Draft Proposed GNU Enterprise Security Framework

Stanley A. Klein

Version 0.3.1

December 5, 2002

1. Introduction and Overview

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

A. There are three major purposes of GNUe security:

(1) 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,

(2) 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, and

(3) To permit GNUe users to obtain the security functionality they require at a level of security assurance appropriate to their needs.

B. GNUe is intended to work with a variety of supporting environments. These environments include security features and functions provided by

(1) The operating system and associated security related tools

(2) The database management system (DBMS) or other data management function

(3) The middleware, or other support for interoperability and distributed processing

(4) The functions provided by GNUe itself, and

(5) Physical protections and manual/paperwork procedures.

C. 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.

D. GNUe provides support for enforcement of business rules. There is often a fine line between enforcement of business rules and enforcement of security policies. GNUe will not itself provide advanced levels of assurance in enforcement of security policies. If such advanced levels of assurance are needed for example, if they are required as a result of external obligations −− GNUe will facilitate use of the advanced assurance security services and security policy implementation capabilities provided by the operating system and DBMS. Implementing advanced assurance security functions will require that the user select an operating system and DBMS that provides such functions at the level of assurance appropriate to the user's needs. It is also recognized that some security policies may need to be implemented using paperwork and manual procedures.

E. 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.

F. 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.

Establishment of security policies is a function that must be performed at the highest levels of enterprise governance, such as the Board of Directors, Chief Executive Officer, Owner, Managing Partner, or other high level governing entity. The security policies must be aligned with other policies regarding business practices, authority to perform business functions, applicable laws snd contractual obligations of the enterprise, and authority to originate, modify, and disseminate business documents. The scope of issues addressed in these policies must at least cover the issues that would be addressed in policies governing records and business processes intended to be implemented in paper documents.

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. The System recovery can be started only after the evidence has been collected and secured. An example of a requirement for evidence collection is the chain of custody requirement For example, in the US (and probably in many other places) evidence is subject to a in which an audit trail must be established to document the custody and handling of the evidence. If the chain of custody can not be proven in court, the evidence can be suppressed and disallowed

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. Discussion of Security Policy Drivers

2.1 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 proceses. 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.

2.2 Need−to−Know/Need−to−Perform

Need−to−know and need−to−perform are components of most security policies. Users should be given access only to data for which they have a business reason to access. Need−to−know applies to the confidentiality goal and need−to−perform applies to the other goals.

Examples of need−to−know and need−to−perform include the requirements that a patient's medical records be accessible only by providers to that patients and others whose duties require them to access the records. Similar requirements can apply to customer records, student records, and other information.

2.3 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 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.

2.4 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.

2.5 Other legal requirements

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

! Legally mandated privacy policies. Examples of these include policies mandated in the US for medical providers (including hospitals, physicians, and others) under the Health Insurance Portability and Accountability Act (HIPAA) and for educational institutions under the Family Education Rights and Privacy Act (FERPA)

! 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.

2.6 Consequences to the enterprise from security breach

Many organizations began in the late 1980's to adapt the hierarchical concept, traditionally used by governments for military and diplomatic purposes, of categorizing the needed intensity of protection for information based on the consequences to the organization of a security breach. The consequence definitions generally apply to the security objectives of confidentiality (consequences if the information is disclosed to unauthorized recipients), integrity (consequences if the information is maliciously modified or destroyed), and availability (consequences if the information is made unavailable to legitimate users). The consequence definitions generally include three levels, with a possible fourth:

! Minimal consequences − minimal but identifiable damage to the enterprises operations, image, reputation, customer relations, supplier relations, regulatory relations, investor relations, or employee relations. This is typically damage in the category of nuisance, identifiable expense, waste of time, and minor embarrassment.

! Significant consequences − Significant damage to the enterprises operations, image, reputation, customer relations, supplier relations, regulatory relations, investor relations, or employee relations. This is typically damage that is costly and time consuming, involves major embarrassment and major loss, and may involve significant violation of the legal obligations of the enterprise or its leadership.

