# INFOSURE: AN INFORMATION SECURITY MANAGEMENT SYSTEM

by

DIEDERIK PETRUS VENTER

**DISSERTATION**

submitted in the fulfilment of the requirements for the degree

**MASTER OF SCIENCE**

in

**INFORMATICS**

in the

**FACULTY OF SCIENCE**

at the

**RAND AFRIKAANS UNIVERSITY**

**SUPERVISOR: PROF S.H. VON SOLMS**

**MAY 2003**

## Abstract

Information constitutes one of an organisation's most valuable assets. It provides the modern organisation with a competitive edge and in some cases, is a requirement merely to survive. An organisation has to protect its information but due to the distributed, networked environment of today, faces a difficult challenge; it has to implement a system of information security management.

Software applications can provide significant assistance in managing information security. They can be used to provide for centralised feedback of information security related activities as well as for centralised configuration activities. Such an application can be used in enforcing compliance to the organisation's information security policy document.

Currently there are a number of software products that provide this function in varying measures. In this research the major players in this space were examined to identify the features commonly found in these systems, and where they were lacking in terms of affordability, flexibility and scalability.

A framework for an information security management application was defined based on these features and requirements and incorporating the idea of being affordable, but still flexible and extendable. This shifted the focus from attempting to provide a comprehensive list of interfaces and measurements into general information security related activities, to focusing on providing a generic tool that could be customised to handle any information fed back to it. The measurements could then be custom-developed as per the needs of the organisation. This formed the basis on which the prototype information security management application (InfoSure) was developed.

## Opsomming

Inligting vorm een van ‘n organisasie se belangrikste bates. Dit verskaf die moderne organisasie met ‘n kompeterende voordeel wat soms bloot nodig is vir oorlewing. Die verspreide netwerk omgewings van vandag maak dit baie moeilik vir ‘n organisasie om sy inligting te beskerm. ‘n Stelsel van inligting sekerheids bestuur moet geïmplementeer word.

Sagteware produkte kan genoegsame steun verskaf in die bestuur van inligtings sekerheid. Dit kan gebruik word vir die sentrale terugvoer van informasie oor inligting sekerheids aktiwiteite, asook vir sentrale konfigurasie. Só ‘n produk kan gebruik word om te bepaal of ‘n organisasie voldoen aan die veresites wat gestel word in hulle inligting sekerheid beleids dokument.

Daar is huidiglik ‘n aantal produkte op die mark wat hierdie funksies tot ‘n mindere of meerdere mate verskaf. In hierdie dokument is die belangrikste produkte in hierdie kategorie bestudeer en die eienskappe wat algemeen in hulle voorkom geïdentifiseer. Hierdie pakkette se tekortkominge in terme van bekostigbaarheid, buigsaamheid en uitbreibaarheid is ook uitgelig.

‘n Raamwerk vir ‘n prototype inligting sekerheids bestuurstelsel produk is gedefinieer. Dié raamwerk is gebaseer op die geïdentifiseerde eienskappe, asook die vereitses om bekostigbaar, buigsaam en uitbreibaar te wees. Die fokus het dus verskuif van ‘n produk wat slegs ‘n omvattende stel metings en koppelvlakke verskaf, na ‘n generiese produk wat verpersoonlik kan word om enige teruggevoerde inligting te kan hanteer. Die metings kan dan self ontwikkel word deur die organisasie. Die prototipe (InfoSure) is teen hierdie raamwerk ontwikkel.

# Table of Contents

# Chapter 1

# 1 Research Overview and Objectives

## 1.1 Problem Statement

In today's information security management space there are several applications aimed at assisting the information security officer (ISO) with the management of information security. Unfortunately these packages lack in a number of areas.

Due to their extensive nature the packages are generally expensive to acquire and implement and often rely on other information security packages as information sources. These additional packages are themselves costly, thus compounding the expense factor.

Most of the current packages offer additional modules that plug into their management systems, but these are also expensive and are generally developed for existing systems. In other words, an organisation is not guaranteed the capability to extract information from any future system it implements, unless the software vendor has specifically developed interfacing modules for these systems.

An application should have the flexibility to provide the ISO with the ability to monitor compliance on both single-entity and multiple entity levels. For example (using users as an entity) compliance could be related to a level affecting all users, or to a finer grained level that only applies to a sub set of users. An organisation might also require more complex evaluations involving different types of entities, e.g. users and software. The system should be able to cross evaluate the results of more than one set of activity information fed into it. Current packages allow for evaluations of individual activity feedback, but do not allow

creation of complex rules based on the feedback from different activities that can be related in interpretation.

Although many smaller organisations need to implement information security management, they either lack the necessary funds or cannot justify the excessive expenditure of acquiring any of the current packages. What they require is an inexpensive application that can easily be customised and configured to their specific needs, but also extendable to cater for growing, changing needs in the future.

In summary, the currently available applications are too expensive for smaller organisations to implement and lack the flexibility as an entry-level system that can be grown alongside the organisation.

## 1.2 Objective

The objective of this research is to:

- Examine and evaluate the current enterprise security management packages available on the market.
- Identify the general baseline requirements for an information security management system, based on the features of the current packages on the market. In addition identify additional requirements for creation of entry level, flexible systems with potential for expansion, as well as complex sets of rules.
- Design and develop a prototype application based on these key requirements that will enable the ISO to effectively manage information security, and afford an organisation the flexibility to cost-effectively expand the system based on its changing needs.

## 1.3 Research path

During the course of this research a number of products currently operating in the information security management arena were identified. These products offer the ISO varying degrees of assistance. They will be examined more closely, and the common requirements of the prototype application will be defined based on this evaluation. Additional requirements will be added in an attempt to fill identified gaps where current systems are lacking. The prototype application shall be developed as a proof of concept.

Before we start developing such an application, we will step back and examine the need for information security and its management in an organisation.

Next, we shall examine the concept of an organisational security policy. We will examine what it provides in defining what information requires security in the organisation, as well as the direction it provides regarding information security management. We'll also examine how the security policy can be incorporated into the information security management system.

Once we have a clear understanding of the concept of information security, information security management and the organisational security policy, we can define a framework of reference on which to develop our information security management system prototype. This will be based on our understanding of what the system should provide and what current systems have to offer.

## 1.4 Structure of the Document

In Chapter 2, why we need to protect information, is discussed. It also explains what information security is, why this should be managed and the importance of a security policy document.

In Chapter 3 a number of the currently available enterprise security management systems are examined and a comparative matrix constructed using criteria based on functionality identified in these products.

Chapter 4 provides the framework and key requirements of the prototype information security management system to be developed. The prototype is dubbed 'InfoSure'.

Chapter 5 discusses the architecture of the prototype system, and the approach taken to fulfil the key requirements identified in Chapter 4.

Chapter 6 will expand on some future additions and improvements that could be made to the prototype system.

Chapter 7 presents a number of scenarios from hypothetical security policy documents, as well as solutions for each scenario developed in the prototype system.

Chapter 8 contains a walkthrough of the developed prototype system.

Finally, in Chapter 9 conclusions are drawn on the validity of the prototype system and the study as a whole.

# 2 Information Security and Information Security Management

## 2.1 Introduction

Due to the nature of the modern networked environment, and the importance of internal and external information sharing by organisations in order to remain competitive, or indeed just to survive, a need for information security in the organisation has arisen. For the organisation to maintain the security of their information, they need to define a set of guidelines - an information security policy that can be enforced by the organisation. This policy should be a living document, based on a cycle of monitoring current activity and events, comparing these to the policy guidelines, and initiating actions to enforce and monitor compliancy to the guidelines [ISO 17799].

## 2.2 The Need to Protect Information

Why do we need to protect information? Information constitutes one of an organisation's most valuable assets. The advent of the computer made the management of information substantially easier, and its storage and retrieval much quicker. There used to be good control over the access to the information, but with the move towards networked environments, this control has become a lot more difficult.

In particular, there are two developments over the last decade that has emphasized the importance of information security management. The first is the way in which organisations have had to adjust their business model. In the past many companies confined themselves to specific geographical areas in which they did business. These days, a lot of companies are forced to become location-independent for strategic advantages. Many organisations used to process their information centrally, which allowed for a larger degree of control over the security

of the information. Due to the organisational and transactional spread over larger geographical regions, information, which still fulfils a strategic role, followed suit, becoming decentralised across broad geographical areas.

Secondly, the advent of the Internet and its subsequent growth led to a critical need for organisations to be connected to it, not just to gain strategic advantage, but for mere survival. This has resulted in increased vulnerability of company information as it is forcibly exposed to networks outside of the organisation's control [OLI1999], [SCH2001].

The Internet has become a double-edged sword providing many opportunities for organisations, but bringing with it a greatly increased information security risk.

All these factors mean that any company wishing to survive in today's networked environment will be forced to expose its information to others. There is no way around this, and all an organisation can do is attempt to secure its information from unauthorized parties. They need to provide some form of information security.

## 2.3 Information Security

Security is about protecting valuable assets against loss, disclosure or damage. This concept also applies to information within organisations. It must be protected against threats that could lead to its loss, alteration or disclosure. Generally, information security is seen as the preservation of [ISO17799], [PFL1989]:

- Confidentiality: ensuring that information is accessible only to those with authorization;
- Integrity: safeguarding the accuracy and completeness of information and processing methods.

- Availability: ensuring that authorized users have sufficient access to information and associated assets when required.

This should be achieved by a technical means, but also by means of non-technological safeguards [OLI1999], [PIE2001].
Information security is achieved by the implementation of a suitable set of controls to ensure the specific security objectives of the organisation are met. These controls could comprise policies, practices, procedures, organisational structures, and software functions [ISO17799]:

There are eight core principles that apply to the management of information security [OLI1999]:

- Accountability – Information must have an owner who accepts responsibility for it and accountability for its security or breach thereof. This responsibility should not be passed on to information managers when it is clearly the responsibility of business managers.
- Awareness – Everyone in an organisation needs to be aware of the risks associated with insecure information and must know the company's security processes, as well as those relating specifically to their own areas of responsibility.
- Multidisciplinary – Information security is more than just a technology. It also covers issues of an administrative, organisational, operational and legal nature. Although there should be technical standards, these need to be reinforced by way of codes of practice, educational awareness and training.
- Proportionality – The cost of the control needs to match the level and probability of risk, as well as the potential loss of availability, integrity or confidentiality.
- Integration – Security should improve our ability to process information for business benefit, and not impede it.

- Reassessment – Security measures should be subject to regular review as technology, the types of attack and business processes and patterns change.
- Timeliness – As modern systems are often interconnected and operate in real time, a breach of security can result in an instant loss. Thus, systems need to be monitored in real-time and have effective responses planned.
- Societal factors – Organisations have an ethical duty to their customers and suppliers to protect these parties' information.

Since an organisation's information is regarded as an asset, which could potentially lead to substantial competitive advantage, it is becoming increasingly important that all facets of the creation, usage and storage of this information are protected [PIE2001], [DHI2001]

Before an organisation can implement security measures to protect its information, it needs to identify exactly what information to protect and what a 'sufficient' level of security would be. There are three main sources through which an organisation can achieve this [ISO17799]:

- Threats to assets can be identified via risk assessments. The likelihood of occurrence and vulnerability to threats is evaluated, and the potential impact estimated.
- Legal, statutory, regulatory and contractual requirements that must be met by an organisation, its trading partners, contractors and service providers.
- The specific principles, objectives and requirements for information processing in an organisation have to be developed to support operations.

Due to the changing nature of organisations and the evolving scope of information processing, managing information has shifted from maintaining confidentiality, integrity and availability to also establishing

responsibility, personal integrity, trustworthiness and ethicality [DHI2000].

There also seems to be a 'policy vacuum' in dealing with information security related issues, and the management of these issues. An example of this is the case of Randal Schwartz [DHI2000a] In this case there were difficulties in establishing whether the illicit use of computers by him had amounted to an occurrence of computer crime.

Taking all of this into account, an organisation has to identify its own security requirements and, based on this, select the necessary controls to ensure that risks are reduced to an acceptable level. Each information system or environment could thus have its own unique set of controls put in place that would be sufficient for that environment specifically.

A number of controls either based on legislative requirements or considered good practice can be considered to provide a good starting point for implementing information security [ISO17799]:

- Data protection and privacy of personal information
- Protection of organisational records
- Intellectual property rights
- Information security policy document
- Allocation of information security responsibilities
- Information security education and training
- Security Incident Reporting
- Business continuity management

According to [ISO17799], experience has shown a number of factors critical to the successful implementation of information security in an organisation:

- Security policy, objectives and activities that reflect business objectives
- Approaching security implementation in a way that is consistent with the organisational culture
- Visible support and commitment from management
- Good understanding of security requirements, risk assessment and risk management
- Effective marketing of security to all managers and employees
- Distribution of guidance on information security policy and standards to all employees and contractors
- Providing appropriate training and education
- A comprehensive and balanced system of measurement, which is used to evaluate performance in information security management and feedback suggestions for improvement

In summary, we can see that there are two main parts to effective information security. It is necessary that there is an explicit definition by the organisation of expected behaviour, activities and events regarding information security maintenance (policy), as well as a process to measure the organisations performance against this defined behaviour (management). How would an organisation go about defining its preferred behaviour in such a way as to be enforceable? This introduces the concept of an information security policy document.

## 2.4 Why an Information Security Policy Document?

The objective of an information security policy document is to provide direction for management and to support an organisation's information security [ISO17799].

An organisation's investment in the necessary security measures to protect the availability, integrity and confidentiality of its information has a substantial cost attached to it. It will most likely consume considerable time from skilled IT staff and involve expenditure on hardware, software and services as well. This can only be judged by assessing the information asset's value to the organisation and the consequences should it be lost, compromised or services interrupted. The impact on the organisation thus has to be assessed in the context of an information security policy.

Additional advantages of an information security policy are that [ROB2001]:

- A framework is provided in which roles and responsibilities can be defined with respect to data security
- Necessary regulations can be defined and justified
- The organisation can explicitly relay its attitude towards actions that could threaten the security of its information

By issuing and maintaining an information security policy across the organisation, management can set a clear policy direction and demonstrate its support for information security [ISO17799].

Based on an information security policy, an organisation can draw up an information security plan.

Converting this policy into a practical plan requires the following main elements [ROB2001]:

- Assigning value to an organisation's information and technological assets
- Assessing the risks to these assets. (Determining the threats to an asset, as well as the probability of that threat materializing)
- Determining the appropriate level of security to protect assets (based on the aforementioned points)
- Ensuring that the appropriate resources are made available to realise the required security measures
- Providing the necessary training and publicity to support the effective operation of the necessary security measures
- Creating a timetable by which to review the security plan on a regular basis and so to keep up with changing requirements, including personnel and the external environment

There needs to be an ongoing cycle of monitoring of the elements defined in the security policy, using the feedback from the monitoring to check for compliance with the security policy, and lastly, adjustment and maintenance of the security policy based on the compliancy indicators. There needs to be a system of information security management.

