Mesmerize - an open framework for enterprise security management

Daniel Bradley
Securizance, Brisbane, Australia
daniel.bradley@securizance.com

Audun Josang
Distributed Systems Technology Centre*
Level 7, General Purpose South
University of Queensland
Queensland 4072 Australia
ajosang@dstc.edu.au

Abstract

We have identified five problems that inhibit effective enterprise security management - policy divide, lack of reproducibility, lack of consistency, lack of coverage and lack of flexibility in current management systems. We discuss these problems and suggest features an enterprise security management framework should have to address them.

Mesmerize is an enterprise security management framework that allows holistic enterprise security policy to be interpreted into technology specific *directives* then translated into device specific configuration.

The Mesmerize framework incorporates an *information repository*, which is accessed and interpreted by *manager programs* that - in turn - communicate with *configuration agents* that configure specific devices.

The information repository stores *network element* information as well as security policies that are associated with those network elements. Manager programs make use of the information repository to generate technology specific *directives* that are sent to configuration agents during policy enforcement. A configuration agent is responsible for translating the technology specific directive into the configuration language of a device or service implementation.

Currently we have proof-of-concept management sub-systems for IPChains firewalls (IPChains), BIND domain name servers (BIND), and FreeSWAN virtual private network end-points (FreeSWAN).

* The work report in this paper has been funded in part by the Co-operative Research Centre for Enterprise Distributed Systems Technology (DSTC) through the Australian Federal Government's CRC Programme (Department of Industry, Science & Resources).

Keywords: enterprise security management, policy, network security, expert system.

1 Introduction

The enterprise security management cycle can be divided into six stages: threat and risk assessment, business process analysis, security policy definition, security policy interpretation, security policy enforcement and security audit and monitoring.

The threat and risk assessment identifies an organization's information assets, while the business process analysis identifies staff roles that need access to those assets. Using this information an organization's security policy can be defined. Effective security management relies on the subsequent interpretation of this policy by the technical staff responsible for enforcing the policy. Auditing and monitoring is required to ensure that the policy remains enforced.

We have identified five problems that inhibit effective enterprise security management. First, there exists a policy divide between policy makers and policy enforcers. Second, security management isn't reproducible. Third, consistency of configurations cannot be easily assured. Fourth, networks are not configured as securely as is technically possible. Lastly, current security management systems are not flexible enough to be used in heterogeneous networks, and are also too expensive for small to medium enterprises (SMEs).

This paper presents Mesmerize, an open framework for enterprise security management. Its purpose is to increase the effectiveness of the enterprise security management cycle by overcoming the problems mentioned above.

Section 2 discusses the stated problems in further detail. Section 3 suggests features required of an enterprise security management framework attempting to solve those problems. Section 4 introduces the Mesmerize enterprise security management framework. Section 5 describes our current implementation. Section 6 discusses related work. Section 7 describes future work.

2 Problems in enterprise security management

Effective enterprise security management is currently hampered by the following five problems.

2.1 Problem 1: the policy divide

Computerized enterprises face the problem that there is a divide between the formation of enterprise security policy and its enforcement. This divide is two-fold: first there exists the divide between the intent of managers and the interpretation of policy by technical staff, second there is the translation of this interpretation into the configuration languages of the many different devices and services found in today's organizations.

The policy divide between management and technical staff is exacerbated by the rate of introduction of new technologies. Vendors now offer security products as "solutions", whereas actually they perform a single security enforcement function.

2.2 Problem 2: reproducibility

Security management is currently a black art. If a technician that had previously configured a device were required to reconfigure the same device a week later, the probability that they would be able to reproduce the exact same configuration would be small. Now imagine if that staff member has left the organization and another person had to reconfigure the device in one years time.

The arcane syntax and semantics of different configuration languages guarantee that unless specific security measures are thoroughly documented they are incomprehensible to new staff. This problem has lead to the use of software that has the simplest management consoles, and a tendency to ignore the active configuration of services and devices until they "break" or they are implicated in a security incident - an "if it ain't broke don't fix it" attitude.

2.3 Problem 3: consistency cannot be ensured

A side effect of non-reproducible configurations is that it is very hard to maintain consistency between the configurations of devices in a particular technology domain (for example, firewalls from different vendors), let alone the consistency of configurations between devices in different technology domains. Security relevant devices are currently configured in isolation from each other, meaning that consistency of effective security can be compromised.

Current best practice is to follow strict guidelines when performing configuration. However, organizations typically do not have the technical and security expertise required to develop such guidelines. Or, when developed, often the guidelines fall quickly out of date.