! Severe consequences − Severe, and often irreparable, damage to the enterprises operations, image, reputation, customer relations, supplier relations, regulatory relations, investor relations, or employee relations. This is damage that seriously affects the health, viability, competitive capability, or sustained existence of the enterprise, and may involve very serious violation of the legal obligations of the enterprise or its leadership.

The fourth category that is sometimes added is public information that is publicly disclosed and generally available, but might involve minimal consequences if it were vandalized.

2.7 Formal security capability definition, evaluation, and certification

For at least the last decade, government agencies and international standards bodies have been engaged in a process of developing criteria for defining, evaluating, and certifying security protection of computer systems and associated networks. The first of these criteria was developed in the 1970's to address US Defense Department issues and is known as the "Orange Book" (because of the color of its cover). The Orange Book was principally focused on confidentiality. Expanded criteria, including criteria for integrity, important to non−defense and commercial users, have been developed by an international consortium of information security agencies and have been adopted as an international standard.. The criteria specify levels of trust, including security features, development practices, documentation, and test procedures. The choice of which level of trust is appropriate has been left to the user.

The Orange Book criteria illustrate the approach. Orange Book Class D is security that is present but too weak to evaluate. Class C provides access control based on data−owner determined permissions. Subclass C1 provides basic features, including passwords, access control, formal quality assurance, and basic documentation. Subclass C2 adds requirements such as audit capability. Class B adds requirements for security sensitivity labels attached to files and security clearance levels attached to user accounts. A user can access a file only if the files owner has provided access permission *and* the files sensitivity label is at or below the users security clearance level. Class B has three subclasses with increasing requirements for features and quality assurance, such as having a team of experts take several months full time to attack the system. Class A1 is similar to Class B3, except arcane mathematical methods must be used to prove that the system meets its specifications.

The expanded criteria developed by the international consortium of information security agencies is known as the Common Criteria and Methodology for Information Technology Security Evaluation. The equivalent international standard is International Standards Organization IS 15408, which has formatting changes but is aligned with the Common Criteria. Information on the Common Criteria can be found on the US National Institute of Standards and Technology (NIST) web site.

The Common Criteria provides a collection of terms and specification components for preparing Protection Profiles, which are user/acquirer expressions of their security requirements, and Security Targets, which are provider expressions of the security capabilities of their products. Protection Profiles and Security Targets have similar formats.

The international consortium has established an arrangement for standardizing Protection Profiles and for evaluating products against the profiles. Work done in one country under the Common Criteria arrangement is generally accepted in all countries in the consortium.

The Common Criteria provides a set of Evaluation Assurance Levels (EAL) that address the levels of trust that can be placed in the security evaluations and products. EAL−1 is similar to Orange Book Class D, i.e., security that is present but too weak to evaluate. EAL−2 is sometimes described as common, commercial grade security. EAL−4 is the highest level that can be evaluated in the US on a commercial basis, i.e., outside government auspices. EAL−7 is comparable to Orange Book A−1, requiring arcane and extrordinarily complex methods to demonstrate that the security protection is provided.

The development methods required under the Common Criteria are difficult to interpret and apply to software developed under free/open−source software projects, that involve the efforts of volunteer developers and reuse components resident in what Lawrence Lessig (in his book The Future of Ideas) has called the Innovation Commons. The George Washington University has a project to evaluate Security Enhanced Linux (discussed in further detail in Section 6) under the Common Criteria as a pilot study of how to apply the Common Criteria to free/open−source software projects. Their intent is to evaluate at EAL−2 in the next year or so, and at EAL−4 within a few years. They are working with NIST to develop appropriate interpretations of the Common Criteria that can be applied to similar projects.

3. Security categories

The following sections describe a categorization of potential GNUe users, the threats they may regard as relevant, and the security assurance levels that may be appropriate for their needs. In describing the assurance levels, a simplified set will be used. These include low assurance security (roughly corresponding to EAL−1), medium assurance security (roughly corresponding to EAL−2), and high assurance security (roughly corresponding to EAL−4).

3.1 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.

A company in this category could easily use a system having a low security assurance level.

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 protection of personal privacy in medical facility records under the US Health Insurance Protection Availability and Accountability Act (HIPAA).