## 2.5 Information Security Management System

Information security management can essentially be defined as practices and procedures put in place to help in the preservation of information's confidentiality, availability and integrity. To be effective it needs to be adaptive to change, while remaining consistent with the organisation's strategy. It should deliver benefits such as manageability, assurance and efficiency [PIE 2001].

[ISO17799] defines an information security management system as a systematic approach to managing and protecting sensitive information in an organisation, which could encompass people, processes and IT systems. Essentially an information security management system should allow you to coordinate your security efforts in an effective way.

The typical security lifecycle consists of three parts [CON 1992]:

- *Policy*: (Discovery phase) This is the part of the cycle in which possible threats and risks are identified. Assets in need of protection are identified. A strategy is developed to protect these assets against identified threats. This strategy will dictate the technologies, resources, tactics, and training required for enforcement.

- *Enforcement*: (Action phase) Policy design, data collection, assumptions, the education of users and enforcers, tactics, enforcement and prosecution methodologies are tested in this phase. Execution of the security policy and operational life form part of the security lifecycle's enforcement phase. In this phase, all security assumptions are tested and either survive or dissolve.

- *Assurance*: (Proof phase) The policy, the strategy and their effectiveness are tested. Any failures should be analysed for incorporation into the policy. Information collected through the execution of the enforcement strategy will be used for this purpose. The assessment itself will provide additional information on the compliance of the organisation to the policy, and therefore on its success or failure.

In terms of this research, an information security management system will be used in reference to the policy, enforcement and assurance

parts, but also to an actual software application which will attempt to assist the information security officer in fulfilling the needs of an organisation's information security management requirements. This will be done by means of regular monitoring and interpreting of security policy related events to enforce and gauge compliancy.

But how would the application be able to monitor activities and events, and how would the results of the monitoring be interpreted?

## 2.6 Information Security Monitoring and Measurement

In the context of this research, information security measurements imply the comparison of the performance of an organisation's information security management effort to that proposed in a security policy document or, for instance, a standard such as [ISO17799].

Once these security requirements have been identified, specific actions pertaining to the requirements can be monitored and the results of the monitoring process used to measure the organisations compliance to this.

From a technical point of view, software agents could be used to facilitate the monitoring and relaying of specific information security related activities in the organisation. These agents could effectively run scheduled checks, or be triggered by events occurring in the organisations measured information security environment. Based on the feedback provided by these agents, the organisation can measure its performance against the stipulations of a security policy. The information security management system should serve to empower the information security officer to properly and effectively manage information security.

In addition it should provide the administrator with the capability to set up some form of comparison between the monitored events and the expected behaviour stipulated in the policy document or information security rules of the organisation.

## 2.7 Summary

Information constitutes one of an organisation's most valuable assets and needs to be protected. The advent of the computer has made the management of information much easier, and access to information a lot quicker. It has also lead to greater risk to an organisation's information and thus increased the importance of information security management. In a nutshell, information security management can be viewed as the maintenance of confidentiality, integrity and availability of information.

Firstly, an organisation needs to identify what information it wants to protect and then what a sufficient level of security would be for the information. An explicit definition of this leads to the concept of a security policy document. It sets out a clear policy direction for the organisation and, based on this, an information security plan of action can be drawn up. The typical security lifecycle consists of three phases, namely policy development, policy enforcement and assurance. To assist the information security officer, software agents can be used to monitor and feed back activity related to the information security policy.

Based on the information gathered in this chapter, we are nearly ready to define the framework for our information security management system application. The first step is to evaluate the current packages available in the market to identify the basic requirements of the prototype system.

# 3 Enterprise Security Management Tools Available

## 3.1 Introduction

There are various products currently available in the market that assist to some degree in the achievement of an organisation's security objectives. Typically, the enterprise security management systems either cover the policy creation aspect of the security cycle, or the enforcement and assurance aspects. The major players shall be examined and their features discussed. These provide a good cross section of what currently is on offer. The purpose of this chapter is to examine each of these packages and identify key traits that can be used as criteria for comparing the products with one another, as well as to provide a basis for the key requirements of the prototype system's design.

## 3.2 Current Enterprise Security Management Systems

### 3.2.1 AccessMaster by Evidian

*AccessMaster ensures high security level by federating your existing security solutions, while ensuring at the same time user's convenience with Single Sign-On and security officer's ease of administration with centralized, Ldap-compliant, user and PKI management. In this way, AccessMaster reduces IT security cost of ownership, with rapid return on investment.* [EVI 2002].

Accessmaster by Evidian is a modular suite of software, which allows an administrator to implement an enterprise wide security policy that secures internal systems and applications, while allowing controlled access from beyond the corporate walls.

The Accessmaster product suite consists of:

- A modular security framework
- A single point of administration to ease deployment of an organisation's security policy and consistently enforce it.
- A number of software modules, namely:

    - Policy-based Network security manager for enterprise wide connectivity
    - Policy-based single sign-on for intranet and extranet users
    - Policy-based single sign-on for enterprise users
    - Policy-based PKI manager for web and e-mail applications
    - Policy-based user security manager for distributed systems

The majority of Accessmaster modules are available for the following operating systems:

- UNIX: AIX, Solaris, HP-UX, Linux
- MVS: RACF, ACF2, Top Secret
- Windows NT and 2000
- NetWare Lotus Notes OSF DCE

The main focus of this software suite is on unifying an organisation's access into one point of administration. Although it is all policy based, it is specific to the user entity (users and their related access activities). Because modules are purchased individually, and the software aims itself at the upper enterprise market, implementations and acquisitions are costly. The software does not provide for complex rules or for extensibility for its predefined entities. It does offer a comprehensive coverage of access related security and the centralised management thereof.

### 3.2.2 Proteus by NOWECO

*Proteus is a software tool to manage information security. Proteus contains a comprehensive questionnaire of about 900 questions that assist you in auditing your information security management system according to ISO17799.* [NOW 2002].

This product can be used to assist in the auditing of information security in terms of ISO 17799. It can be helpful in analysing an organisation's information security management system for gaps. It also allows the administrator to create reports to inform on and document the status of the information security management process.

The product consists of 2 parts. The first part is an authoring site, which is used to create profiles of the organisation and its people. This is also used to create questionnaires with regards to the organisation's security policy. The second part of the product provides the users with a means of completing these questionnaires. The questions cover controls as mentioned in BS7799: 1999. In fact, Proteus comes with BS7799.

Although the product plays a role in the enforcement and assurance phases of the security lifecycle, it does not offer any other functionality to assist the ISO. Its main focus is to provide the ISO with a means to analyse the information security environment of the organisation.
Besides the questionnaire feedback aspect, this system does not provide much else relating to the automated management of information. It does not provide for any active monitoring or measuring of information security related activities.

### 3.2.3 Control SA by BMC Software

*Provides effective enterprise-wide security management from a central point of control on a large variety of platforms and applications*
[BMC 2002].

Control SA by BMC Software claims to provide complete, unified and effective enterprise wide security management from a central point of control. The solution affords complete management of all users, their access to computing resources and security policies across a large variety of platforms and applications.

All actions are transaction based, are executed online in real time and can be tracked from initiation to completion. A two-way management infrastructure is provided; there is thus a bi-directional information flow between the managed security systems and the Control SA central repository. Roles can be assigned to users and groups and there is a central point of control.

The main focus of this application is specifically on password management and user administration. The entities are therefore predefined. Although the system pulls together feedback from disparate security related systems, it still does not provide for cross-feedback system rules to cater for the handling of softer information security issues (for example, a user that logs onto his or her organisation's network with a login account belonging to another user who is on leave).

The advantage of this system is that it makes the centralized control of user permissions very easy for large-scale organisations.

### 3.2.4 Open e-Security Platform by e-Security

*The e-Security Management Platform aggregates, standardizes, analyses and reports all security event information from multiple devices across the enterprise (Firewalls, Intrusion Detection Systems, Anti-Virus, VPNs, etc.) in a centralized console in real-time. This information is then correlated with the Security Focus Attack and Vulnerability Database, the most comprehensive database of known threats available, to deliver customers insight into their vulnerabilities, expert advice, and recommended steps toward remediation.*
[ESE 2002].

The Open e-Security Platform brings together all the components of the security infrastructure, including network hardware and software point products. It has the ability to monitor an organisation's entire distributed security environment from a single console and provides multiple views, including:

- Information security tools and point products (firewalls, etc.)
- Applications and services (OS, databases, e-mail)
- Other security sources (ERP security)
- Correlation of output from each security source
- Meaningful display alerting the manager to incidents that are potentially related.

The Open e-Security Platform can also generate reports useful for identifying trends, spotting vulnerabilities, and supporting policy-making decisions. The Open e-Security Platform consists of a central processor and a database that links to various security point sources through rule-based e-Security agents. The e-Security agents allow for direct communication via SNMP v1/v3 between the platform console and point products. In some cases 'Meta Agents' are created to monitor

other agents in order to provide more sophisticated correlation of events in the distributed security environment.

The Open e-Security Platform provides a comprehensive security management product that allows for complex rules and interpretations of monitored activity. It does, however, make use of predefined agents only, created to monitor existing point products. The product does not provide for the monitoring of entities that are not linked to point products (for example, the monitoring of software for approved changes). Once again it is intended as an enterprise entry-level product.

### 3.2.5 Trusted Global Security Manager by Trustworks

*Trusted Global Security Manager (TGSM) is a comprehensive network security management platform that provides a corporate-wide platform for centrally managing TrustWorks Client, Server and Gate VPN/firewall agents, Cisco(tm) IOS gateways and PIX Firewalls, and Check Point FireWall-1/VPN-1 Gateways.* [TRU 2002].

Using Trusted Global Security Manager, an administrator can define a so-called Global Security Policy (GSP), from which local security policies applicable to users, servers and gateways can in turn be derived and managed. The Global Security Policy provides a means by which enterprise security policies can be defined, irrespective of the underlying devices that actually enforce security. It also provides administrators with the ability to centrally modify the security policy and to change access rules for any secure node in the organisation.

Again this is a product focused specifically on access control, particularly that of users.

### 3.2.6 PoliVec by PoliVec

*The PoliVec security management product suite automates the complete cycle of defining, detecting, deploying, and documenting policy compliance throughout the enterprise. We are the only company that offers an end-to-end set of integrated, policy-driven security management products.* [POL 2002].

The PoliVec security management suite consists of the following products:

- PoliVec Builder
- PoliVec Scanner
- PoliVec Enforcer

The PoliVec products can be purchased separately.

PoliVec Builder allows IT professionals to quickly customize the security policy implementation standards needed to minimize security risks.

- It automates the development of a security policy
- It generates implementation standards (specific to an operating system) that are compliant with the organisation's security policy
- It provides 3 policy templates applicable to general organisation types.

It also assists in the creation of a specific security policy document and the creation of implementation standards for specific operating systems, including:

- Windows NT 4.0 and 2000
- Solaris
- HP-UX
- AIX
- Linux
- Novell Netware.

PoliVec Scanner audits networks for inconsistencies and vulnerabilities and compares them to the security policy. The security policy generated with the use of PoliVec Builder can be used and is imported into PoliVec Scanner.

It also allows for remote correction of configuration errors. An administrator is able to modify remote system settings for password management, user accounts, audit event logging and account lockout features. In addition, remote services can be stopped and started. PoliVec Scanner does not make use of agents for remote configuration management, but is limited to the Microsoft Windows NT/2000/XP environments.

PoliVec Scanner has a built in password-cracking utility to identify insecure passwords pro-actively.

The third product in the suite, PoliVec Enforcer, enables real-time monitoring and enforcement of an organisation's security policy as it is defined in PoliVec Builder across large, complex networks and also provides notifications when a system is out of compliance. It provides a central management console that proactively detects security vulnerabilities in real time.

It is aimed at the following environments:

- Linux
- Unix
- Windows

The product consists of agents that run security-related tests on target-devices and host computers. There is an Agent Manager that communicates with the agents, and a Controller that manages the entire system. Lastly, there is the graphical user interface, which provides the interface for administration and operation.

The suite is a comprehensive security management tool, but yet again, limited almost entirely to the user entities and their related accesses. The PoliVec Enforcer does not provide for custom developed measurement and monitoring agents to plug into the system, and the PoliVec Scanner is limited to the Microsoft Windows environment. In addition, although the PoliVec Scanner and PoliVec Enforcer products integrate with PoliVec Builder, the products run separately. This means that the administrator has to work through three management consoles.

### 3.2.7 SAM Jupiter by Systor

*SAM Jupiter sets a new standard for Security Management: The new security solution not only offers secure access privileges for employees, business partners and customers but also Premium Security Management based on Security Provisioning and Automated Identity Management. [SYS 2002].*

SAM Jupiter makes use of a single sign-on set-up with automated security management, and support for role based access control procedures. Using SAM Jupiter an organisation can manage user authorizations and privileges, passwords as well as resources across multiple platforms, via automated identity management. SAM Jupiter

promotes a feature called 'targeted security provisioning' in which a new user is assigned access rights based on processing of the user's properties. These would be properties such as job description, unit, location etc. The system can make use of human resources systems, corporate directories and organisation databases as sources of information to utilise for targeted security provisioning. SAM Jupiter allows for integration with existing point security systems.

SAM Jupiter is specifically aimed at the central management of user access control.

### 3.2.8 VigilEnt Policy Center by Pentasafe

*Provides a comprehensive, security policy management tool that automates the creation, distribution and management of corporate security policies. Employs web technologies to deliver policies so corporate security officers never have to print and distribute policy manuals again. Easily verify employee policy knowledge via on-line quizzes. Allows reporting and tracking of security incidents through the Web. VigilEnt Policy Center incorporates best practices for information security policies from Charles Cresson Wood, the recognized expert in information security and author of Information Security Policies Made Easy* [PEN 2001].

Policy Center provides an automated means by which to create and implement security policies. It also provides, and makes use of, best practice templates. Policy documents can be distributed to employees and online tests are performed to gauge their understanding and comprehension of the security policy.

The system is web-based and provides a centralized point of administration for the importing, adding and modifying of existing policies. It also provides a customisable web-based user interface that allows employees to review policies.

As with Proteus by NOWECO, the product plays a role in the enforcement and assurance phases of the security lifecycle but does not offer any other functionality to assist the ISO. It does not provide functionality to automate information security management and does not provide for the active monitoring or measuring of information security related activities.

## 3.3 Comparative Matrix of Current Systems

### 3.3.1 Comparative Matrix Criteria

A set of criteria was defined based on the functionality provided by current systems, and the problem areas to be addressed by this research. The comparison will enable us to identify what the current systems have in common, and where they are possibly lacking.