Also guidelines only allow an organization to perform configuration as needed. To check that configurations are correct an expensive audit must be undertaken. This means that system administrators cannot see the true configuration state of their networks, and therefore do not know the real security state.

2.4 Problem 4: coverage

Computer networks are not as secure as they potentially could be. This is due to the amount of effort it would require to initially configure every device in the network, and then the extra maintenance effort to maintain those many configurations.

2.5 Problem 5: current systems are proprietary and inflexible

Management systems currently exist that claim varying degrees of enterprise security management (Computer Associates, Hewlett Packard, Tivoli). These systems, however, were originally designed for enterprise systems management. As information security has become more prominent (and lucrative) security management functionality has been added, but tends to be constrained by the software's pre-existing management paradigm. These have tended to rely heavily on strictly defined management protocols such as SNMP, CMIP, and WBEM.

These and newer systems (Cabletron) are constrained by their proprietary nature as it is difficult for third parties to extend the systems to better support heterogeneous networks. Also due to expensive license fees and support contracts, these systems are not affordable for small to medium enterprises (SMEs).

3 Features of an enterprise security management framework

A new enterprise security management platform should endeavour to reduce or eliminate the problems discussed.

3.1 Policy languages

To reduce the effect of problem one - the policy divide - enterprise security policies must be able to be specified in a standard policy language or representation. This must represent business requirements related to the five security principles of *authentication*, *confidentiality*, *integrity*, *availability* and *non-repudiation*, while remaining technology neutral.

Policies should specify the security requirements of the organization's information assets and business processes, not how those requirements should be enforced. The policy representation also needs to be intuitive and easily understood by both management and technical staff.

The other half of this problem is the interpretation of policies by technical staff into the terms of different technologies and the consequent translation into the configuration language of a device or service. An intermediate language is needed that specifies the intent of policies in terms of a specific technology. This language could then be mapped to specific configuration languages from different vendors.

3.2 Reproducibility

The second problem – lack of reproducibility – is due mankind's inability to perform the same task twice in exactly the same manner. We (humans) tend to unpredictability while machines are better suited to tasks that require reproducible results.

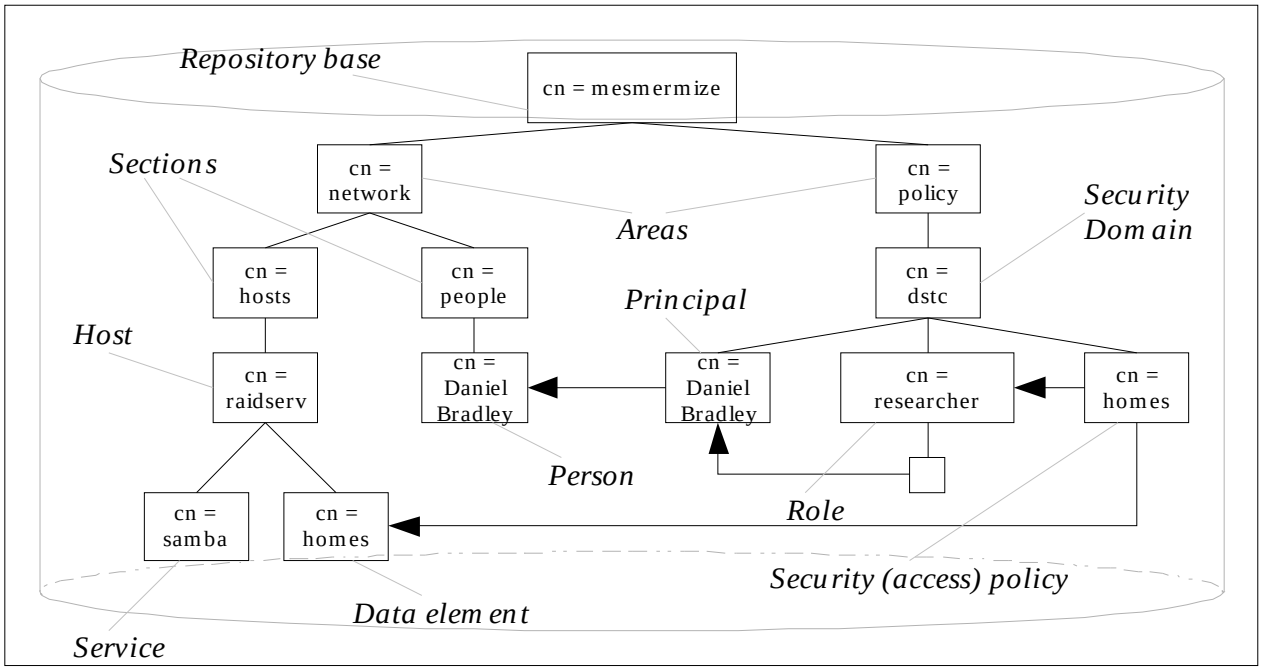

Figure 1 - A simple example of objects within a Mesmerize information repository

Therefore an enterprise security management framework should act as a form of expert system, providing programs that interpret the collective security policy then generate a specific device configuration using domain knowledge. These programs could be maintained by domain specialists - allowing networks to be always configured according to current best practice.

