

Draft Proposed GNU Enterprise Security Framework

Stanley A. Klein

1. Introduction and Overview

This document is intended as a framework for GNU Enterprise security. The document is based on the following principles:

1. There are two major purposes of GNUe security:
 - a. To ensure that GNUe provides the requisite security functionality to enable its users to satisfy their the legal obligations regarding issues such as trustworthiness of data, auditability of financial records, and protection of personal privacy, and
 - b. To enable GNUe users to implement such policies as they may reasonably adopt for protecting the confidentiality, integrity, and availability of their valuable data and business processes.

2. GNUe is intended to work with a variety of supporting environments. These environments include security features and functions provided by
 - a. The operating system and associated security-related tools
 - b. The database management system (DBMS) or other data management function
 - c. The middleware, or other support for interoperability and distributed processing
 - d. The functions provided by GNUe itself, and
 - e. Physical protections and manual/paperwork procedures.

3. GNUe is intended to operate in a variety of architectures, ranging from two-tier (client-server) architectures implemented on a single-user machine to multi-tier architectures implemented on distributed processing networks, possibly including Internet connectivity.
4. To the maximum extent feasible, GNUe will provide its security functions by facilitating use of the security services and security policy implementation capabilities provided by the operating system, DBMS, and middleware standards. If a potential user security policy is identified that can not be met using these features, it will be recommended that the policy, if adopted, be met using paperwork and manual procedures.
5. The security functionality of GNUe will be focused on support for Role Based Access Control (RBAC), which essentially implements the traditional methods used in business for protecting the confidentiality, integrity, and availability of valuable data and business processes, and is compatible with the leading trends in information security.
6. The approach used for GNUe security will be to provide the appropriate infrastructure, together with user guidelines, necessary to enable users to meet legal obligations and to implement reasonable security policies under a range of anticipated environments. The detailed security requirements will be determined by the enterprises that use GNUe. The guidelines will be organized into a set of security levels, appropriate to the anticipated needs of various categories of users.

1.1. An overview of security

The underlying concept of GNUe security is that the user has the basic responsibility for security and that GNUe can only provide appropriate tools within its scope to support the user in implementing security.

The basic goals of information security are to protect information from disclosure to unauthorized recipients (confidentiality), to ensure that only authorized sources make changes to information (integrity), to ensure that the legitimate users of a system can receive its services when required (availability), to ensure that actions of an individual can be uniquely traced to that individual (accountability) and to ensure that agreements made electronically can be proven to have been made (non-repudiation).

Information security is always a tradeoff. Perfect security does not exist in the real world. Any real world system can be successfully attacked if the attacker is willing to spend enough time and money. The defender's objective is to make the time and cost of a successful attack much greater than either the useful life or value of the information

in the system. There is also a tradeoff among the potential losses to the owner if information is successfully attacked, the costs of protection, and the inconvenience to legitimate users caused by the protective measures.

Within the scope of security the tasks include providing protection, detecting intrusions, and recovering from intrusions. In some cases, there is a tradeoff between preventing an access and allowing the access but logging its activity. In such cases improper activity is detected by evaluating the logs.

In general, an enterprise should identify and inventory the data it expects to manage using GNUe, perform the relevant tradeoffs, and establish security policies and associated protections for each category of data it identifies. It should then configure its implementation of GNUe to satisfy the policies it adopts.

Recovery from an intrusion involves treating the computer system as a crime scene. Just as with a physical crime scene, the system must be secured against disturbance and evidence must be collected in a manner that is appropriate under applicable law. System recovery can be started only after the evidence has been collected and secured.

1.2. Outline of the document

The remainder of this document presents the details in the areas of assumed environments, RBAC, and the range of security policies that will be supported.

2. Security categories

The following sections describe a categorization of potential GNUe users and the threats they may regard as relevant.

2.1. Category A: Very small company, all users fully trusted for all functions, no legal or contractual constraints

This could be a single-user, home based business running GNUe in two-tier mode on a single processor system or a 5-person office where all users are highly-trusted partners in the business.

The principal threat of concern to such an enterprise is:

- Inadvertent error

The system can be expected to be used for connecting to the Internet, although not for GNUe functionality. The user must be expected to provide appropriate protection to prevent intruders from attacking the system by the various common means of Internet-based attack, such as by exploiting vulnerabilities of actively-connected systems or by inducing the user to accept malicious software such as viruses or Trojan Horses.