The criteria that will assist in addressing the issues of requirements for creation of entry level, flexible systems with potential for expansion, as well as complex sets of rules are all underlined. The full set of criteria used in the comparative matrix is as follows:

1 *Governed By Security Policy* - Does the system allow the information security officer to configure the application to reflect what is defined in the organisation's security policy? Will the information security officer be able to make use of the application to facilitate at least one of the three security phases (discovery, action, proof) based on the security policy?

2 *Active Feedback & Monitoring* - Does the system provide a way, using software agents for instance, to allow for feedback from security related activity to the management system for interpretation by the information security officer?

3 *Active Management (Auto Enforced)* - Will the information security officer be able to actively configure and adjust security related settings from within the application itself (for example, the configuration of user account password rules for domain logins)? Normally this would be done on the authentication server, but in some applications this function can be managed from within the security management tool.

4 *Only Predefined Measurements* - Can the organisation define their own set of measurements, or do they have to rely on measurements put in place by the system developers?

5 *Expandable* - Will an organisation be able to expand on what the application can measure? Is it limited to one operating system environment for instance?

6 *Extensible* – Does the system allow the flexibility to add fields and information dynamically to entities and in such a way extend them?

7 *Generic Entities* - Does the application make use of predefined entities, such as users, or does it allow for the definition of custom entities particular to the organisation?

8 *Centrally Managed* - Does all management occur from a central console?

9 *Cross Measurement Rules* - Does the application allow implementation of rules that extend beyond the interpretation of only a single activity's feedback, and also for different activity feedback to be used in rules?

10 *User Friendly* - Does it have a graphical and intuitive user interface?

### 3.3.2 Comparative Matrix

| Product | 1) Governed By Security Policy | 2) Active Feedback & Monitoring | 3)Active Management (Auto Enforced) | 4) Predefined & Custom Defined Measurements * | 5) Expandable |
|---|---|---|---|---|---|
| Accesmaster | 2 | 2 | 2 | 1 | 2 |
| Proteus | 2 | 0 | 0 | 0 | 0 |
| Control SA | 2 | 2 | 2 | 1 | 2 |
| Open e-Security Platform | 2 | 2 | 0 | 1 | 2 |
| Trusted Global Security Manager | 2 | 2 | 2 | 1 | 0 |
| PoliVec | 2 | 2 | 2 | 1 | 2 |
| SAM Jupiter | 2 | 0 | 2 | 1 | 0 |
| VigilEnt Policy Center | 2 | 0 | 0 | 0 | 0 |

Figure 1 a: Comparative Matrix of Current Enterprise Security Management Systems

**0 – Not Implemented**
**1 – Partially Implemented**
**2 – Fully Implemented**

* In this case a 0 indicates that no measurements are implemented, a 1 indicates that the product supports predefined measurements and a 2 that predefined and custom measurements are supported,

| Product | 6) Extensible | 7) Generic Entities | 8) Centrally Managed | 9) Cross Measurement Rules | 10) User Friendly |
|---|---|---|---|---|---|
| Accesmaster | 0 | 0 | 2 | 0 | 2 |
| Proteus | 0 | 0 | 2 | 0 | 2 |
| Control SA | 0 | 1 | 2 | 0 | 2 |
| Open e-Security Platform | 0 | 0 | 2 | 2 | 2 |
| Trusted Global Security Manager | 0 | 0 | 2 | 2 | 2 |
| PoliVec | 0 | 0 | 2 | 0 | 2 |
| SAM Jupiter | 0 | 0 | 2 | 0 | 2 |
| VigilEnt Policy Center | 0 | 0 | 2 | 0 | 2 |

Figure 1 b: Comparative Matrix of Current Enterprise Security Management Systems Continued

**0 – Not Implemented**
**1 – Partially Implemented**
**2 – Fully Implemented**

### 3.3.3 Summary

Generally the applications looked at fall into two categories. Firstly, those that assist the information security officer with the creation of the actual security policy document, and subsequent user awareness programme; secondly, those that also provide for centrally managing security related configurations, measurements and monitoring.

All the applications that provide for active security management, measurements and monitoring, do so specifically with user management and user access control in mind. They do not provide for the possibility of information security related entities in the organisation that do not explicitly involve users. One system, PoliVec, did provide for a non-access related measurement in the password cracker, but in most products this was not the case. PoliVec also did not make provision for organisations to add similar, but custom developed, measurements.

All of the applications providing monitoring and measurement functionality also provided for various types of rule based evaluations of the activity feedback to the central system. These were mostly specific to the activity being monitored only. In the case of Open e-Security Platform by e-Security however, the system did provide for the creation of complex rules, based on the feedback from more than one type of activity. It unfortunately did not provide for the custom creation of these rules.

Based on this understanding of the current enterprise security management systems, their identified functionality, as well as the additional functionality to support the creation of entry level, flexible systems with potential for expansion as well as complex sets of rules, we can define the key requirements of the prototype system to be developed. This is done in the next chapter.

# 4 Prototype Application System Design Framework

## 4.1 Introduction

The purpose of this chapter is to identify the key requirements of the prototype information security management system application to be developed. These will be based on the functionality of the products evaluated in the previous chapter, as well as the additional functionality required in order to solve the problem of costly implementations and inflexible designs. These being extensibility, catering for generic entities and allowing for cross measurement rules. This will create a reference framework on which the design of the prototype application can be based.

The prototype application must allow the information security manager to centrally monitor an organisation's information security environment, and to initiate appropriate reactions to events or activities based on the organisation's information security policy.

Furthermore, the system should be flexible in its monitored activities and events, the entities to which these apply, the information security rules to which it measures, the information that is relayed back to the central system and the environment in which it is monitoring.

## 4.2 Key Requirements

The following key requirements are presented based on our understanding of what the system should be capable of and how the system should function from a management point of view, as well as the criteria used to compare the currently available products in:

### 4.2.1 Governed By Security Policies

The system should facilitate the phases of a security cycle as described in Chapter 2 based on an organisation's security policy. An information security officer should be able to take an appropriate policy document and impose on the system the business rules therein. The system should provide for a rich set of configurations to allow for any number of complexities in this regard. For instance, the system should allow the system administrator to set up any number of combinations and levels of events to occur within a given time frame before action is taken. The information security officer should be able to measure the organisation's compliance to its security policy based on the monitored environment and rules.

### 4.2.2 Active Feedback & Monitoring

The prototype system should provide feedback of monitored activities and measurements. To accomplish this, the system needs to make use of software agents that would do the monitoring and feed information back to the system. The system will need to expose an application programming interface, or API, via which the updates to the central system could occur.

### 4.2.3 Active Management in Terms of Auto Enforcement

The system will not provide functionality for in depth configuration such as user access management, but will rather provide a limited, yet totally customisable, enforcement framework. The system will handle this by providing the ISO with the freedom to execute any custom developed scripts. These scripts are executed by the system as a result of custom defined rules, which are applied to the information fed back to it.

### 4.2.4 Predefined And Custom Developed Measurements

The system needs to provide predefined measurement agents, but should additionally provide the APIs to allow an organisation to plug in their own, custom developed, measurement services.

### 4.2.5 Expandable

Since the system needs to be applicable to any custom environment configuration, it must allow for custom monitoring services to be plugged into it as and when required. This means that application programmable interfaces (API) need to be developed and exposed, so information can be fed back to the central management system through it. It also means that these interfaces have to be platform independent. In addition these APIs will facilitate the interaction with any code custom developed by an organisation.

### 4.2.6 Extensible

The system should be flexible in how it handles the information passed to it from the monitoring and measuring services, as well as in the actual entity information that may need to be stored (i.e. information specific to a particular implementation of the system). This implies that the system should provide functionality to dynamically add information fields to any of the types of entities in the system.

### 4.2.7 Make use of Generic Entities

The system should provide the information security officer with the ability to make use of generic entities and to define and configure entities with specific attributes. An example of this would be users, software, documents etc. It should be possible to define any number of field properties for these entities, to populate them with actual information and to update the status of these entity instances as security related activities occur.

### 4.2.8 Centrally Managed

The system needs to provide a central point of management. This means that the ISO should be able to log onto a single management console from where he or she will be able to view all information fed back to the system from monitoring and measuring activities and make the necessary configuration changes.

### 4.2.9 Cross Measurement Rules

A rules engine needs to be included in the system. The rules should be able to evaluate the feedback from measurements and monitored activity and, if required, initiate the appropriate action. The engine should be able to construct simple to complex rules with which to evaluate information, either from a single source, or across disparate information sources.

### 4.2.10 User Friendly

An obvious requirement is that the system has a graphical user interface. There should be one-glance reporting, as well as more detailed reports available to the information security officer. In the case of custom defined entities and rules, dynamic reports should be generated on the fly.

## 4.3 Comparison of InfoSure with Current Enterprise Security Management Systems

| Product | 1) Governed By Security Policy | 2) Active Feedback & Monitoring | 3)Active Management (Auto Enforced) | 4) Predefined & Custom Defined Measurements * | 5) Expandable |
|---|---|---|---|---|---|
| Accesmaster | 2 | 2 | 2 | 1 | 2 |
| Proteus | 2 | 0 | 0 | 0 | 0 |
| Control SA | 2 | 2 | 2 | 1 | 2 |
| Open e-Security Platform | 2 | 2 | 0 | 1 | 2 |
| Trusted Global Security Manager | 2 | 2 | 2 | 1 | 0 |
| PoliVec | 2 | 0 | 0 | 1 | 0 |
| Security Administration Manager | 2 | 0 | 2 | 1 | 0 |
| VigilEnt | 2 | 0 | 0 | 0 | 0 |
| InfoSure | 2 | 2 | 1 | 2 | 2 |

Figure 2 a: Comparative Matrix of Current Enterprise Security Management Systems and *InfoSure*

**0 – Not Implemented**
**1 – Partially Implemented**
**2 – Fully Implemented**

* In this case a 0 indicates that no measurements are implemented, a 1 indicates that the product supports predefined measurements and a 2 that predefined and custom measurements are supported,

| Product | 6) Extensible | 7) Generic Entities | 8) Centrally Managed | 9) Cross Measurement Rules | 10) User Friendly |
|---|---|---|---|---|---|
| Accesmaster | 0 | 0 | 2 | 0 | 2 |
| Proteus | 0 | 0 | 2 | 0 | 2 |
| Control SA | 0 | 2 | 2 | 0 | 2 |
| Open e-Security Platform | 0 | 0 | 2 | 0 | 2 |
| Trusted Global Security Manager | 0 | 0 | 2 | 2 | 2 |
| PoliVec | 0 | 0 | 2 | 0 | 2 |
| Security Administration Manager | 0 | 0 | 2 | 0 | 2 |
| VigilEnt | 0 | 0 | 2 | 0 | 2 |
| InfoSure | 2 | 2 | 2 | 2 | 2 |

**Figure 2 b: Comparative Matrix of Current Enterprise Security Management Systems and *InfoSure***

**0 – Not Implemented**
**1 – Partially Implemented**
**2 – Fully Implemented**

## 4.4 Summary

The requirements of the prototype system were identified based on what the currently available enterprise security systems have to offer, as well as the problem areas this research would like to address. The requirements are as follows:

1 Governed by security policies
2 Active feedback and monitoring
3 Active management in terms of auto enforcement
4 Predefined and custom developed measurements
5 Expandable
6 Extensible
7 Use of generic entities
8 Centrally managed
9 Cross measurement rules
10 User friendly

These requirements provide the framework for the prototype system, and based on this the system itself can now be designed.

# 5 Prototype Information Security Management System

## 5.1 Introduction

***InfoSure*****,** as the prototype is named, was developed to test the feasibility of an information security management system that improves on what currently is available. The areas in which the proposed system will provide the most benefit will be in the rich cross-entity rules functionality and cost-effective implementation (extensible and expandable to the needs of an organisation).

The system is to provide a centralized management console to which custom developed measurement services can feed back results and information, allowing information security officers to evaluate this information and initiate appropriate responses. These measurement services include any custom developed application or code which monitors processes and events related to information security as defined in the information security policy. They should be able to run according to a predefined schedule, and be able to return their results to the management system for interpretation.

The current ***InfoSure*** system was designed to run on the Microsoft platform using Microsoft technologies. The example monitoring services provided are aimed at information security risks in the Microsoft NT environment. However the structure and logic of the design should prove applicable in other operating environments as well. The generic entities functionality is not Microsoft specific, and can be applied to any security related entities the information security officer wishes to keep track of.

## 5.2 Key Requirements

***InfoSure*** was designed around several key requirements. The system's aim was to implement the set of requirements as identified in Chapter 4. These requirements are:

1 Be governed by information security policies
2 Active feedback and monitoring
3 Active management (Auto enforcing)
4 Allow for predefined and custom developed measurements
5 Be expandable
6 Be extensible
7 Make use of generic entities
8 Centrally managed
9 Implement cross measurement rules
10 User Friendly

In addition, a number of measurement services had to be provided to demonstrate the system in action. These services provide mere examples of what the system can handle and do not serve as a definitive list of services. The system should be able to handle any number and variety of these measurement services.

The following section provides a conceptual diagram of the prototype system and the proposed interfaces into the system.