3.3 Consistency

As well as achieving reproducibility, the use of expert programs make it possible to achieve configuration consistency between similar technologies (for example, packet filters from different vendors) and dissimilar technologies (for example, between packet filtering rules and access control permissions).

As all expert systems would interpret the same security policy language and semantics, the development of the expert systems of complimenting technologies could be reliably tested together, this would allow true consistency within a network.

3.4 Coverage

Automation of enforcement of policy using expert programs would also empower organizations with a width of security configuration coverage that would previously have been impractical, as it should be possible to implement expert programs for any type of technology.

3.5 Flexibility and openness

The framework itself should be fully open and well documented. Third parties should be able to easily audit the security of the system as well as develop expert system modules. Openness would encourage adoption of the framework, which would further encourage vendors to develop for the framework.

The framework should also be simple to deploy and maintain, lowering the total computing cost of ownership and not requiring a large dollar service contract.

4 The Mesmerize framework

The Mesmerize framework is our attempt to provide a cohesive enterprise security management solution. An *information repository*, which stores network and security policy information; and technology-domain-specific *manager programs* with associated *configuration agents* that provide the required expert system functionality; are its main components. Additional common infrastructure services also help initiate and facilitate policy enforcement.

4.1 The information repository

The information repository is an object-oriented database that is responsible for maintaining a holistic view of an organization's computer network. The purpose of the repository is to allow security policies to be expressed in terms of real security requirements rather than in terms of technologies or device configurations. The objects stored within the database are separated into two areas corresponding to network elements and security policies, respectively (see figure 1).

In the first area each object stores information about a particular network element eg. subnet, computer, device etc. An extensible schema allows any conceivable object to be represented within the repository. Network elements may contain other elements; therefore a computer object might contain service objects representing the services that run upon it. Objects may also have associations with other objects, thereby allowing many-to-many relationships to be represented, such as domain and subnet membership, or what hub/switch a computer is connected to.

In practice, the network element area of the repository should contain enough information to allow the duplication of a network's functionality and topology at another site. Clearly this information could be used to set up NIS, Net Info or any other network information service; indeed this will be a necessary function of the Mesmerize framework in the future as many security mechanisms depend upon the correct configuration of information infrastructure. In other words, the repository describes what technicians would "need to know" before they could set up a network, before any configuration "tweaking" is performed, and any security policy is applied.

The second area of the repository describes a security domain hierarchy similar to that described in the CORBA security services specification (CorbaSec). Principal objects are used to place network elements such as people or hosts within specific domains. They are also used to assign principals into groups or roles.

Within the Mesmerize framework the domain hierarchy implicitly implies the following. Objects within the same domain can communicate with one another. Objects within a sub-domain can communicate with objects in their parent domain (though those objects may not be able to communicate a response). Objects cannot communicate with objects that are in sub-domains below their domain. An object that is not a member of a domain is treated as if it is the sole member of a root level domain.

Security policy objects may then be used to change these default rules. Each policy object specifies a target, an agent and a policy. The target must be a network element; if the target has child objects, those objects are implicitly also targets of the policy; if the target has parent objects, any interactions with those parents are also implicitly allowed. The agent represents a principal who is covered by the policy; if the principal is a composite principal (such as a group) then each of those principals is also covered by the policy. The policy specifies the action that should occur should a covered principal attempt to interact with a target. The actual policy will depend upon the policy type and the type of the target network element.

4.2 Manager programs

Manager programs act as technology-specific expert systems. They interrogate the repository for information related to the network elements that they are responsible for managing, then determine what security polices affect those elements. Using this information they produce technology-domain-specific *directives* that are then sent to the configuration agent of each element.

Directives are technology-specific interpretations of policy, thus a different manager program is required for each technology domain (for example; packet filter, print server). The core framework does not specify the protocols used to describe and transmit directives - we thought it more appropriate that these be developed by consensus within individual technology domains.

Typically manager programs are non-interactive, being normally instantiated (run) by a launcher program provided by the framework.