2.2. Category B: Very small company, all users fully trusted for all functions; legal or contractual constraints

This is the same kind of enterprise as in (A), but there are legal and/or contractual constraints that apply to the accounting or other recordkeeping systems. One example of such a constraint is a legal trustworthiness requirement such as the Province of Quebec statute prohibiting accounting systems from being capable of deleting or modifying transactions after they have been posted. Another example is the requirement for accounting system certification established for certain US Government contractors under applicable Federal Acquisition Regulations.

2.3. Category C: Small/medium company, legal and contractual requirements, no external network connection

This could be a company of small-to-medium size that does its accounting and other corporate administration in a headquarters with air-gap security from external intrusion, i.e., no Internet connection and no dialups to the GNUe system.

The principal threats of concern to such an enterprise are:

- Inadvertent error
- Embezzlement and employee fraud
- Disgruntled employees
- Violation of legal/contractual/business-policy requirements for non-disclosure or integrity-protection of certain data (such as employee records, proprietary data, and accounting audit trails).

2.4. Category D: Small/medium company, legal and contractual requirements, external network connection

This is the same kind of company as in (D) with Internet or other external network connection to the GNUe system.

The principal threats to such an enterprise include:

- Inadvertent error
- Embezzlement and employee fraud
- Disgruntled employees
- Violation of legal/contractual/business-policy requirements for non-disclosure or integrity-protection of certain data (such as employee records, proprietary data, and accounting audit trails).
- Common business threats from external intruders

2.5. Category E: Medium to Large company with special concerns

This could be a company of any size, but most likely a multi-divisional corporation having additional special concerns affecting security. The kinds of concerns than may arise include:

- Critical infrastructure protection concerns (such as US policies affecting financial entities, utilities, health care, transportation, emergency services, and other selected industries)
- Trans-border data flow and personal privacy regulations of the European Community

3. Discussion of Security Policy Drivers

3.1. Legal trustworthiness requirements

A legal trustworthiness requirement generally arises out of the need for trustworthy

recordkeeping for official processes such as stockholder reporting, fiduciary reporting, tax determination and auditing, and use as legal evidence in trials. In all these cases there is a need for assurance that the records in question are the complete and actual records they purport to be and have not been altered since they were finalized.

As a general rule in computer systems, if it is desired for data to be unalterable, the data must be recorded on unalterable media. Regardless of how secure the operating system may be, an appropriate group of people having sufficient privileges on the system can do anything they want to do with any of the data on the system. This is an important reason why, in the case of an information security breach, it is usually required that all relevant data (such as the contents of the hard drive) be immediately written to CD-Recordable media, sealed and labeled as appropriate for evidence, and placed in a legally-compliant chain-of-custody.

While criteria for the trustworthiness of legal evidence (and even for some analog-stored electronic evidence) have been long established, the equivalent criteria for digitally-stored and computer-based evidence are in their infancy. The issues are being addressed in evolving policies, legislation, and court cases related to e-commerce, management of digitally-stored government records, and other areas. Often, the legal system is found to be seriously out-of-touch with the world of technology and requires extensive education (and sometimes the development of new institutional approaches) for dealing with underlying technical issues.

One recent example of a legal trustworthiness requirement is the Province of Quebec accounting auditability statute. This statute was adopted after a situation in which it was reported that some Quebec restauranteurs were using a “zapper” program to delete sales records in point-of-sale terminals, thereby reducing their reported tax liabilities by significant percentages. After investigation, Quebec adopted a statute making it illegal for any accounting-related computer program to be capable of deleting posted data, and providing severe penalties for the use of such programs.

3.2. Critical Infrastructure Protection requirements

Critical infrastructure is a term for facilities, including their information technology and communications capabilities, that are deemed essential for public health, safety, or critical economic reasons. Examples of critical infrastructure facilities are banking, electric/gas/water utilities, transportation, health care, and emergency services. The assumed threat comes from highly sophisticated potential attackers, such as organized crime, international terrorists, or other international adversaries. These potential attackers are assumed to have goals that may transcend the specific nature of the facilities threatened, and may be willing to spend large sums of money to achieve their

goals. Accordingly, the protections that may be required are likely to go well beyond the kinds of protections needed to defend against lesser threats.

3.3. Other legal requirements

There are numerous other legal requirements that may affect the security policies of an organization. Examples include:

- Legally-mandated privacy policies
- Insider trading rules - that govern the scope and timing of financial and other disclosures of information that could affect publicly traded stocks, bonds, or other financial securities.
- Standards-of-conduct, such as those that were mandated as part of electric power restructuring in the US. The restructuring separated the functions of an electric utility into regulated and unregulated parts. Under the Standards of Conduct, information in the regulated parts of an electric utility can not be disclosed to personnel working in the unregulated parts of the same utility.