## 5.3 Conceptual Diagram of InfoSure

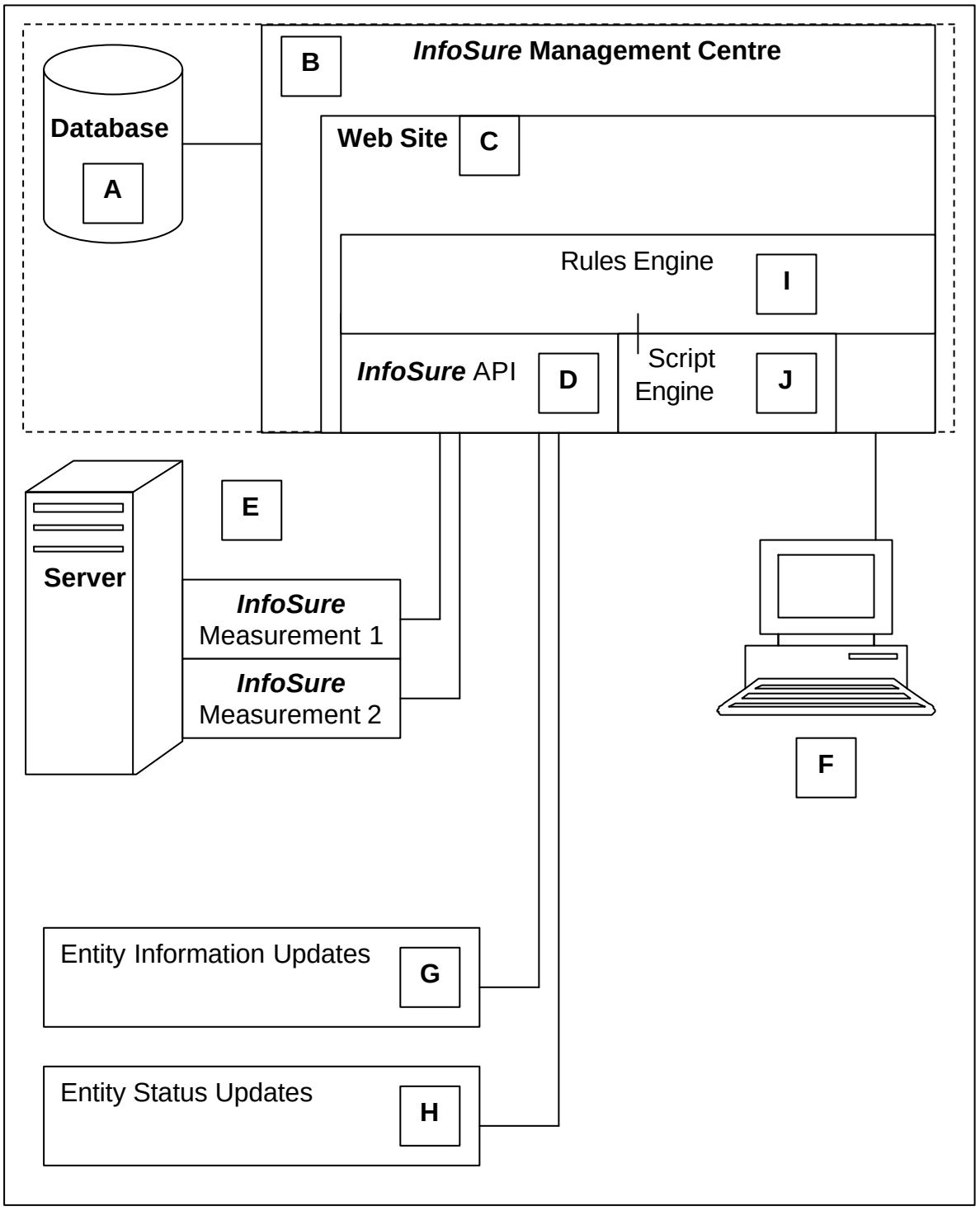


Figure 3: Conceptual diagram of the Prototype Information Security Management System

**A** ***InfoSure*** database, which houses all information security related information passed to the system, either via the web front-end, or via the APIs.

**B** ***InfoSure*** Management Centre - forms the focal administrative point of the system. Information fed back or into the system is processed here.

**C** Part of the ***InfoSure*** management centre consists of a graphical interface to the user through which measurements and other system information pertaining to the system is presented. In this case the interface is web browser based.

**D** The API layer, through which communications to and from external measurement, monitoring and other updateable services occur. The XML standard over an HTTP protocol is used.

**E** Measurements 1 and 2 represent service modules or agents that are plugged into the ***InfoSure*** network environment. Each module would provide a different set of measurements to the system, and is designed to feed back information to the central ***InfoSure*** Management Centre. There can be any number of these modules, in any number of combinations on each server in the network. It should be possible to add these modules individually to a system running on any operating system.

**F** The ISO(s) can access the system for maintenance, reporting, etc. using a browser from any workstation that is connected to a network that can see the web site.

**G** This represents the custom developed code that an organisation can use to manage instances of custom developed entities in the central system. An example would be the adding of users.

**H** This represents the custom developed code that an organisation can use to update the status of its custom defined entities in the central system. These status updates would indicate activity related to the entities, which would be of use to the information security officer in managing adherence to the organisation's policy document.

**I** The rules engine, which can be used to evaluate activity information fed back to the central management system.

**J** Based on the outcome of activity feedback evaluation by the rules engine, it is also possible to fire an event by making use of the scripting engine. This engine basically provides a shell within which custom code can be executed on the management server, by using Visual Basic script. Using this engine, e-mails can be generated, system settings changed, etc.

## 5.4 Approach taken in Fulfilment of Identified Requirements

A number of design principles were employed to fulfil the requirements imposed upon the system. These are as follows:

- Open standards;
- Dynamically generated data fields;
- A high level of user configuration;
- Measurement service APIs;
- A rules engine
- Web-based design
- Generic entities
- Custom event scripts

### 5.4.1 Open Standards

Open standards were used for the design of both the system application programmable interfaces, and APIs, as well as the transfer of the measurement service information to the system. This was necessitated by the need for accessibility to the central management system by services running on non-Microsoft platforms. The API was designed to accept XML messages, which describe the state of the monitoring process or program and the state of the measured entity. XML stand for Extensible Mark-up Language and is the de facto standard for transferring information between heterogeneous systems, and many systems already make use of this means of information transfer [W3C 2002].

The messages are transferred to the system via the commonly used hypertext transfer protocol (HTTP) [W3C 2000]. This also allows for HTTPS to be used in scenarios where sensitive information will be transferred to the central system, and security is of importance.

The result of these design elements is a system that allows for the development of multi-platform services that can still return information to the central management system, irrespective of platform.

### 5.4.2 Dynamically generated data fields

Additional fields can dynamically be added to the system in a number of areas. This facility, available from the administrative sections of the system, provides the means with which to extend the information gathered by measurement services. This provides added flexibility to both the system administrator and the developers of the services by enabling the passing of additional information across the network to the management system. The XML messages used by the API have been designed so that dynamically generated field information can be easily added to these messages, and will be discussed in greater detail later.

### 5.4.3 High level of user configuration

Almost all aspects of the system are totally user configurable, most notably the status codes. The system administrator has full control over all the status codes to be used. The system starts out with no defined statuses. Each identified status can be assigned a code and a description of this status. As an additional visual setting, the administrator can also assign predefined colours to each of these statuses. Assigning colours can assist the system administrator to easily distinguish between various statuses or types thereof (for example he/she could assign red to certain critical statuses to distinguish from the less critical ones at a glance).

### 5.4.4 Measurement service APIs

The API of the ***InfoSure*** system is based on passing XML messages to relevant URLs via HTTP. As discussed before, these URLs then parse the XML and update the central management system. Currently, the parsing is done using active server pages, or ASP, and the Microsoft XML document object model (DOM), but theoretically could be any application that can be invoked via an HTTP call and accept XML messages. This would also allow a more secure interface. A dynamically linked library (DLL), for example, would be more difficult to modify than the interpreted ASP code.

For the updates, and passing back of information to the central server, the following XML message template is used:

```
<?xml version='1.0'?>
<measurementmodule loginid='first' password='first'>
    <item id='status_code' value='{status code}' type='p'/>
    <item id='last_run' value='{date and time}' type='p'/>
    <item id='error_str' value='{error string}' type='p'/>
    <item id='status_detail' value='{text detail}' type='p'/>
    <item id='{dynamic field}' value='{text detail}' type='d'/>
</measurementmodule>
```

The 'loginid' and 'password' attributes identify the specific measurement service instance. The other elements comprising this message are:

**status_code**: The predefined status codes as set up by the administrator of a specific implementation.

**last_run:** The date and time stamp of the last execution of the measurement service.

**error_str**: A free text field containing the status

information of the service itself, which can be passed back to the central management system.

**status_detail**: A free text field containing the actual detail regarding the measured event or process that can be passed back to the central management system for display purposes.

The last item in the example code above is used when dynamically created field data must be passed back to the central management system. The id attribute should contain the unique name assigned to the dynamic field, and the value attribute should contain the value of the dynamic field. By passing a‘d’ in the type attribute the system is informed that the particular field is dynamically generated, and will then handle it accordingly.

Similar APIs are used for updating defined entity instances and their status history. These can also be updated manually via the web front-end.

For maintaining (adding, editing, deactivating, and activating) a predefined entity instance (such as a specific user) the following XML template is used:

```
<?xml version='1.0'?>
<entityupdate type='{entity_type_name}'>
 <header>
  <datetime>{datetime}</datetime>
 </header>
<entitydetail>
 <entity type='add'>
  <fielditem id='{entity_option_name}' value='{value}'/>
            .
            .
```

```
            .
  </entity>
  <entity type='edit' id='{option_id}' value='{value}'>
   <fielditem id='{entity_option_name}' value='{value}'/>
            .
            .
            .
  </entity>
  <entity type='deactivate' id='{option_id}' value='{value}'>
  </entity>
  <entity type='activate' id='{option_id}' value='{value}'>
  </entity>
 </entitydetail>
</entityupdate>
```

The elements comprising this message are:

| | |
|---|---|
| **entity_ type_name:** | This indicates the predefined entity type's name used during set-up. This value uniquely identifies an entity type. |
| **datetime:** | The date and time stamp of the message. |
| **entity_option_name**: | The unique name of the entity field. |
| **value**: | The value of the mentioned attribute |
| **option_id**: | The unique name of the entity field that has been marked as the unique identifier for entity instances. |

For updating an entity instance's status history, the template is:

```
<?xml version='1.0'?>
 <statusupdate type='{entity_type_name}'>
  <header>
   <datetime>'{datetime}'</datetime>
  </header>
  <statusdetail>
   <entity 'id='{option_id}' value='{value}'>
   <statuscode value='{status}' field='{entity_option_name}'
                                      fieldvalue='{value}'/>
   </entity>
  </statusdetail>
</statusupdate>
```

The elements comprising this message are:

| | |
|---|---|
| **entity_ type_name:** | This indicates the predefined entity type's name used during set-up. This value uniquely identifies an entity type. |
| **datetime:** | The date and time stamp of the message. |
| **entity_option_name**: | The unique name of the entity field. |
| **value**: | Value of the mentioned attribute |
| **option_id**: | The unique name of the entity field that has been marked as the unique identifier for entity instances. |

### 5.4.5 Rules engine

***InfoSure*** allows an administrator to define sets of rules (which could be based on the organisation's security policy) to determine actions to take when certain specified states are returned by any of the measurement services. Rules comprise a number of individual rule items, which combine to form rule elements, which in turn can be combined to create complex sets of logic. These elements and items are combined using the **OR** and **AND** Boolean logic operators.

Each rule element can consist of:

- A rule item on its own
- Two rule items joined by Boolean logic
- A rule item and element joined by Boolean logic
- Two rule elements joined by Boolean logic

A rule can be likened to a mathematical expression, and is interpreted in a similar way. For example, figure 4 shows how the rule logic:

((*condition* **OR** *condition*) **AND** (*condition* **OR** *condition*)) **OR** (*condition* **OR** *condition*)

is represented in the system.

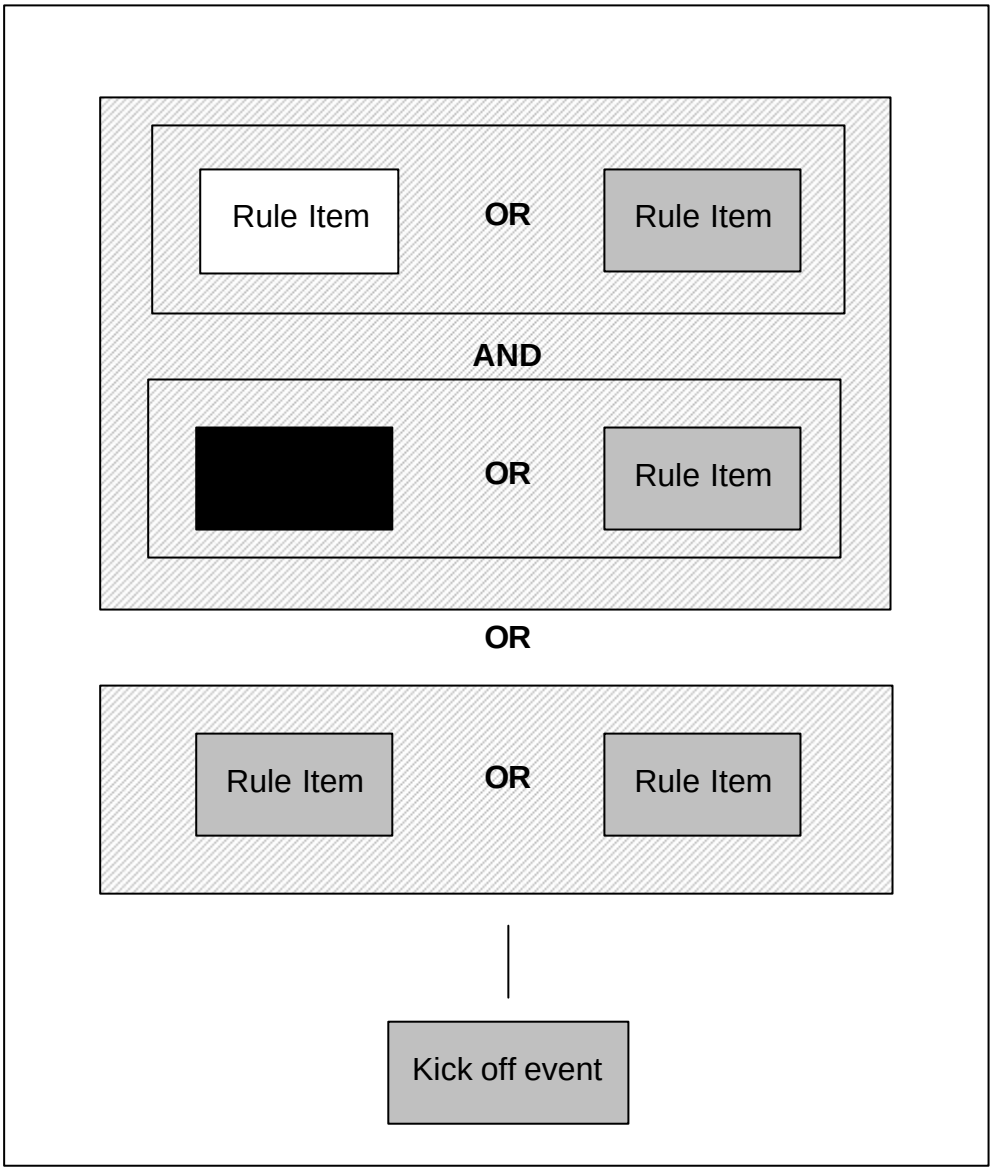


Figure 4: Generic example of a rule in the prototype system

Once a rule has been broken, its associated event is kicked off. Events are predefined sets of actions to be taken. Additionally, tolerance levels can also be built into the rule evaluations.

A practical example is, for instance, a case where a system administrator needs to be notified if any user tries to log on to the network, without the necessary permissions to do so. In this scenario, the rule would state that if a monitoring agent returns a status indicating a network access violation, then an action is kicked off. In this case the event would be e-mail to the administrator.