As an example, it is the responsibility of the firewall manager program to facilitate the configuration of each firewall in the network. To do this it interrogates the repository to determine which network elements represent firewall services and how they are related topographically. It then determines what services exist and what security polices involve them ie. who are allowed access to those machines. The manager program then uses that information to produce directives for each of the firewalls that specify what connections each firewall is to allow. These directives are then delivered to the appropriate configuration agents.

4.3 Configuration agents

Configuration agents are responsible for translating directives received from manager programs into appropriate configurations for the specific device they support.

Configuration agents normally run on the device that requires configuration, but in some cases may act as an intermediate (proxy) that is responsible for applying the configuration using a different management protocol, e.g. SNMP, for the target device.

5 Implementation

Our implementation consists of open source software and custom written libraries and programs. Currently the information repository and manager programs run on a single bastion host referred to as the *Mesmerize bastion*.

5.1 Information repository and tools

To provide an open basis for future development the information repository has been implemented using Open LDAP. The only additional requirement is a custom LDAP schema that is available from us.

We have also developed an auto-population tool that can monitor a network and automatically populate the repository with network elements that are detected. At the moment this is limited to host information (name and domain) and the services that they are running.

So far we have developed two repository management applications. The earliest, written in Java, manages the repository directly using LDAP. The second, written in C++, is limited in functionality but displays the repository information as a 3D landscape.

5.2 Mesmerize libraries

Object-oriented C++ libraries have been developed to allow easy manipulation and searching of the repository by manager programs. These libraries provide an abstraction layer that hides the LDAP back-end.

5.3 Common infrastructure

A problem is "what or who determines when a service or device should accept a directive from a manager program?" For those services that are currently managed via a remote management protocol it might be acceptable to have a configuration agent listen on an open port, provided an equivalent or higher level of security was available. However, security critical devices, such as firewall components, should have as few open network ports as possible.

To cater for sensitive security services, infrastructure services have been developed that allow devices to initiate reconfiguration themselves. The device may send a directive request that specifies the port the manager program is to connect to, and may additionally include a one-time secret that the manager program must also present when it sends the directive.

The *directive request service*, which runs on the Mesmerize bastion, receives the request and queues it to another service - the launcher - that executes an appropriate manager program for the device. Interactive management applications may also make use of this service to request that specific devices (or groups of devices) be configured.

5.4 Manager programs

At time of writing we have a functioning Firewall manager program that communicates with an IP Chains configuration agent. By time of publishing we expect to have DNS and VPN manager programs finished that are currently at the proof-of-concept stage.

5.5 Security Considerations

A consequence of using the Mesmerize framework is that it becomes a potential single point of failure. A compromise of the information repository would amount to a compromise of the entire network. To address these concerns the Mesmerize framework has been designed from the ground up with real security in mind (not just encryption). The following risks have been identified.

5.5.1 Compromise of Mesmerize Bastion

Apart from physical compromise, there are two attack paths that may be used to compromise the mesmerize bastion - both of these involve network communication with other hosts.

The first attack is the exploitation of a vulnerable service running on the bastion. To reduce the chance of a service being compromised the Mesmerize bastion only runs one service that listens to an external network interface - the directive request service. To reduce the likelihood of a successful buffer overflow attack, this service strictly interprets and rewrites the packets it receives before re-communicating them to the application manager launcher. To reduce the consequences of a successful buffer-overflow attack the service also runs with reduced rights.

The second attack path involves the exploitation of a vulnerable local program that has connected to a malicious remote service. To reduce this likelihood the only programs that are permitted to communicate to other hosts are the manager programs (DNS requests are handled locally).

To protect against subversion, manager programs run with reduce rights. These managers must only ever have one connection open at any time. Initially a connection is established with the information repository to read policy. This connection is established using one-time application-level credentials that are used once then destroyed before further connections are made to configuration agents. When the connection to a configuration agent is made the application manager should have no rights. While the directive is communicated to the configuration agent incoming communication must not be read.

5.5.2 Subversion of manager-agent communication

It is possible that the connection between a manager program and a configuration agent could be subverted. An intermediate host could perform a man-in-the-middle attack by intercepting a directive request, thereby getting access to the one-time-secret, allowing modification, or substitution, of the intended response. Attacks such as this are well documented, and the best current form of protection is the use of cryptography.

Note encryption is not mandated for the connection between a manager application and a configuration agent. Therefore, it might be possible for an attacker to hijack these unencrypted connections.

5.5.3 Deployment specific measures

We intend that this framework will be used in a variety of settings from home up to large enterprise. In home deployments it might be desirable to co-locate the Mesmerize repository with other services such as print and file sharing. However, for any organizational setting we recommend that the repository be run as a true bastion host. For larger installations the Mesmerize Bastion might also perform packet filtering, and might also sit behind another firewall.

