base prior to installation of the software upon the system. These standard profiles can be generated either by the application suppliers or the CISS suppliers.

Computer terminals can be given behaviour profiles so that their actions upon a network can be monitored for anomalous activity. Activities that a terminal may be audited for, and for which a behaviour profile may be generated, include network connections, remote logins, traffic intensity, etc. This sort of profiling can help in the prevention of machines being used for activities such as hacking.

Profiles take time to be generated, so the most vulnerable time for a system is when a new entity is placed on it (user/software). Therefore, until enough information has been gathered for an initial profile, standard security mechanisms must suffice.

Within the monitoring agent there should be real-time analysis and off-line analysis. The real time analysis will be made up of three analysers (Figure 7).

(1) The task of the expert system will be to detect anomalous behaviour by applying the experience that security experts have imbued into it through the rules that they have generated for its inference engine. The expert system will use the essential rules as dictated by the same experts monitoring in real-time.

(2) The task of the statistical analyser is to detect anomalous behaviour of entities through activities outside of their norm as dictated by their profiles. This is more specific than that of the expert system, although the real-time expert system will use the results of statistical analyser in its own analysis.

(3) The neural network/statistical classifier should be used in very specific cases where the samples that are needed for its training/model creation can be collected and the problem is very specific. Previous work on the application of neural networks within security can be found in (Furnell, 1996).

As well as the anomaly detection through activity profiles there should be other real-time supervisory mechanisms requiring analysis, such as keystroke timings, network traffic analysis, etc. Ther is also an off-line security log analyser, that is, an expert system with a full rule base that can conduct thorough analysis of the security log. Anomalies detected through these mechanisms are reported to the evaluation block of Figure 7.

The evaluation block can be either a simple sum threshold scheme with logic for further refinement of profiling and sampling, or it can be an expert system programmed to perform a decision on the current security of CISS through the data provided by the monitoring agents real-time analysers. The monitoring agent will have an awareness rating, the greater the monitoring agents rating the more sensitive the analyser becomes so that slow, continued, suspicious activity will eventually cause a response from the system.

5. Conclusion

The division of the CISS agents into separate functional units (as described within the 3-L architecture) allows CISS to be deployed on computing platforms of different capacity, thereby reducing the impact in system performance by tailoring the security activities to the capabilities of machines within a heterogeneous computing environment. When each of the level units within the 3-L architecture are equipped with hardware cryptographic capability, it is envisaged that the security system will provide quick response and low system impact. A prototype is in development at the Network Research group within the University of Plymouth.

The extended monitoring agent can provide increased security through monitoring system activity. It is important that new techniques be found in the detection of anomalous behaviour upon a system. The increasing frequency of reports in the world news of computer security breaches

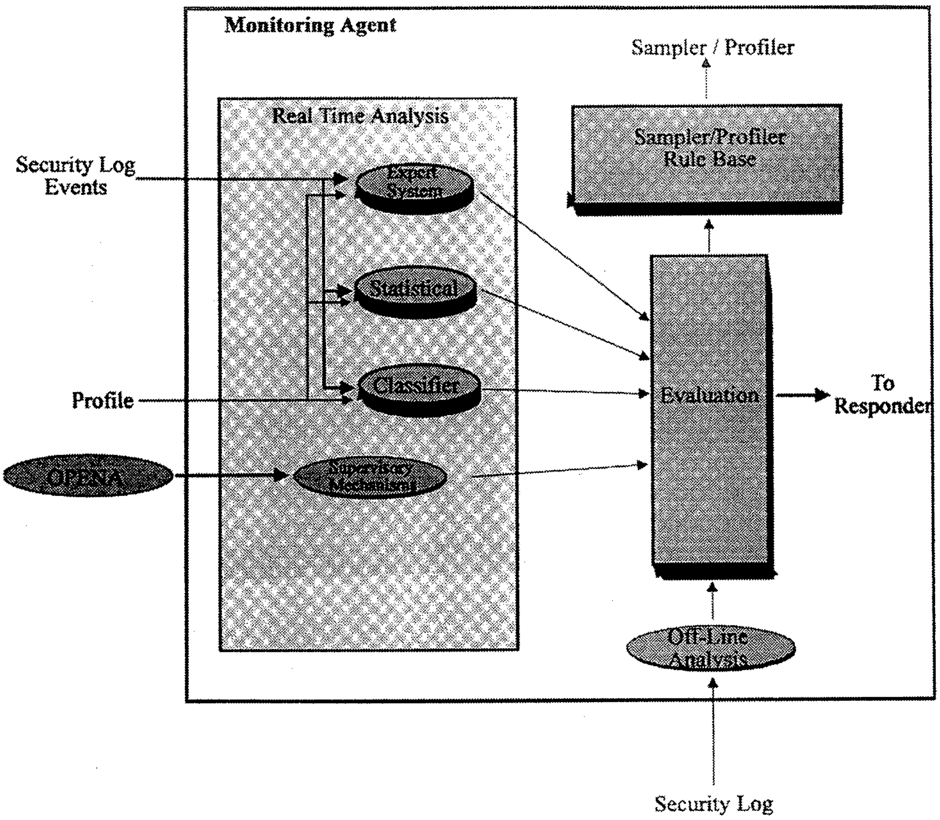


Figure 7: *Analysis within CISS.*

only lends greater urgency to the development of innovative and effective ways of securing computer resources.