As mentioned, the rules engine allows for a level of tolerance to be built in so the rule can be fine-tuned to only notify the administrator in case of repeat offenders. So, for example, it may be configured to only kick of the notification action if a user has made more than 3 attempts to access the network.

Because the rules engine also allows for more complex rules, involving Boolean logic and different activity measurements, the scenario can be further expanded to include additional checks (see figure 5). The rule in the scenario above can be modified to only notify the administrator of unsuccessful access attempts where the user is also on leave. A separate monitoring agent would return information regarding leave status to the central management system.

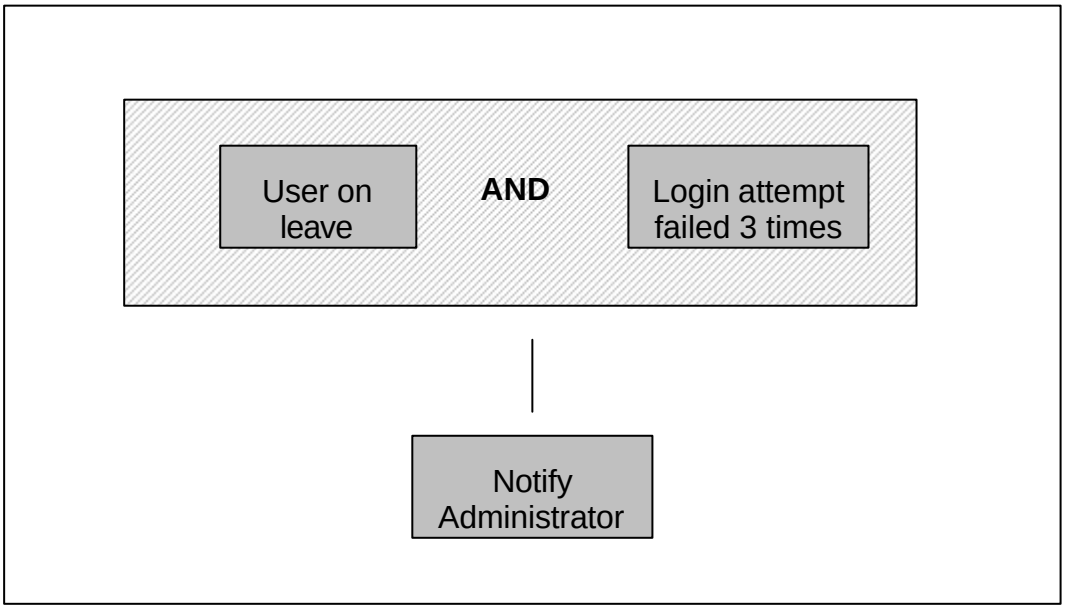


Figure 5: Representation of an actual rule in the prototype system

A slightly different type of rule set-up is also available; that which associates entity options with predefined actions based on certain status updates. The rule would evaluate that if a specific instance of a specific entity's latest status were of a predefined nature, and the option for that instance has a certain value that an action needs to be kicked off.

These rules are also used to generate dynamic reports. The report would display a listing of all the entity instances that have failed the rule's check. The information security officer is able to limit this listing to certain date ranges of occurrence.

### 5.4.6 Web-based design

The system's user interface is browser based, and was developed using active server pages (ASP), running on Microsoft's IIS 5.0 web server software [MIC 2001]. This browser-based approach allows users of the system to log in from any computer on the network, or even over the Internet.

All the system's application programmable interfaces are exposed to be accessible via HTTP.

### 5.4.7 Generic Entities

The system allows for custom defined and configured entities. Entity types, such as users, can be created and entity field options assigned to each type. 'User' for instance would typically have options such as employee number, first name, last name, etc. One of these options will need to be flagged as the unique identifier. This would then be used to uniquely identify instances of each entity type. In terms of a user entity, this would most likely be the employee number.

The information relating to instances of these entities can be entered into the system either by means of the browser-based front-end, or by making use of specifically defined APIs. These APIs allow the maintenance of these entity instances to be automated. There are also APIs defined to handle status updates of activities related to these instances. An example would be the logging of remote accesses to the network.

### 5.4.8 Custom Action Scripts

Custom scripts can be created that will be executed based on the outcome of rules and feedback information. This is accomplished by making use of the Microsoft Windows Scripting Host [SHU 2001]. The Windows Scripting Host is a language-independent scripting host for 32-bit Windows platforms. Windows Scripting Host enables scripts to be executed directly on the Windows desktop or command console, without the need to embed those scripts in an HTML document. The scripts themselves can in turn instantiate other component object model (COM) components and execute methods and functions of those components.

Additionally two measurement services have been custom developed for the prototype system. These demonstrate the use of the system's APIs.

## 5.5 Currently Included Measurement Services

The measurement services included with the prototype are:

- Easy password cracker
- Microsoft SQL Server 'sa' account password checker

These measurement services just give an indication of the type of functions a measurement service can provide. The range of applications is limitless, and is up to what the organisation wants. Obviously this does present an issue if an organisation does not have the technical skills in house to develop measurement services. However, the solution allows for one to add measurement services to those already supplied with the application.

### 5.5.1 Easy password cracker

This measurement service was developed using the C++ language and runs as a Microsoft NT service on the hosting server. The purpose of the service is to run a scheduled check on Microsoft Windows 2000 user accounts set up on the system, attempting to detect easily guessable passwords that would compromise system security. If any passwords do not qualify as secure, the central management system is notified and specific information on the offending accounts is transmitted.

The service consist of 3 distinct areas of functionality, executed in sequence:

- Password Dump
- Password Cracking
- Central Management System Update

#### 5.5.1.1 Password dump

The first section dumps the password hashes from Window 2000's SAM database, whether or not SYSKEY is enabled on the system, by making use of Todd Sabin's pwdump2 utility [SAB 2000]. By default, only administrators have this right, so this program does not compromise Windows 2000 security. It uses a technique known as DLL injection.

#### 5.5.1.2 Password Cracking

This section takes the output from the password dump, and tries to 'guess' the correct password by means of a dictionary attack. This involves running through a text file of words, encrypting each word found, and comparing that to the encrypted account password from the

dump. If an account is compromised, the account and compromised password is noted. It was developed using the L0phtCrack 1.5 source code.

#### 5.5.1.3 Central Management System Update

This section takes the result of the password-cracking attempt and returns it to the central management system via the exposed API using HTTP and XML. The result would include a status code (as predefined in the management system by the information security officer)

The service also allows the user to create a schedule for running these checks at set intervals.

### 5.5.2 Microsoft SQL Server 'sa' account password checker

The aim of this measurement service is to run scheduled checks on Microsoft SQL servers to try and see if the administrative 'sa' account can be compromised using a dictionary attack, or if the 'sa' account might have a blank password. It makes use of a custom word list that can be modified by the organisation to suit their requirements.

## 5.6 Technical Requirements

The ***InfoSure*** system was developed on the Microsoft environment, and makes use of Microsoft technologies. The central management system runs on a Windows 2000 server, which has IIS 5.0 loaded, and has ASP enabled. The measurement services can run on any platform, as long as it can communicate with the central management system via HTTP or HTTPS calls. The two demonstration services supplied with the prototype run on Windows 2000.

## 5.7 Summary

The prototype was developed to test the feasibility of an information security management system that improves on what is currently available in terms of expandability, extensibility and rules functionality. The system needed to be cost-effective for an organisation to implement, and was designed around several key requirements.

The system's aim was to implement the following:

- Be governed by information security policies
- Active feedback and monitoring
- Active management (Auto enforcing)
- Be expandable
- Be extensible
- Allow for predefined and custom developed measurements
- Make use of generic entities
- Centrally managed
- Implement cross measurement rules
- User Friendly

The following design principles were employed to fulfil the requirements:

- Open standards
- Dynamically generated data fields
- A high level of user configuration
- Measurement service APIs
- A rules engine
- Web-based design
- Generic entities
- Custom event scripts

In addition, a number of custom developed measurement services were supplied. The system still has major scope for improvement and refinement, and only a relatively small part of the system's actual potential was developed for demonstration. In the next chapter a number of additional developments to improve the functionality of the current system are discussed.

# 6 Potential Additional Development

Although a substantial amount of the core functionality has been included in the ***InfoSure*** system, there are a number of areas where improvements and additional development can be made. The main areas in which these improvements can be made relate to the rules engine, the actions section and the measurement services.

## 6.1 Extending Rules Engine Functionality

The idea of the rules engine can be extended to combine generic entity based rules with rules for the standard measurement services of the system. Currently there are separate rule engines for the generic entities and the measurement services section. Extending it would allow for complex rules to be created involving decisions based on results from measurements and the entity status updates, e.g. rules that involve the results of weak password checks with 'user' entities that do or do not have remote access.

## 6.2 Actions

The actions section can also be improved to provide, in addition to the custom action creation, 'out of the box' actions such as commonly used functions e.g. e-mail, SMS etc.

Additionally, it could be useful to incorporate functionality that would allow the ISO the ability to kick off actions manually from the central management console as opposed to the rules initiating the actions. This would effectively provide a means to incorporate active configuration management into the system.

## 6.3 Measurement Services

Only two measurement services were supplied as part of this prototype system – the easy password cracker and the Microsoft SQL server ‘sa’ account password checker. Greater value would be added to the system by the inclusion of more measurement services in the prototype.

# 7 Hypothetical Scenarios

To demonstrate the use of ***InfoSure***, we will construct a hypothetical situation for organisation XYZ, and then provide the solutions that can be implemented using ***InfoSure***.

Suppose that organisation XYZ has a security policy, and they want to start enforcing it and monitoring compliance. The policy contains statements that lend it to automated monitoring and enforcement, and these statements can be grouped as follows:

- *Non-entity specific policy statements*: These are statements that are not applied to any entity specifically, but are instead applied to all entities within a type. An example of this is strong passwords. It applies to all users.

- *Entity specific policy statements*: These are statements that can apply to individual or to subsets of entities. An example of this would be a statement regarding RAS access, which users or groups of users are granted permission to utilize.

- *Non-user statements*: Other entities can also be involved in the policy, for example statements relating to updates to software.

- *Statements that stipulate combinations of events to occur*: These are statements whose checking can involve the combination of more than one event. An example of this is users logging in using other users' logins. Thus one could identify through event one that a user is marked as on leave; event two indicates that that user's login has been used. The combination of the two would indicate that a user might have used another user's login.

Covering all of the above-mentioned types of policy statements, the following represents extracts from organisation XYZ's security policy document:

1) **Policy Statement:** *Users must not have weak passwords.*
**Policy Compliance Indicators**: *Passwords should not be:*

- *Words that are found in a dictionary.*
- *Words that are common usage words.*
- *Words that form patterns (i.e. 12345)*

2) **Policy Statement:** *A register of all authorized remote access users will be maintained.*
**Policy Compliance Indicators**: *The contents of the register will be provided on a weekly basis to the information security officer, who will then have to re-authorize the list.*

3) **Policy Statement:** *The remote access server shall log all connections, sessions and related user-ids.*
**Policy Compliance Indicators**: *A list containing all users who logged on but who are not on the register of authorised users (Clause above), shall be provided to the information security officer on a daily basis.*

4) **Policy Statement:** *Changes to software shall be approved prior to implementation.*
**Policy Compliance Indicators**: *A list of all changes made to software shall be provided to the information security officer on a weekly basis.*
*A list of non-approved software changes shall be provided to the information security officer on a daily basis.*

5) **Policy Statement:** *No one can access the network while on leave*

**Policy Compliance Indicators***:* *No one can log onto the network with a user's login while that user is on leave.*

Let's examine each of the statements, compliancy indicators, and how it can be handled by the ***InfoSure*** system.

## 7.1 Strong passwords

**Policy Statement:** *Users must not have weak passwords.*

**Policy Compliance Indicators**: *Passwords should not be:*

- *Words that are found in a dictionary.*
- *Words that are common usage words.*
- *Words that form patterns (i.e. 12345)*

In this case the information security officer can make use of the 'Easy Password Cracker' measurement service to check the passwords of users on a regular basis, and to feed the results back to the central ***InfoSure*** system. The information security officer will be able to ensure that in cases where the service was successful (i.e. where a user was using a weak password) the appropriate administrator will be notified. It would also appear in the reporting.

## 7.2 Register of authorized remote access users

**Policy Statement:** *A register of all authorized remote access users will be maintained.*

**Policy Compliance Indicators**: *The contents of the register will be provided to the information security officer on a weekly basis who will then have to re-authorize the list.*

The information security officer can use the generic entities section of the ***InfoSure*** system to set up a 'user' entity type. Entity field options can then be added to this to represent, for instance, a unique employee number, name, surname, etc. as well as a field to indicate if the user is allowed remote access.
This can then be kept up to date, either automatically using a feed through the relevant API, or manually by means of the web interface.

## 7.3 Remote access information to be logged

**Policy Statement:** *The remote access server shall log all connections, sessions and related user-ids.*

**Policy Compliance Indicators**: *A list containing all users who logged on but who are not on the register of authorised users (Clause above), shall be provided to the information security officer on a daily basis.*

It is possible to update the status history of the different entity instances set up in the system either by means of an API or by using the web front-end. In this case as logins occur, a service can be used to feed information through to the ***InfoSure*** system for logging via the 'Entity Status Update' API. A rule can be set up on the 'Allow Remote Access' field of the user entity type, which will allow the information

security officer to identify unauthorised accesses using the dynamic reports.

## 7.4 Pre-approved changes to software

**Policy Statement:** *Changes to software shall be approved prior to implementation.*

**Policy Compliance Indicators***: A list of all changes made to software shall be provided to the information security officer on a weekly basis.*

The information security officer can use the generic entities section of the ***InfoSure*** system to set up a 'software' entity type. Entity field options can then be added to this to represent, for instance, a unique software serial number, package name, version, etc. as well as a field to indicate last changes and type of changes made.
This can then be kept up to date, either automatically using a feed through the relevant API, or manually by means of the web interface.

**Policy Compliance Indicators**: *A list of non-approved software changes shall be provided to the information security officer on a daily basis.*

It is possible to update the status history of the different entity instances set up in the system either by means of an API or by using the web front-end. In this case, as changes occur, they can be entered into the system via the web front-end. A rule can be set up on, for instance a 'Non-Approved changes' field of the software entity type. This will allow the information security officer to identify, by using dynamic reporting, software which has had non-approved changes made.

## 7.5 Users only allowed to use own logins

**Policy Statement:** *No one can access the network while on leave*

**Policy Compliance Indicators***:* *No one can log onto the network with a user's login while that user is on leave.*

This can be accomplished by, once again, making use of an entity type of 'users'. In this case it could contain an entity field option of 'on leave'. This can then be kept up to date manually or via the appropriate API, as mentioned before.
Login information can be fed to the system using the normal service instance set-up. Using the rules, a check can be executed for each login attempt, to see if the associated user has been flagged as on leave or not. An appropriate action can then be initiated.