6 Related Work

Related work in the area of enterprise security management is spread over many fields including autonomous and mobile agents, policy languages, business process analysis, remote configuration protocols, access control systems and systems management. Therefore below we only mention work that we consider directly relevant.

(Roeckle 2000) discusses the business process analysis to determine roles for RBAC. Unicenter TNG also boasts the use of business process analysis to manage systems. (Computer Associates).

In (Moffet 1990) domain hierarchies are discussed. Also usage of security domains is presented in the CORBA Security Services Specification (CORBASec).

There has been a substantial amount of work on policy languages, (Damianou 2002) provides a review of policy specification approaches. In particular work at Imperial College has produced the Ponder policy language, which specifies an "obligation policy" that is similar in concept to a Mesmerize "directive".

Policy servers are available that make available this "directive-type" of policy to clients (Tivoli).

Most work in the policy area related to access control is heavily tied to role-based access control (RBAC). Early papers describe (Ferraiolo 1992) and extend (Sandhu 1995) the various models of RBAC. Later papers describe implementations that use RBAC for access control management (Awischus 1997), (Barkley 1998), (Ferraiolo 1999).

(Bellovin 1999) proposes the use of distributed firewalls, which distribute packet filtering responsibilities among the hosts on a network rather than locating it in a single host on the network border. This is achieved by distributing tailored policies to each host. In (Fuller 1999) the use of expert systems for security management is proposed.

The Napoleon system (Thomsen 1999) shares the concept of a high level policy layer that is interpreted into lower level policies, but is targeted more at application level access control mechanisms.

7 Conclusion

The aim of the Mesmerize framework is to turn the black art of enterprise security management into a reproducible, automatable science. The core of the framework is complete and now just waits for more manager programs and configuration agents to be implemented. We hope that in the future the Mesmerize framework will be supported as an open standard for ensuring the security of computer networks.

8 References

Awischus, R. (1997): Role-based access control with the security administration manager (SAM). *Proceedings of the 2nd ACM Workshop on Role-Based Access Control*, Fairfax, Virginia, USA, pp. 61-68, ACM Press.

Barkley, J., Cincotta, A. (1998): Managing role/permission relationships using object access types. *Proceedings of the 3rd ACM Workshop on Role-based Access Control*, Fairfax, Virginia, USA, pp. 73-80, ACM Press.

Bellovin, S.M. (1999): Distributed firewalls. *Usenix login.*

BIND: Berkeley Internet Name Domain, ISC. http://www.isc.org/products/BIND/. Accessed 18 Sep 2003.

Computer Associates: Unicenter TNG, Computer Associates International Inc. http://ca.com. Access 18 Sep 2003.

Damianou, N, Bandara, A.K., Sloman, M. and Lupu, E.C. (2002): A survey of policy specification approaches. citeseer.nj.nec.com/damianou02survey.html

Ferraiolo, D. and Kuhn, R. (1992): Role-based access control. *In 15th NIST-NCSC National Computer Security Conference*, pp.554-563, Baltimore, MD.

FreeSWAN: Linux FreeS/WAN. http://www.freeswan.org. Accessed 18 Sep 2003.

Fuller W. (1999): Network management using expert diagnostics. *International Journal of Network Management*, 9:199-208.

HP: Open View. Hewlett Packard Company. http://www.hp.com. Access 18 Sep 2003.

IPChains: Linux IP Firewalling Chains. http://www.netfilter.org. Accessed 18 Sep 2003.

Moffet J.D. and Sloman, M.S. (1991): Delegation of authority. *In Integrated Network Management II*, North Holland (April 1991) pp. 595-606.

Roekle H. Schimpf, G. and Weidinger R. (2000): Process-oriented approach for role-finding to implement role-based security administration in a large industrial organization. *Proceedings of the 5th ACM Workshop on Role-Based Access Control*, Berlin, Germany, pp. 103-110, ACM Press.

Sandhu R.S. Coyne, E.J., Feinstein, H.L. and Youman, C.E. (1996): Role-based access control models. *IEEE Computer* 29(2): 38-47, IEEE Press.

Tivoli: Tivoli, IBM Corporation. http://www-3.ibm.com/software/tivoli/. Accessed 18 Sep 2003.

Thomsen, D., O'Brien, C. and Payne, C. (1999): Napoleon: network application policy environment. *Proceedings of the 4th ACM Workshop on Role-Based Access Control*, Fairfax, Virginia, USA, pp. 145-152, ACM Press.