The principal threats of concern to such an enterprise are:

! Inadvertent error

! Violation of legal/contractual/business policy requirements for non disclosure or integrity protection of certain data (such as employee records, proprietary data, health records, or accounting audit trails).

The security assurance level required for a user in this category will depend on the particular provisions associated with the legal or contractual obligations. It may be possible to satisfy some obligations using a system having low security assurance. However, if the obligations require the system to be subject to outside audit, an assurance level of at least medium may be advisable.

2.3 Category C: Small/medium company, legal and contractual requirements, any external network connection tightly controlled

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 or otherwise tightly controlled protection 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/businesss policy requirements for non disclosure or integrity protection of certain data (such as employee records, proprietary data, and accounting audit trails).

The security assurance level appropriate for a user in this category will depend on the nature of the external contractual/legal obligations and the relative technical sophistication of the threat from insiders. If the external obligations are relatively simple, the insider technical threat relatively unsophisticated, and the consequences from both threats relatively minimal, a system having low security assurance may be sufficient. As the external obligations become more stringent, the insider threat more sophisticated, and the consequences from both threats more serious, the need for higher levels of security assurance increases.

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/businesss 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

The threat of external intrusion has the effect of raising the level of sophistication of potential attacks. A user in this category will most likely require at least a medium level of assurance, unless the most critically affected systems can be effectively isolated. The user must choose between the security benefit and the enterprise integration disbenefit of isolation or stovepiping.

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

Enterprises faced with special security concerns will most likely require high assurance systems.

4. Types of Access Control and Other Security Functions

4.1 Role−Based Access Control (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.

4.2 Discretionary Access Control (DAC)

Discretionary Access Control is the traditional user−group−other/read−write−execute type of control traditionally found in operating systems and DBMSs. It is also the kind of control provided by access control lists. Under DAC, the owner of the data or file essentially has discretion to provide access to whoever the owner determines should have access. The system enforces the owners access decision, but does not otherwise enforce constraints on access to the data.

4.3 Mandatory Access Control (MAC)

In Mandatory Access Control, objects (e.g., data) and subjects (e.g., users, devices) are given sensitivity labels according to a hierarchy. The label is part of the access control associated with the subject or object. Security policies govern the access and movement of objects by subjects. An example of a security policy is the Bell LaPadula Security Model that prohibits a subject having a lower level sensitivity label from reading an object having a higher sensitivity label and also prohibits a subject having a higher level sensitivity label from writing an object to a subject (e.g., a user directory or a printer) having a lower sensitivity label. The policy is often summarized as No read up, no write down and is enforced by the operating system.

4.4 Authentication

Authentication is the process of determining that the user is authentic, i.e., that the user is who the user claims to be. This is done by receiving information about the user and comparing the received information to a stored version of the information for the authentic user. Up to three factors may be used:

! Something the user knows, such as a password

! Something the user has, such as a device or smartcard, usually identified by some kind of encrypted information. Some devices automatically change the information periodically in synchronism with other software or devices in the authentication system.

! Something the user is, essentially data regarding a biometric characteristic of the user, such as a fingerprint or eyeball pattern, generally stored in some encryption protected format.

There are numerous ways in which an authentication system can be attacked and compromised. These include various means of tricking a user into revealing a password, various strategies for guessing passwords and validating the accuracy of the guesses, and various methods of capturing passwords (or other authentication information) as it moves in the system. There are also ways in which an authentication system can be bypassed, essentially involving attacks on the security of the overall system.

4.5 Captured User Approaches

A captured user approach involves capturing or jailing the user to prevent any access to capabilities that a malicious user could exploit to engage in unauthorized activities on the system. For example, this would generally involve sending the user from system login directly into a menu system from which the user can't escape. Sending the user into the menu system generally involves a function that is automatically executed upon startup of a computer or upon user login. However, there are a wide variety of system functions that must be blocked to ensure that the user remains captured.

In general, the capturing fails if a user is able to access a system prompt, or also in the case of GNUe a Python traceback and prompt, that enables access to commands that can be used for performing functions that support disallowed activity. Among other things, this may mean that the user must be prevented from starting the system or logging in without going throught the auto−execute function that starts the menu system. It means that functions that can stop a process and return to the system prompt (such as Control−C or Control−Z on some systems) must be disabled. It means that any exception that could result in a crash leading to a Python traceback or system prompt must be handled and returned instead to the menu system. It is best if functionality not needed by a legitimate user is not present on the system.