## 7.6 Summary

Policy documents can contain statements pertaining to specific entities, as well as statements that are non-entity specific. Statements don't necessarily involve users, and many statements can involve the actions of more than one type of activity. ***InfoSure*** could be used to monitor for compliance in any of these statement scenarios.

# 8 Conclusion

***InfoSure*** was designed around several key requirements. The system's aim was to implement the following:

1 Be governed by information security policies
2 Active feedback and monitoring
3 Active management (Auto enforcing)
4 Be expandable
5 Be extensible
6 Allow for predefined and custom developed measurements
7 Make use of generic entities
8 Centrally managed
9 Implement cross measurement rules
10 User Friendly

1 *Governed by organisational security policies*: It is possible to map many statements contained in a policy document, although not all. There will always be statements for which compliancy cannot be measured using automated methods (for example, a statement that prohibits employees from writing down any of their passwords).

2 *Active feedback and monitoring:* The ***InfoSure*** system provides adequately for feedback by means of a number of APIs available for this purpose.

3 *Active management (Auto enforcing):* The ***InfoSure*** system provides for a certain level of auto enforcing, but the system does not offer much in terms of active management. It does not allow for central management of decentralised security (the information security officer can not, for instance, update RAS login rules from the ***InfoSure*** system. He or she would still have to apply these configuration rules on the physical server.)

4 *Expandable*: The ***InfoSure*** system would be a low cost solution to implement and does allow for expansion with regards to adding additional entities, measurements and rules. The organisation would also have the ability to create its own measurement services to feed back information back via the ***InfoSure*** APIs.

5 *Extensible*: The ***InfoSure*** system provides for the dynamic addition of fields to any of the measurement related objects.

6 *Custom and predefined measurements:* A number of predefined measurements were supplied, and an API provided for the integration of custom developed measurement services an organisation would want to add.

7 *Generic Entities:* No restrictions are placed on the information security officer when it comes to defining entities. Any number of entities can be configured in any number of ways.

8 *Centrally managed*: The system is centrally managed as it was developed as a web-based solution, which can be accessed from any browser.

9 *Implement cross measurement rules:* Cross measurement rule functionality was included in the system, but rules pertaining to generically defined entities were defined separately. The rules engine can easily be extended to also incorporate rules relating to generic entities.

10 *User Friendly:* The system does present a user-friendly graphical interface, and makes use of reports that allow the information security officer to obtain interpreted information at a glance. There is, however, a lot more that can be done in terms of reporting to refine the information displayed, and the opportunity exists for many new reports to be implemented.

The conclusion drawn from the research is that the system does meet these requirements to a certain degree, catering for each objective and handling each issue acceptably. The major shortcoming is in terms of active management. The problem is that in order to actively management a security configuration, the system would have to be designed to specifically be able to integrate with that tool. This would affect the systems generic and customisable nature. On the other hand it does restrict the information security officer of maintaining security related configurations from a central point. This could conceivably be handled to a certain degree by extending the action functionality to include actions that can manually be executed.

In conclusion, from the research performed it has been found that not only is there a gap in the market for comprehensive, affordable software to monitor and manage information security, but that a system such as ***InfoSure*** can fill this gap adequately and efficiently.

The focus of ***InfoSure*** is on providing the central management console as generic as possible, while allowing the organisation to develop custom measurements and monitoring services. This is in contrast to current enterprise security management solutions that are restrictive in their customization abilities, but offer a comprehensive range of pre defined information security measurements.

This might be restrictive in organisations where technical skills are lacking. The prototype system illustrates not only that the product is possible, but that it can be configured and improved on in time. The design is sound and flexible as well as extensible. The prototype, meeting the market requirements either in its developed implementation example or facilitated by it's basic design, proves the concept of the proposed system to be a feasible, and practical product in today's information driven world.

# Appendix A

# InfoSure System Walkthrough

This section will provide a short walkthrough of the actual system. How to navigate through it, what each section is for, and how it ties in to the design objectives. Lastly it will give a breakdown of the time spent on development of the system and the design specifications.

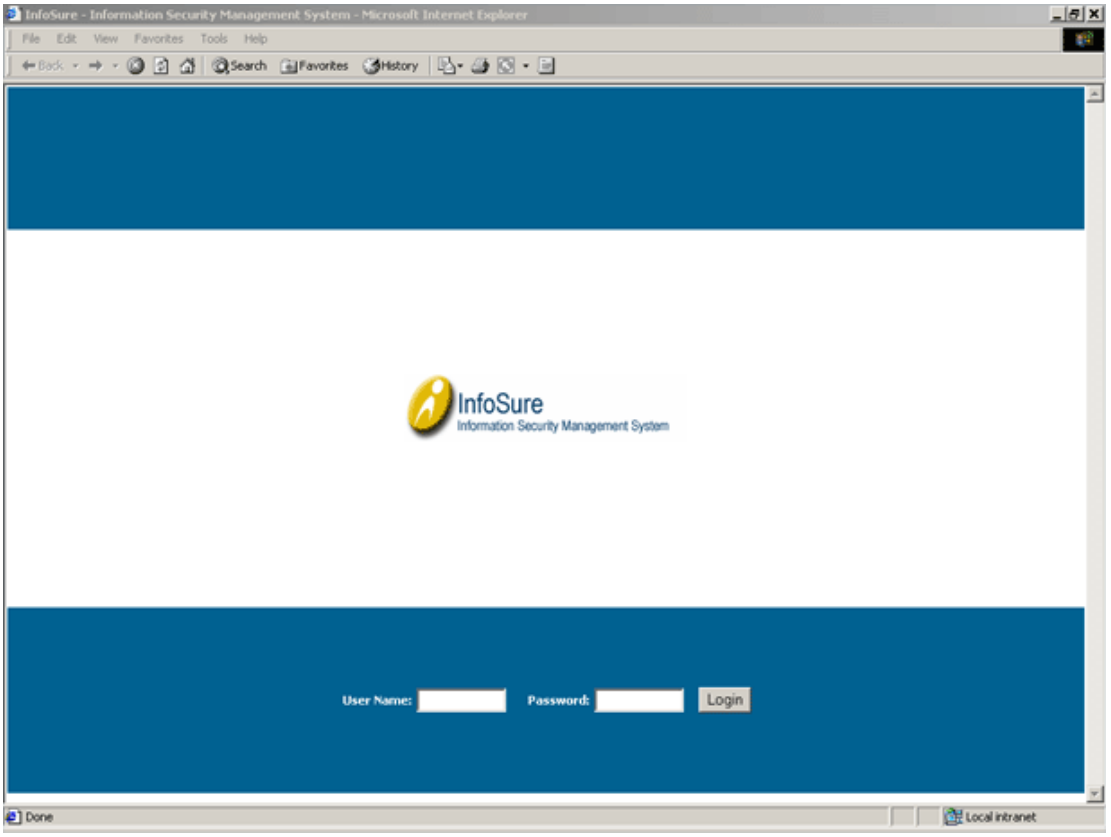


Figure 6: Login Screen

## A.1 Personal Settings

All settings specific to the user currently logged on to the system can be modified here (see figure 7). This includes passwords as well as personalisation settings. Current personalisation includes which windows of information are displayed on the administrator's ***InfoSure*** home page.

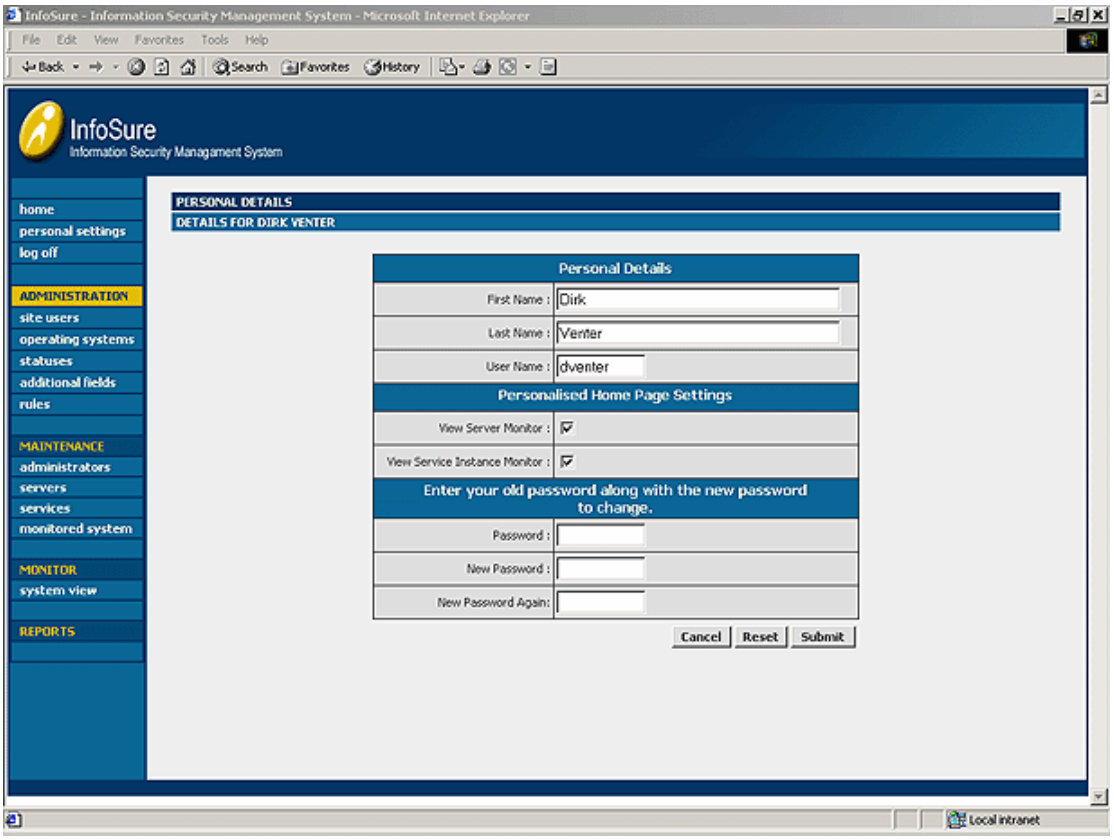


Figure 7: Personal Settings

## A.2 Administration

This section is only available to the assigned system administrator - the super user who sets up the infrastructure for the other users of the system to configure. This demonstrates the level of user configuration the system offers.

### A.2.1 Site Users

This is where the system administrator sets up and maintains other users of the system and their appropriate permissions on the site.

### A.2.2 Operating Systems

Here the administrator can set up the operating systems being used on the servers that will run the service instances. This option gives an indication of how different environments can be set up in one ***InfoSure*** implementation, allowing it to cater for non-Microsoft environments.

### A.2.3 Statuses

These are the codes that identify the specific states of the measurement services particular to the implementation and its environment. The system starts with no defined statuses; these must be set up before any other part of the system can be used. Figure 8 is a screen shot displaying some statuses that have been set up for a particular installation, as well as the colours assigned to each status.

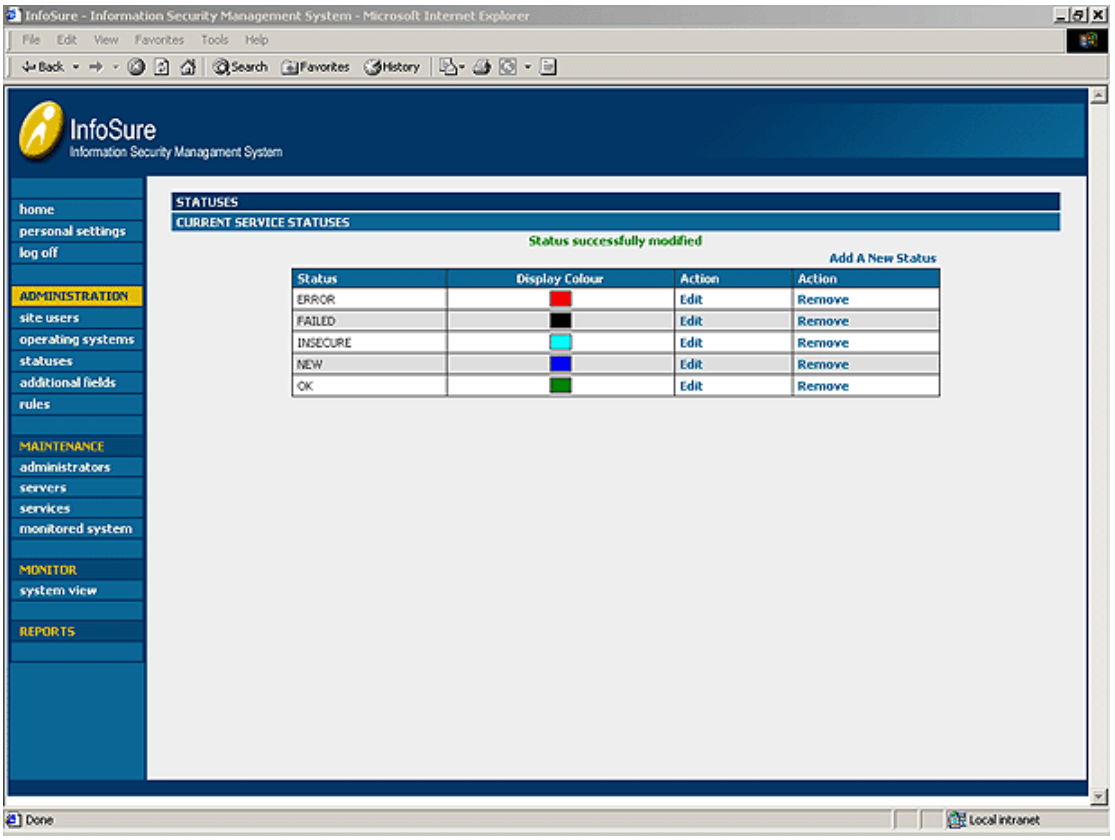


**Figure 8: Statuses**

### A.2.4 Entities

The system administrator can dynamically create new entities to be monitored. These would typically be users, software, hardware etc. The administrator has the option to create the entity type and then to add information fields (options) to it. Once the elements that comprise an entity have been defined, the administrator has the ability to associate each field option with one or more rules.

For example, if an entity of type ‘user’s’ option field ‘RAS access’ contains a ‘false’, and a status update of ‘Accessed RAS’ occurs, it is identified that the ‘user’ entity has accessed the illegally. The administrator also has the ability to add actual entity instance data via the system. This can alternatively occur via the system ‘Entity Update’ API. Status updates for monitoring purposes occur via the ‘Entity Status Update’ API, but can also be done manually by the administrator for reporting purposes. Figure 9 depicts a listing of the ‘user’ entity instances set up in the system. In this display the system administrator has selected the user id, first name and surname fields to be displayed.