3.4. Separation-of-duty business policies

Separation-of-duty is the standard method used by organizations -- both business and government -- for ensuring the integrity of their business processes. In general, separation-of-duty policies are an ordinary part of the policies adopted under organizational governance. These policies are commonly published in a policy manual distributed to relevant managers and employees within the organization. Examples of separation-of-duty policies include:

- The common requirement for two signatures on checks
- Requirements for approval or counter-signature of expense vouchers, even if the submitter is a member of senior management
- Requirements for multiple signatures (customer, originator, supervisor) on refund vouchers in retail businesses
- Requirements for certain small disbursements to be made by check rather than cash (where the check is prepared by a different part of the organization than the one requesting the disbursement)

- The use of multiple keys or combinations (intended to be entered by separate people) for opening safes or vaults
- Requirements that purchases be made by a purchasing department based on purchase requests initiated outside the purchasing department

Separation-of-duty policies may depend on the size and scope of transactions or on other factors as determined by the governing body of the organization. For example, certain people may be authorized to commit the organization at certain monetary values and risk levels, but increasing monetary value or risk may require higher level approval.

4. RBAC

Role Based Access Control essentially implements the separation-of-duty approach that has long been taken by businesses in protecting the integrity of their business processes and critical data. Interest in RBAC arose as a result of an evaluation of information security technology, which at one time was focused on the confidentiality needs associated with military and diplomatic matters. Recognition that business (and some government) applications are more focused on the need for integrity resulted both in the development of the Common Criteria for Information Security Evaluation (ISO-15408) and research attention to RBAC. Indeed, one of the first examples of a Protection Profile prepared and published using the Common Criteria was a specification for evaluating RBAC.

The description of RBAC presented here is based on a proposed standard for RBAC prepared by NIST. Under the proposed standard, RBAC deals with the elements of Users, Roles, Objects, Operations, and Permissions. A user is a person, but can be extended to a process. A role is a job function within the context of an organization. A user may be assigned multiple roles and a role may be occupied by multiple users, although the relationship between users and roles may be limited by constraints. Objects and operations depend on the system context. For example, in a DBMS an object may be a table and an operation may be a select or update. A permission is the approval to perform the operation on the object.

Core RBAC requires the capabilities to manage assignment of users to roles and manage assignment of permissions to roles. It requires that a user be able to assume multiple simultaneous roles. The proposed standard describes this as capturing the functionality of group permissions in current operating systems.

Hierarchical RBAC introduces role hierarchies, with senior roles in the hierarchy inheriting the permissions of their juniors and users assigned to senior roles being

assigned as well to the associated junior roles. Constrained RBAC introduces separation-of-duty relationships, which are static or dynamic constraints on the roles to which a user can be simultaneously assigned. An example of a static constraint is that a billing clerk is never allowed to also be an accounts receivable clerk. An example of a dynamic relationship is that the originator of a document is never also allowed to be the approver of the same document, but may approve other documents.

5. Security Environment

The security environment for GNUe consists of a number of participating components:

5.1. The operating system

Possible assumptions for the operating system environment include:

5.1.1. Linux/BSD

5.1.1.1. Security as currently supported

Currently, this consists of the familiar user-group-world, read-write-execute permission set. Access control lists can be simulated in this type of environment with some difficulty. A project is ongoing to provide support in Linux kernel 2.5 (development version of kernel release 2.6) for loadable kernel modules that can implement a variety of security improvements and security-hardened versions now offered as kernel patches.

5.1.1.2. Security hardened or enhanced versions of Linux

There are a variety of projects or proposals for enhancing the security of Linux. These include security-hardened distributions (or modifications to distributions) and projects for enhancement of Linux security. Security Enhanced Linux (SE-Linux) is one of the most important new concepts for improvement of Linux security (and indeed for advancement of operating system security in general). Rule Set Based Access Control (RSBAC) is another enhancement that has a research basis comparable to that of SE-Linux. There are other proposals and offerings also available.

SE-Linux is a concept posted on the NSA web site. It is currently an incompletely developed addition to Red Hat 6.1. It also could possibly be further developed or

extended by the DARPA project on advanced open-source operating system security for which proposals closed in early March 2001.

SE-Linux is an extension and generalization of Role-Based Access Control (RBAC), which in turn is based on the separation-of-duty procedures long used by businesses for protecting the integrity of their business processes. An example of separation-of-duty is the common requirement for a several people (such as a customer, clerk, and supervisor) to sign documents such as refund vouchers in retail businesses.