Captured user approaches are good for purposes such as point−of−sale terminals and specialized kiosk terminals. Also, any user accessing a web page is essentially a captured user of the system containing the web server.

5. Examples of actual security requirements

This section will provide some examples of actual security requirements drawn from a variety of sources.

5.1 Corporate Mandatory Access Control Security Labels

Appendix A provides an extract from Internet Request for Comments (RFC) 3114 describing the sensitivity labels used by three major corporations. The lowest level is usually some variant of "public information" and the top level some variant of "highly sensitive" information that could result in severe damage to the enterprise if disclosed or subjected to tampering. For example, Amoco Corporation has HIGHLY CONFIDENTIAL, CONFIDENTIAL, and GENERAL for confidentiality and MAXIMUM, MEDIUM, and MINIMUM for integrity, with an additional category, CRITICAL, for availability of time−critical information. Caterpillar, Inc. uses Caterpillar Confidential Red, Caterpillar Confidential Yellow, Caterpillar Confidential Green, and Caterpillar Public for confidentiality. Whirlpool Corporation uses confidentiality categories of WHIRLPOOL CONFIDENTIAL, WHIRLPOOL INTERNAL, and WHIRLPOOL PUBLIC, with WHIRLPOOL CONFIDENTIAL possibly having additional markings such as MAKE NO COPIES, THIRD PARTY CONFIDENTIAL, ATTORNEY−CLIENT PRIVILEGED DOCUMENT, DISTRIBUTION LIMITED TO ____, and COVERED BY A NON−ANALYSIS AGREEMENT. The definitions of these categories are provided in the appendix.

5.2 RBAC Example From Banking

An RBAC example from banking can be found in a document entitled Application of XML Tools for Enterprise−Wide RBAC Implementation Tasks by Ramaswamy Chandramouli of the US

National Institute of Standards and Technology (NIST). The document is posted on the NIST RBAC web site. The roles described in the document include:

(a) Teller Input and Modify transactions against Customer Deposit Accounts.

(b) Customer_Service_Rep In addition to the functionality for the Teller Role, create and delete Customer Deposit Accounts.

(c) Loan_Officer Create and Modify status of Loan Accounts

(d) Accountant Input all bank business transactions and generate General Ledger Reports.

(e) Accounting_Manager In addition to the Accountant functions, the ability to modify Ledger Posting Rules

(f) Internal_Auditor Verify all Transactions and Ledger Posting Rules.

(g) Branch_Manager Ability to perform any of the functions of other roles in times of emergency and to View all transactions, Account Statuses and Validation Flags.

Some of the constraints imposed on roles include:

(a) The maximum number of users that can be assigned to Bank_Manager and Internal_Auditor roles is ONE.

(b) The following pair of roles cannot be assigned to the same user (Static Separation of Duty (SSD) or Membership Mutual Exclusivity):

(1) Customer_Service_Rep and Accounting_Manger

(2) Customer_Service_Rep and Internal_Auditor

(3) Loan_Officer and Accounting_Manager

(4) Loan_Officer and Internal_Auditor

(5) Accounting_Manager and Internal_Auditor

(6) Teller and Accountant

(7) Teller and Loan_Officer

(8) Teller and Internal_Auditor

(9) Accountant and Loan_Officer

(10) Accountant and Internal_Auditor

(c) The following pair of roles cannot be activated or enabled at the same user session (Dynamic Separation of Duty (DSD) or Activation Mutual Exclusivity):

Customer_Service_Rep and Loan_Officer

5.2 RBAC Example From Health Care

An RBAC example from health care can be found in a document entitled Application of XML Tools for Enterprise−Wide RBAC Implementation Tasks by Ramaswamy Chandramouli of the US National Institute of Standards and Technology (NIST). The document is posted on the NIST RBAC web site.

The document provides an example of a portion of a system applying to a health care facility. Users are assigned to roles, including Admissions_clerk, Ward_scheduler, Registered_nurse, and

Facilities_specialist. The tables below provide some of the flavor of the requirements.