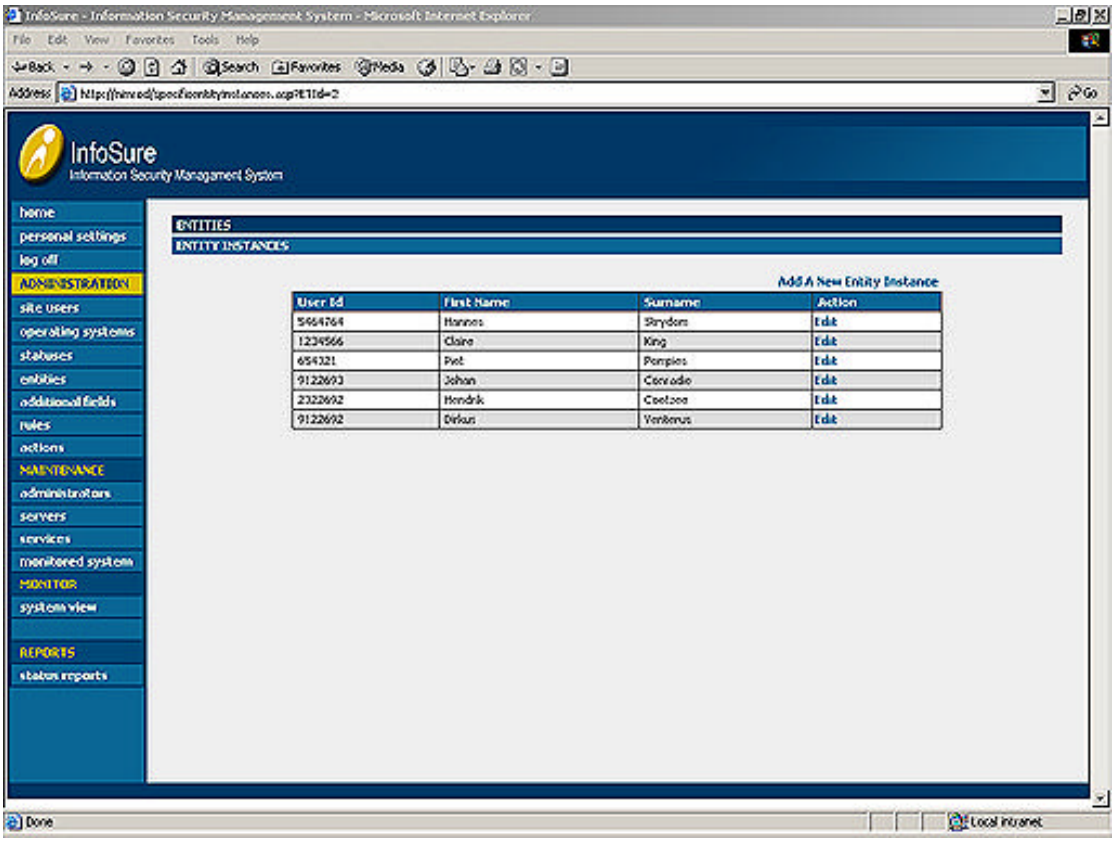


Figure 9: Entities

### A.2.5 Additional Fields

The system administrator can dynamically create additional data fields. Currently, these are only available for the following areas:

- Administrator information
- Service type instance information
- Server information

He or she can set all attributes of this field, from the type to assigning default values, thereby defining the field for the system. In figure 10 the system administrator has added 3 additional fields to the ‘easy password checker’ service type: ‘Dictionary’, ‘Last Updated’ and ‘Re-Edited’. The ‘Dictionary’ and ‘Re-Edited’ fields also have default values assigned to them.

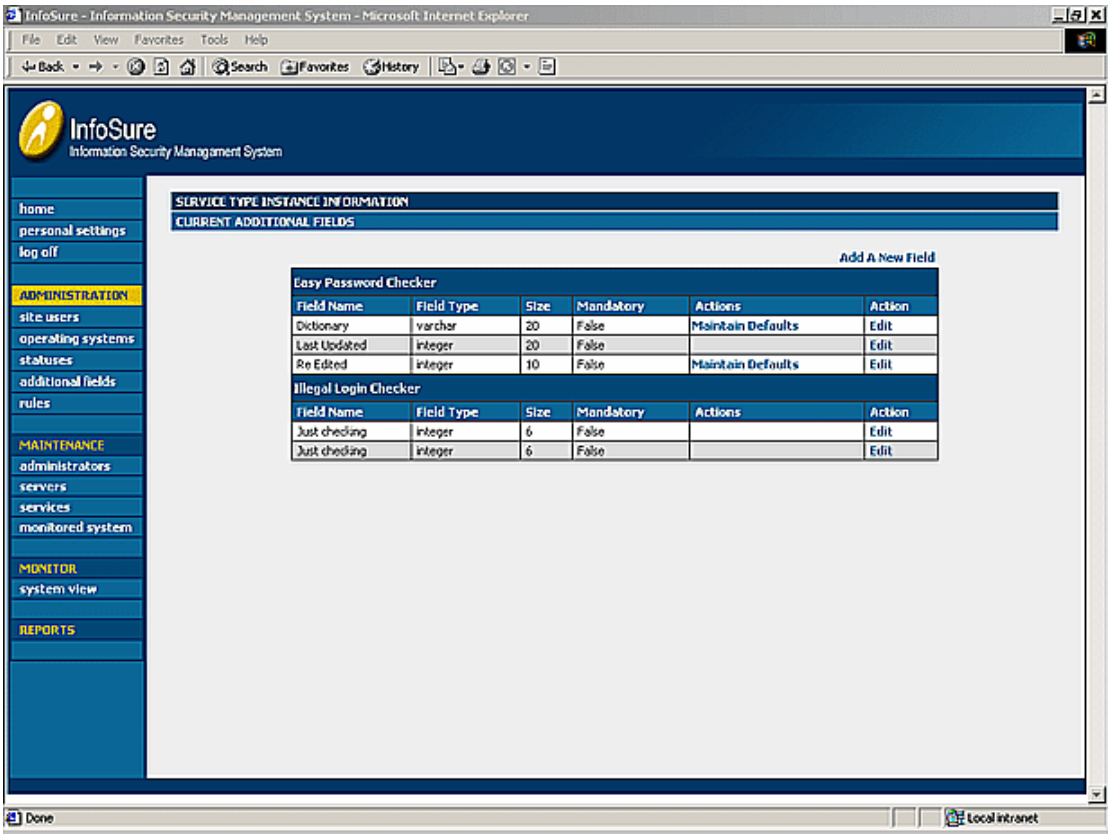


Figure 10: Dynamically Created Fields

### A.2.6 Rules

These rules should reflect the company's security policy document. A rule is defined as follows:

- A rule item is created by defining the service instance involved, the status to achieve and the number of concurrent occurrences of this status to be reached before the rule item's threshold is crossed, and a status of TRUE is assigned to it. In other words, it is assigned TRUE when the rule is broken.
- As mentioned above, one rule item on its own can become a rule element, or could be combined with another similarly created rule item, or another rule element. These are combined using one of the two Boolean logic operators AND, and OR.
- In the case of the AND operator, the rule element is assigned TRUE when both the items comprising the element has a state of TRUE, and FALSE if only one, or none, of the items have a state of TRUE.

- In the case of the OR operator, the rule element is assigned TRUE when either one of the items that comprise the element have a state of TRUE, and FALSE if all the element's items have a state of FALSE.
- Once an element is created, it can then be combined with another element or another item, again using Boolean logic, thus creating a new element.
- This results in a form of 'element hierarchy', which if TRUE at the highest level, implies that the rule was broken, and the appropriate event or events can be triggered.

Figure 11 shows a screen shot of a rule being edited. The top section displays the current rule as it currently is, and the bottom section is used to add rule items to the rule. Rules for generic entities and measurement services are defined separately.

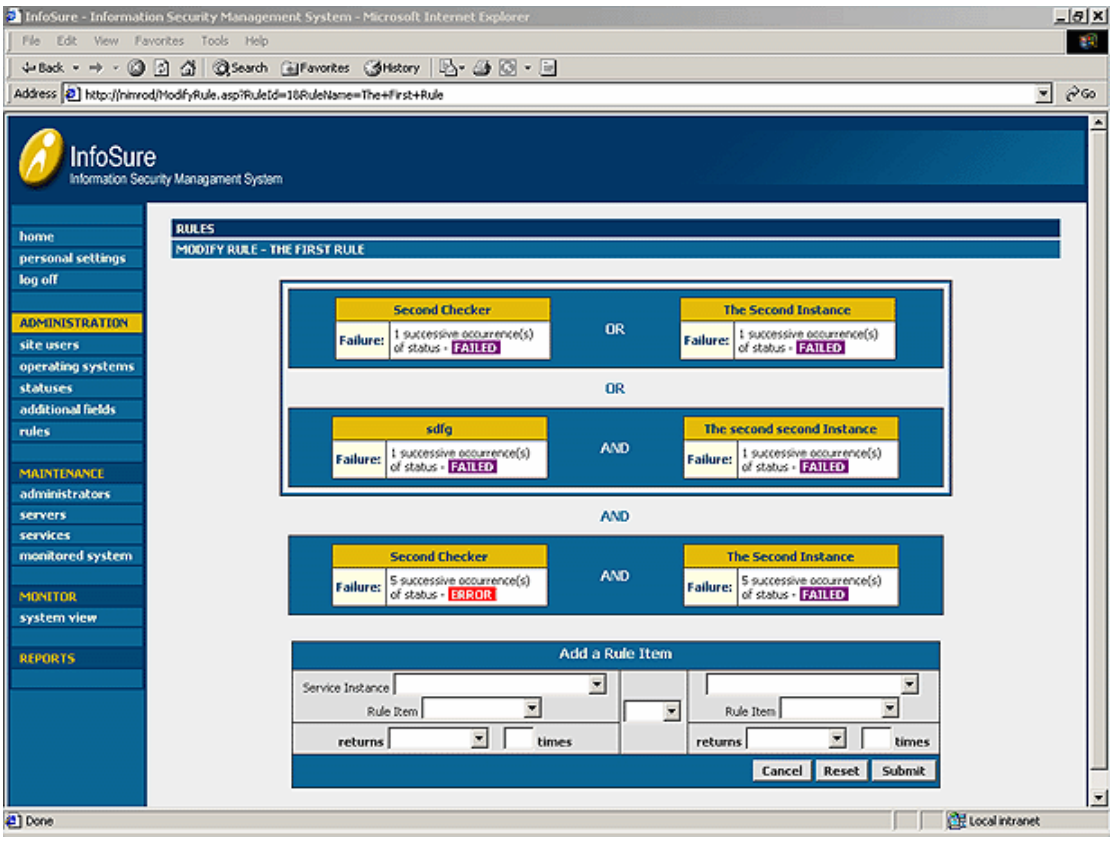


Figure 11: Rule Creation

### A.2.7 Actions

Actions can be set up and associated with the outcome of rule checks. The administrator has the ability to enter actual Visual Basic scripting code that can be executed when the action is executed. For example, script to send e-mail to the administrator can be written for execution when the action executes. The screen shot in figure 12 depicts an action containing Visual Basic script code to create and send e-mail to an administrator.

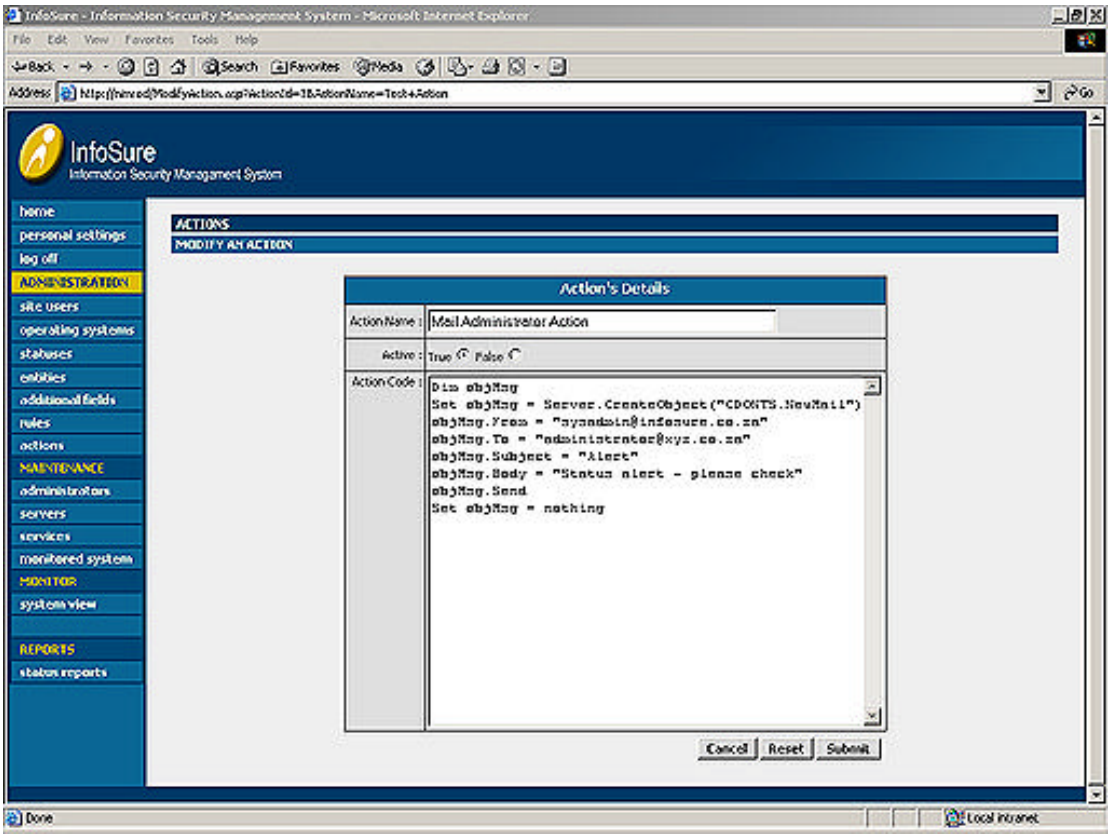


Figure 12: Action Creation

## A.3 Maintenance

The maintenance section is available to both the system administrator and other users of the system.

### A.3.1 Administrators

This section allows the user to set up server administrator details and assigns these administrators to their servers of responsibility. This role is different to that of the users of the system, although a user could potentially also be an administrator.