SE-Linux combines RBAC with other security methods known as Type Enforcement and (optionally) Multi-Level Security sensitivity labels. All three security methods are used in conjunction with a set of user-defined policies. The RBAC and Type Enforcement create a large number of categorizations including object classes, domains, types, and roles. For example, object classes include processes, files, directories, character device, block device, socket, and numerous other system elements. Within each object class there may be a number of types. For example, there may be a type associated with a specific operating system function, such as creation of the system log. User-defined policies could even extend types to specific user functions, such as approving expense vouchers. Users and processes are also assigned roles, such as ordinary user, system administrator, purchasing agent, financial auditor, and other organization-related categories. Sensitivity labels can be optionally used to identify data according to categories of consequences resulting from unauthorized disclosure, alteration, destruction, or denial-of-use.

In SE-Linux, all accesses and transitions among objects of various types and users of various roles are governed by permissions defined by policy rules and enforced by a reference monitor that is part of the operating system kernel. The permissions are much more fine-grained than in current Linux systems. For example, existing Linux systems define permissions of read, write, and execute but SE-Linux permissions may also include create, get attributes, set attributes, create hard link, lock/unlock, mount, unmount, and others.

RSBAC offers support for a variety of “rule sets” each of which supports a different kind of security policy. In RSBAC, the rule sets can also be combined. The policies supported by current RSBAC rule sets include:

- Security sensitivity labels under the “Bell-LaPadula Security Model”
- Functional Control, a form of RBAC focused on security administration
- Security Information Modification - an RBAC-type policy that allows only security administrators to modify security information

- A privacy model focused on satisfying the needs of the European Community privacy requirements
- Malware Scan - scanning files for malware (such as viruses) on execution
- File Flags - expanding the scope of permissions and limiting their setting to security administrators
- Role Compatibility - another form of RBAC capable of handling many more roles, including user roles
- Authorization Enforcement - a policy focused on controlling changes in process ownership
- Access Control Lists - that define subjects authorized to access each object, and their permissions

5.1.2. Windows

The assumption here would be for whatever security is provided under the then-current and previous releases of Windows.

5.1.3. Macintosh

Because the Macintosh OS-X is based on BSD, security for GNUe should assume that it is running on the Darwin capability, which is based on BSD. Security on the Macintosh would then be either the same as in 5.1.1 or would include any enhancements provided by Darwin.

5.2. The Database Management System

GNUe is intended to be capable of running on a variety of database management systems (DBMSs). Each DBMS has different security capabilities and features. The following sections provide examples of the security capabilities and features of various DBMSs.

5.2.1. Postgresql

Postgresql provides security by authenticating users and granting users and groups access to the objects of table, view, and “sequence” covering the privileges of select, insert, update, delete (rows), define rules, and all. The documentation suggests that access can be limited to specific columns by defining a view that contains only the allowable columns and granting access to that view. Currently views are read-only.

User authentication can be done on an individual host or over a network. The DBMS supports Kerberos as one means of user authentication.

5.2.2. MySQL

TBD

5.2.3. SAPdb- SQL

TBD

5.2.4. Others

TBD

5.3. The Middleware

The GNUe middleware is based on CORBA. The CORBA security functions will influence the security capability in areas, such as interconnection of distributed systems, for which the middleware provides important services.

5.4. The GNUe system functions and architectures

This will depend on the functionality provided by GNUe. See Section 7 for relevant discussion.

5.5. Physical protections and manual/paperwork procedures

An enterprise is assumed to have appropriate physical protections as part of its security policies and to enforce appropriate common practices such as password discipline. Depending on the capabilities of the operating system, DBMS, middleware, and other security-relevant components of the enterprises GNUe configuration, certain security policies will need to be enforced by paperwork or other manual procedures. For example, instead of having some transactions approved on-line, the enterprise could have the transactions approved in a paperwork system and then entered on-line after approval.

6. Examples of approaches to supporting security policies

6.1. Legal trustworthiness requirements

The example of a legal trustworthiness requirement to be discussed here is the Province of Quebec accounting auditability statute. Depending on the requirements imposed by the Quebec authorities in enforcing their law, there may be several ways to satisfy the Quebec accounting auditability statute. One way is to use some combination of the following steps:

1. Provide the accounting functions (such as Copy and Reverse and division of transaction status into categories of In progress, In instance of approval, and Posted) as described in the General Ledger Theory of Operation.
2. Divide the accounting database into two parts, a Posted database and a Non-posted database. Make the Posted database read-only, except for writing by a privileged process. Merge the two databases in a view for purposes of display and reporting.
3. Periodically write the Posted database to non-alterable CD-Recordable media, and provide facilities for display and report-generation from the recorded database. This could be done at the end of each accounting period. From an information security viewpoint, this effectively keeps the records added to the database since the last CD-R writing in the state of In instance of approval, because it is feasible for a sufficient number of people having a sufficient combination of privileges on

the computer system to modify the database until it is written to CD-R. Modification of the unrecorded database would be feasible even with the most advanced security systems currently envisioned in the security research community (i.e., those based on the concepts of SE-Linux).