There is a role−domain mapping as shown in the following table:

There is a subject−role mapping as shown in the following table:

There is a subject−domain mapping as shown in the following table:

Finally, there is a domain access matrix that governs the access to types of data in the various domains:

Access Mode Codes: C Create , U Update, D Delete , V View

5.3 Security Requirements Arising from FERPA

This example of a FERPA policy is extracted from the web site of Western Michigan University. Under FERPA, a university must define student directory information and certain other information. Students are allowed under FERPA to restrict the release of their directory information. (Another university allows two levels of control: (1) stating to a requester that the information on the student can not be released, and (2) stating to a requester that there is no information available on any such student.)

Western Michigan University defines student directory information as follows:

Name

Address

Telephone number

Email Address

Date and place of birth

Curriculum and major field of study

Dates of attendance

Enrollment status (full/part time)

Degrees/Awards received

Most recent previous educational agency or institution attended by the student

Participation in officially recognized activities and sports

Weight/Height of members of athletic teams

Under Western Michigan Universitys policies, grades and grade point averages can never be released outside the University without the written consent of the student. Other educational records include admissions, personal, academic, and financial files, academic, cooperative education, and placement records. There are rules for access to these records. According to the policies, Educational records do not include the records of instructional, administrative, and educational personnel, which are the sole possession of the maker and are not accessible or revealed to any individual except a temporary substitute; records of the law enforcement unit;

student health records; employment records; or alumni records. Health records, however, may be reviewed by physicians of the students' choosing.

Student's may not inspect and review the following, as outlined in the Act: financial information submitted by their parents, confidential letters and recommendations associated with admissions, employment, or job placement, or honors to which they have waived their rights of inspection and review; or educational containing information about more than one student, in which case the institution will permit access only to that part of the record which pertains to the inquiring student. The institution is not required to permit students to inspect and review confidential letters and recommendations placed in their files prior to January 1, 1975, provided those letters were collected under established policies of confidentiality and were used only for the purposes for which they were collected.

As can be seen from these polcies, FERPA creates rules for access to a variety of records, including rules for disclosure and for review and correction of claimed errors. Certain records are controlled according to rules selected by the student, others can not be disclosed to the student.

Some of these records are appropriate for databases, and others are more likely to be maintained as paper documents.

6. Security Environment

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

6.1 The operating system

Possible assumptions for the operating system environment include:

6.1.1 Linux/BSD

Security as currently supported and planned:

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. The Loadable Kernel Module effort has developed "kernel hooks" that can be used by a wide range of modules, and includes a stacker that enables multiple modules to be used.. The loadable modules will come with the system, just as modules for various devices are currently supplied. A user can install a module using the "insmod" command and the system will then enforce the security policy it provides (subject, of course to the appropriate configuration of the security policy. The modules offer options that both increase and reduce security. For example, there is a module that eliminates the security that is already provided in the kernel to support the needs of some real time critical users who can't afford the time it takes for the existing, built in access control.

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 includes all the additions as a kernel patch currently being incorporated into the Loadable Kernel Module version.

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

Note that the improvements to the Linux kernel identified in plans for version 2.6 include Access Control Lists, identified separately from the Loadable Kernel Module effort.

6.1.2 Windows

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

6.1.3 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.

6.2 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.

6.2.1 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.

6.2.2 MySQL

TBD

6.2.3 SAPdb SQL

TBD

6.2.4 Others

TBD

6.3 The Middleware

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

6.4 The GNUe system functions and architectures

This will depend ont the functionality provided by GNUe. See Section 6 for relevant discussion.

6.5 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.

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

Some of these functions are already provided in GNUe. Some may need to be modified, for example to facilitate a role−based login or role change under SE−Linux or another RBAC supporting operating system or DBMS. It may be appropriate to provide security adapters for different operating systems to facilitate access to these functions.

7.2 Locations in GNUe where access can be controlled or security enforced

The places in GNUe where access can be controlled or security can be otherwise enforced include:

! Files/directories containing definition data used in the processing. These particularly include:

(1) GNUe form definitions (gfd),

(2) GNUe report definitions (grd)

(3) GNUe process definitions (gpd), and

(4) Possibly GNUe class definitions (gcd).