References

Audit Commission (1990) *Survey of Computer Fraud & Abuse.*

Avolio, F.M. (1994) Building Internet firewalls, *Business Communications Review*, **24**, 15–20.

Barnes, D. (1993) Designing more secure LANs, *Telecommunications*, **27**, pp. 64 & 98.

CCITT, *1989 Data Communication Networks Directory*, X500–X521.

Clough, B. and P. Mungo (1992) *Approaching Zero: Data Crime and the Computer Underworld*, Faber and Faber, London.

Fawcett, N. (1995) Microsoft gives virus to 200 top UK developers, *Computer Weekly*, 23rd February, 1995.

Furnell, S., J. Morrissey, P. Sanders, and C.T. Stockel, (1996) Applications of keystroke analysis for improved login security and continuous user authentication, *Proc. IFIP Sec.*, May 1996.

Guardian (1993) Hacked E.C. Computer, Guardian, 24th February, 1993.

International Standards Organisation (1986) *Use of Encipherment Techniques in Communication Architecture*, ISO/TC 97/SC 20/WG 3, 1986.

International Standards Organisation (1988) *OSI RM Part 2: Security Architecture*, ISO DIS 7498-2, 1988.

Lunt, T.F. (1990) IDES: an intelligent system for detecting intruders, *Proc. Symposium, Computer Security: Threat and Counter Measures*, Rome, Italy.

Madsen, J.B. (1992) The greatest cracker-case in Denmark: the detector, tracing and arresting of two international crackers, *Proc USENIX*, pp. 17–40.

Morrissey, J.P. (1995) *A hardware implementation of the CISS concept*, University of Plymouth, Ph.D. Thesis.

Muftic, S., A. Patel, P. Sanders, R. Colon, J. Heijinsdijk and U. Pulkkinen (1993) *Security Architecture For Open Distributed Systems*, Wiley.

Mukherjee, B. and L.T. Heberlein (1994), Network intrusion detection, *IEEE Networks*, **8**(3), pp. 26–45.

Nechvatal, J. (1991) *Contemporary Cryptography: Public Key Cryptography*, IEEE Press.

Onions, P. (1995) *A High Speed Integrated Circuit for Applications to RSA Cryptography*, University of Plymouth, 1995.

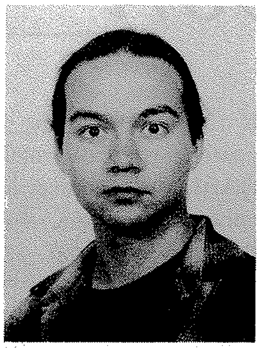
Sandler, C. (1989) *User's Guide to the Vax/VMS Operating System*, Scott, Foresman and Company.

Schneier, B. (1994) *Applied Cryptography Protocols, Algorithms and Source Code in C*, Wiley.

Sherman, R.L. (1992) Biometric futures, *Computers and Security*, **11**(2), pp. 128–133.

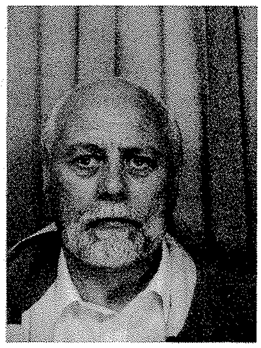
Sunday Times (1993), Freefone lines set off spate of data rape, *Sunday Times*, 14th November, 1993.

The authors



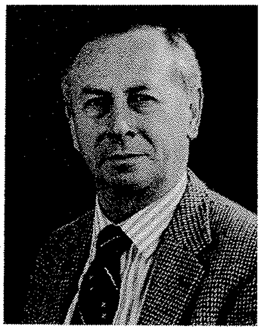
Joseph Morrissey

Dr Joseph Morrissey has completed a PhD at the University of Plymouth. He graduated from the University of Sussex with a degree in Electronic Engineering and then moved into computer security within the Network Research Group. Through research for his PhD he has investigated security within distributed environments, real-time authentication techniques and cryptographic hardware. He is now working for Aldiscon in the area of mobile telecommunications.



Peter Sanders

Peter Sanders is the Director of the Network Research Group at the University of Plymouth. He is currently involved with research into security systems, network design and Integrated Service Engineering in association with industry and the European Union, where he is currently engaged on ACTS and TAP projects. He is the author of a number of technical papers, reports and books in these areas.



Colin Stockel

Dr Colin Stockel is Reader in Computer Science at the University of Plymouth. He graduated from the University of London with a BSc in Physics and a PhD in High Energy Particle Physics. Since 1967 he has taught and carried out research in Computer Science, becoming a Fellow of the British Computer Society and a Chartered Engineer. His current research interests lie in the area of computer security and networking, simulation and mathematical modelling, in which he is the author of numerous publications.