A second way could be accomplished by the following steps:

1. Provide marking and access control at the record level using features of the DBMS and the GNUe Client. Relevant features of the DBMS will need to be determined and relevant features of the GNUe client will need to be designed and implemented.
2. Periodically write the posted records to CD-R media as discussed above.

6.2. Critical Infrastructure Protection requirements

Enterprises that are required to satisfy Critical Infrastructure Protection requirements will need to take the following steps:

1. Create comprehensive security policies and develop an overall security architecture.
2. Create and implement a specific protection plan tailored to the relevant tradeoffs for each type of data managed in the system.
3. Acquire and use a security-hardened operating system at the leading edge of security capability. An example of such an operating system would be one based on the concepts of SE-Linux.
4. Acquire and use the most advanced encryption capabilities available to the enterprise for all communications outside the perimeter controlled by the enterprise's security architecture. For most applications outside specialized government systems, this capability will be the Advanced Encryption System (AES) that resulted from an international competition conducted by NIST.

6.3. RBAC

The ability of a GNUe configuration to support a given level of RBAC will depend on the capabilities of the various elements of the security environment. This will in turn determine the kinds of organizational security policies that the GNUe configuration can support. If an organization desires to implement policies beyond the capabilities of its

GNUe configuration, the remaining capabilities will need to be supported by manual and paperwork procedures such as checking for the necessary signatures on paper documents before the associated transactions are entered into GNUe.

7. Implications for GNUe

7.1. Pass-through functionality

GNUe may need to provide functions that essentially provide the user with direct access to operating system or DBMS security functions. These could include:

- Login screens
- Security administration
- Security policy setup

7.2. Locations in GNUe where access can be controlled

The places in GNUe where access can be controlled include:

- Files/directories containing modules and packages of code. Some modules will be generally required for a wide variety of functions. Others may be specific to a certain business process. While modules must be at least executable by users who need to perform the functions they implement, users should be denied any access to modules they do not need for their business functions.
- Files/directories containing definition data used in the processing. These particularly include:
 - GNUe form definitions (gfd),
 - GNUe report definitions (grd), and
 - Possibly GNUe class definitions (gcd).
- Operating system files/directories containing database tables and metadata. This applies only if the DBMS uses regular operating system files for its tables. Some

DBMSs (such as Oracle) use an operating system file to define a storage region that they manage internally for storage of database tables. With such DBMSs, it will be necessary for the user to depend on the security features of the DBMS to control access.

- Database tables and items, using the features provided by the DBMS.
- ORBs and other middleware functions. Analysis of the CORBA Security documents indicates that CORBA security essentially provides a means for extending operating-system-based security policies across the network. However, the particular ORB needs to have the infrastructure to enforce the policies.
- User logins and other functions directly controlled by the operating system.

7.3. Virtually combined data structures with separate access to components

This aspect especially involves gfd's where parts of the form are read/write for some users and read-only for others, or define actions (such as approvals) that only certain users are allowed to perform. One approach for enforcing control would be to split the form into read/write areas, read-only areas, and action areas each defined by a partial gfd. Implementation of the full form would require two or more of the partial gfd's. Users would be allowed access to the appropriate version (read/write, read-only, and/or action) of the partial gfd as determined by rules enforced by the operating system. However, this will require processing code that allows partial gfd's to be defined and combined in processing to create the overall form.

7.4. Actions that include security calls

In some cases, actions will involve security calls to the operating system. For example, suppose the policies require that after approval of a transaction only certain kinds of accesses can be made to the transaction. Then approval of the transaction will require either that the access permissions on the file containing the transaction data be changed from pre-approval access to post-approval access, or that the transaction data be moved from a file allowing pre-approval access to another file allowing only post-approval access. The former case involves security calls to the operating system to change the relevant permissions. The latter case may involve security calls to allow the movement of the data to take place.

8. Specific guidelines

TBD

9. References

TBD

10. Appendix

GNU Free Documentation License

TBD