! 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.

! 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.

! User logins and other functions directly controlled by the operating system.

! Audit logs written to secure areas by GNUe processing functions. These logs could be generated by trigger functions or could be provided as a feature similar to the existing debug logging functions. Audit logging can be used as an alternative to access control in some situations.

! Data contained in individual files rather than maintained under database management systems. For example, a flat file, an XML file, or a database implemented as a Dbase (dbf) file could be used for providing certain forms of protection.

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.

Such an approach could be implemented by providing include capability in gfd files. The actual gfd's accessed by various users would be created by mixing and matching various versions of the partial gfd's having capability to view various parts of the database and to perform various actions. It may also help to provide capability in Navigator gpd files to provide an entry but grey it out. This would allow the Navigator menus to be common across all users but to actually provide only those functions appropriate to a particular user.

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 permisions 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.

Such an approach could be implemented by supporting the security calls in trigger functions.

7.5 Security for Appserver

Appserver is the GNUe tool that manages multi−tier architectures. Appserver is intended to provide a consistent interface to the user client regardless of where or how in a system any particular item of data is actually stored. Appserver security is made challenging because of the complexity of mapping data items from the user side of appserver to the database side. Having the operating system or database system provide security requires that the mapping be traced and the various data items appropriately protected.

In certain cases, there are potential security requirements that can not be met by either a particular operating system or a particular database system. Appserver can be provided with its own internal security functions. Such functions are likely to have a lower level of security assurance than functions provided by an operating system, although it may be possible to improve the quality of protection by using very simple approaches and providing as much protection as possible using the operating system.

The main point here is that the security features of Appserver must be approached very carefully to avoid precluding their use by enterprises having demanding requirements for security.

8. Examples of approaches to supporting security policies in GNUe

8.1 Static RBAC focused on transactions

This policy involves roles that are static, i.e., each person has a limited number of roles that do not change. It is also focused on transactions, i.e., the access control is by specific screens or reports associated with the transaction..

One way of implementing such a policy would be by operating system access controls placed on the gfd and grd files associated with the transactions, together with providing a set of gpd files, one for each role, capable only of initiating the activities allowable for the role. The system would need to be configured so a user could not substitute a tampered copy of a gfd, grd, or gpd file by somehow redirecting the system to use the tampered copy.

If the transactions involve some critical individual database tables, access control could also be placed on those tables using the access control facilities provided by the database system.

8,2 Subject records viewable only by specified providers

Examples of this would include patient records accessible only by providers to that patient, student records accessible only by personnel of the department in which the student is enrolled, or customer financial records accessible only by personnel of the office at which the customer has an account.

One relatively secure way to support this kind of policy would be to place the record of each patient, student, or customer in a separate operating system file, such as a flat file, an XML file, or a Dbase dbf file. Access control could then be provided by the operating system, and the files could be easily moved if the status of the patient, student, or customer changed.

Another approach would be to create a set of database tables having access limited to the providers, department, or office providing service, and to move the records to those tables.

8.3 Access control on database fields

An example of this policy would be one that restricts the users allowed to access price information. If only a few screens contain price infomation, the same strategy discussed above in Section 8.1 could be used. If numerous screens contain price information the challenge becomes more difficult. Some database systems allow access control by field. If one of these is being used, the facility that supports that access control could support the policy. If such a database is not being used, it will be necessary to use access control facilities provided by GNUe. The GNUe facilities provide less assurance than an operating system or database system, and the user must take that lower assurance into account in evaluating the risk of disclosong database field information to unauthorized personnel.

8.4 More examples TBD

8.n 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.

8.n+1 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.

8.n+2 RBAC of higher complexity

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.

9. Specific guidelines

TBD

Appendix A

Extract from Internet Request for Comments (RFC) 3114

Descriptions of Sensitivity Labels Used by Three Major Corporations.

Network Working Group W. Nicolls

Request for Comments: 3114 Forsythe Solutions

Category: Informational May 2002

RFC 3114 Implementing Company Classification Policy May 2002

2. Developed Examples

2.1 Classification Policies

The following describes the information classification policies in effect at 3 companies.