### A.3.2 Servers

This allows for the creation and maintenance of information regarding the specific servers on which the measurement services will run. Administrators can be assigned to the servers (see A.3.1. above)

### A.3.3 Services

In this section the actual measurement service information will be captured. The section is divided into two areas:

#### A.3.3.1 Service types

This section is used to set up the different types of measurement services that will be used in the current environment. In addition, this is where dynamic fields can be added for use as additional service instance information.

#### A.3.3.2 Service instances

Set-up of the specific measurement services, which type they belong to (e.g. ‘password checker’), identification information for the measurement modules when making use of the API, etc. Figure 13 displays a listing of measurement service instances, as set up for a particular installation.

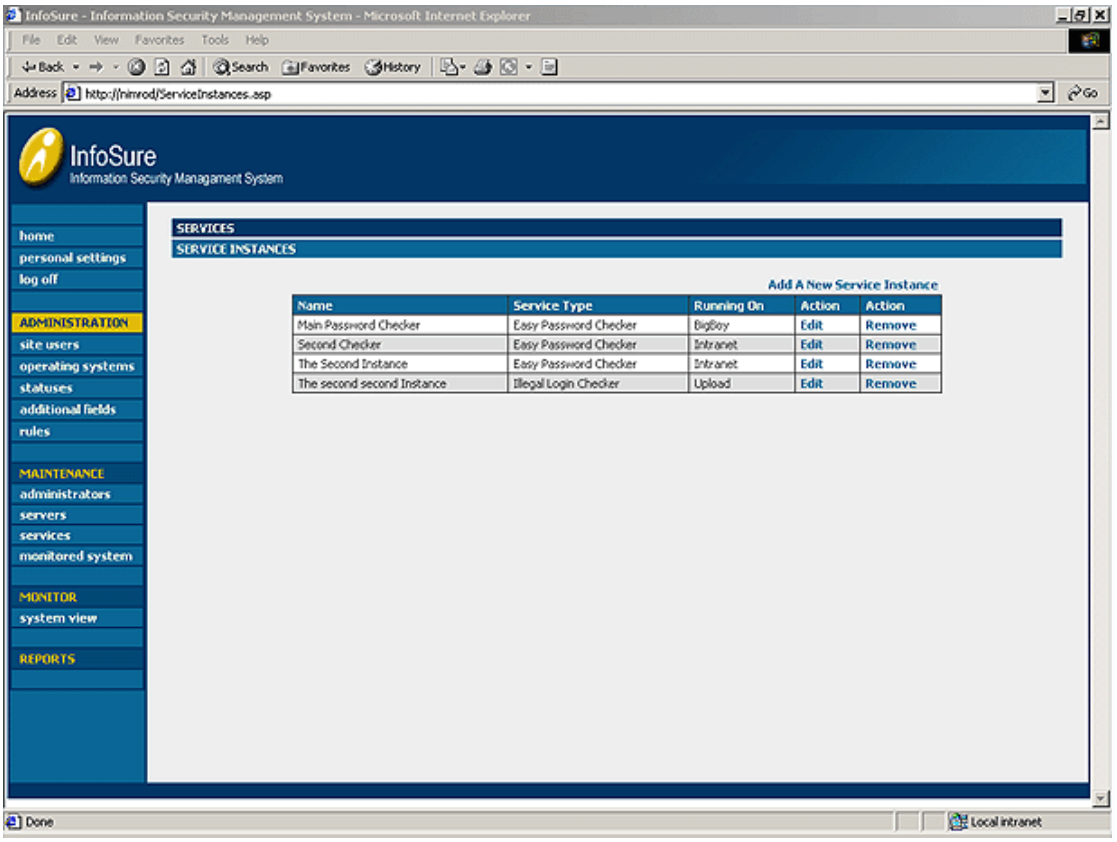


Figure 13: Measurement Services

### A.3.4 Monitored System

#### A.3.4.1 Assign Administrators to servers

Administrators can be assigned to a server, along with their responsibility level (primary, secondary etc.) and timeout information. Timeout information pertains to the period within which the assigned administrator must take action before it escalates to the next responsible person. An administrator can be assigned to more than one server.

#### A.3.4.2 Assign Administrators to service instances

Administrators can be assigned, along with their responsibility level (primary, secondary etc.) and timeout information exactly as per the

previous section. An administrator can be assigned to more than one measurement service instance.

## A.4 Monitor

### A.4.1 System View

The system view area provides an ‘at a glance’ overview of the current installation, including status history information (see figure 15).

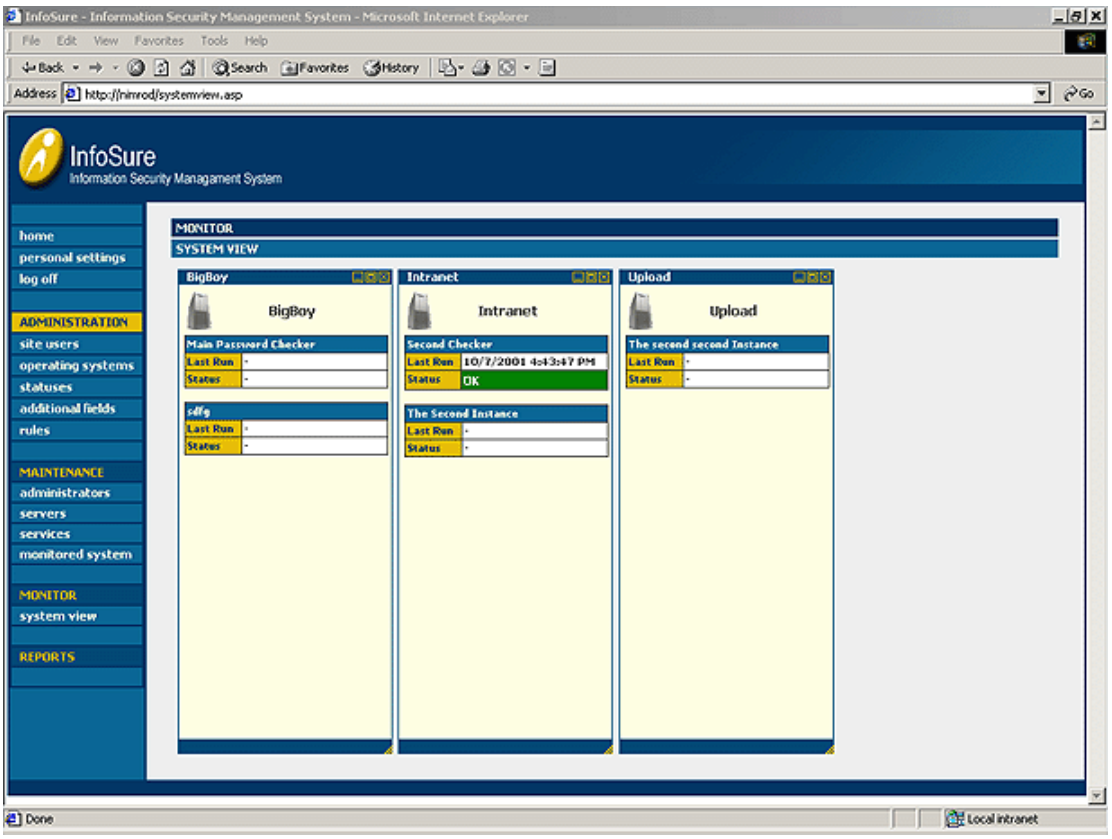


Figure 14: System View

## A.5 Reports

Various reports are available which can be used to extract measurement information as well as set-up related information.

The system dynamically creates reports based on entity set-up and rule combinations. The screen shot in figure 15 show a report of users

that have made use of illegal RAS, and when the attempt had occurred.

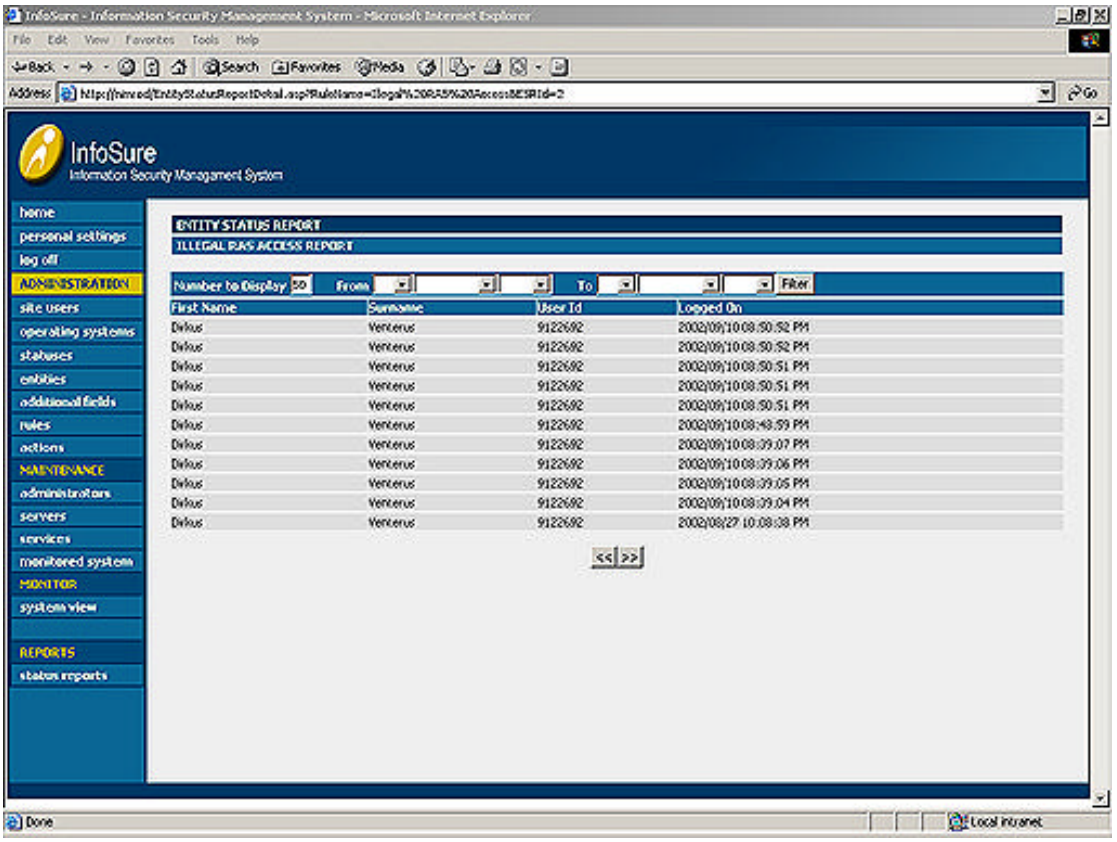


Figure 15: Example of a dynamic report

## A.6 Time consumed for development of *InfoSure*

This section presents a list of tasks and the time consumed for their achievement. Each task represents a part of the design and development of the prototype as discussed in this research paper:

- Research and conceptual design - 20 days
- Database design – 10 days
    - Developed in Microsoft SQL 2000 Server consisting of about 45 tables
- General Web site design and development– 25 days
    - Consists of about 210 ASP pages
- API design and development – 10 days
    - Consists of about 10 SQL stored procedures, 10 ASP pages and Visual Basic testing applications.
- Easy password cracker service – 30 days
    - Written as a C++ service that wraps the C code in which the password dump and password crack was written.
- 'sa' password cracker service – 5 days
    - Written as a Visual Basic windows application.
- Testing – 15 days

The total time for the whole design and development of ***InfoSure*** comes to 115 days or 4 months.

## References


[BMC 2002] BMC Software. CONTROL-SA. http://www.bmc.com, 2002

[CON 1992] Control Data Systems Inc. “Why Security Policies Fail”. http://www.cdc.com, 1999

[DHI 2001] Dhillon, G. *Information Security Management: Global Challenges in the New Millenium*. Rothstein Associates 2001

[DHI 2000] Dhillon, G and Backhouse, J. “Information system security management in the new millennium”. *Communications of the ACM*, v 43 no 7 pp 125-128.

[DHI 2000a] Dhillon, G. and Phukan, S. “Analyzing myth and reality of computer crimes”. *BITWorld*. Mexico, June 2000

[EVI 2002] Evidian. AccessMaster. http://www.evidian.com, 2002

[ESE 2002] e-Security. Open e-Security. http://www.esecurityinc.com, 2002

[ISO 17799] The South African Bureau of Standards. “ISO/IEC 17799 Code of practice for information security management”. The South African Bureau of Standards, 16 February 2001

[LUC2002] Lucent Technologies . Lucent Security Management Server. http://www.lucent.com, 2002

[MIC 2001] Microsoft Corporation. “Microsoft Windows 2000 Server Resource Kit: Supplement 1 Internet Information Services Resource Guide” http://www.microsoft.com/windows2000/techinfo/reskit/en-us/default.asp?url=/windows2000/techinfo/reskit/en-us/w2rkbook/iis.asp , 2001

[NOW 2002] Northwest Controlling Corporation Ltd. PROTEUS. http://www.noweco.com/smhe.htm, 2002

[OLI1999] Oliphant, A. “Managing Information Security – Part 1”. The Institute of Internal Auditors. http://www.theiia.org, March 1999

[PEN2001] PentaSafe Security Technologies, Inc. ViglEnt Policy Center. http://www.pentasafe.com, 2001

[PFL 1989] Pfleeger, C. *Security in Computing*. Prentice-Hall 1989.

[PIE 2001] Pieters, J. “Risk Management, Security Policy – October 2000”. *Secure IT*. December 2001

[POL2002] PoliVec. PoliVec Builder 2.0. http://www.polivec.com, 2002

[ROB2001] Robiette, A. “Developing an Information Security Policy”. Joint Information Systems Committee. http://www.jisc.ac.uk/pub01/security_policy.html, 2001

[SAB 2000] Sabin, T. “PWDUMP2”, http://razor.bindview.com/tools/desc/pwdump2_readme.html, April 2000

[SCH 2001] Schneier, B. “Managed Security Monitoring: Network Security for the 21st Century”. *Computers & Security*, v 20 2001, pp. 491-503

[SHU 2001] Shultz, G. “Taking Advantage of the Windows Scripting Host” Windows Professional. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwinpro00/html/WindowsScriptingHost.asp, April 2000

[SYS 2002] Systor. Security Administration Manager. http://systor.com, 2002

[TIV 2002] Tivoli Systems. Tivoli Policy Director. http://www.tivoli.com, 2002

[TRU 2002] Trustworks. Trusted Global Security Manager. http://www.trustworks.com, 2002

[W3C 2000] World Wide Web Consortium, “HTTP – Hypertext Transfer Protocol Overview”, http://www.w3.org/Protocols/, 2000

[W3C 2002] World Wide Web Consortium, “Extensible Markup Language (XML)”, http://www.w3.org/XML/, 2002