2.1.1 Amoco Corporation

The description for the Amoco information classification policy was taken from the Amoco Computer Security Guidelines. Amoco classifies its information assets based on confidentiality and integrity and defines 3 hierarchical classifications for each. The confidentiality and integrity polices are independent, so either or both may be applied to the information. Amoco also defines an availability classification for time critical information.

HIGHLY CONFIDENTIAL − Information whose unauthorized disclosure will cause the company severe financial, legal or reputation damage. Examples: Certain acquisitions, bid economics, negotiation strategies.

CONFIDENTIAL − Information whose unauthorized disclosure may cause the company financial, legal, or reputation damage. Examples: Employee Personnel & Payroll Files, some interpreted Exploration Data.

GENERAL − Information that, because of its personal, technical, or business sensitivity is restricted for use within the company. Unless otherwise classified, all information within Amoco is in this category.

MAXIMUM − Information whose unauthorized modification and destruction will cause the company severe financial, legal, or reputation damage.

MEDIUM − Information whose unauthorized modification and destruction may cause the company financial, legal, or reputation damage. Examples: Electronic Funds, Transfer, Payroll, and Commercial Checks.

MINIMUM − Although an error in this data would be of minimal consequence, this is still important company information and therefore will require some minimal controls to ensure a minimal level of assurance that the integrity of the data is maintained. This applies to all data that is not placed in one of the above classifications. Examples: Lease Production Data, Expense Data, Financial Data, and Exploration Data.

CRITICAL − It is important to assess the availability requirements of data, applications and systems. A business decision will be required to determine the length of unavailability that can be tolerated prior to expending additional resources to ensure the information availability that is required. Information should be labeled "CRITICAL" if it is determined that special procedures should be used to ensure its availability.

2.1.2 Caterpillar, Inc.

The description for the Caterpillar information classification policy is taken from the Caterpillar Information Protection Guidelines. Caterpillar classifies its information assets based on confidentiality and defines 4 hierarchical classifications.

Caterpillar Confidential Red − Provides a significant competitive advantage. Disclosure would cause severe damage to operations. Relates to or describes a long−term strategy or critical business plans. Disclosure would cause regulatory or contractual liability. Disclosure would cause severe damage to our reputation or the public image. Disclosure would cause a severe loss of market share or the ability to be first to market. Disclosure would cause a loss of an important customer, shareholder, or business partner. Disclosure would cause a long−term or severe drop in stock value. Strong likelihood somebody is seeking to acquire this information.

Caterpillar Confidential Yellow − Provides a competitive advantage. Disclosure could cause moderate damage to the company or an individual. Relates to or describes an important part of the operational direction of the company over time. Important technical or financial aspects of a product line or a business unit. Disclosure could cause a loss of Customer or Shareholder confidence. Disclosure could cause a temporary drop in stock value. A likelihood that somebody could seek to acquire this information.

Caterpillar Confidential Green − Might provide a business advantage over those who do not have access to the same information. Might be useful to a competitor. Not easily identifiable by inspection of a product. Not generally known outside the company or available from public sources. Generally available internally. Little competitive interest.

Caterpillar Public − Would not provide a business or competitive advantage. Routinely made available to interested members of the General Public. Little or no competitive interest.

2.1.3 Whirlpool Corporation

The description for the Whirlpool information classification policy is taken from the Whirlpool Information Protection Policy. Whirlpool classifies its information assets based on confidentiality and defines 3 hierarchical classifications. The policy states that:

"All information generated by or for Whirlpool, in whatever form, written, verbal, or electronic, is to be treated as WHIRLPOOL INTERNAL or WHIRLPOOL CONFIDENTIAL. Classification of information in either category depends on its value, the impact of unauthorized disclosure, legal requirements, and the manner in which it needs to be used by the company. Some WHIRLPOOL INTERNAL information may be authorized for public release."

WHIRLPOOL CONFIDENTIAL − A subset of Whirlpool Internal information, the unauthorized disclosure or compromise of which would likely have an adverse impact on the company's competitive position, tarnish its reputation, or embarrass an individual. Examples: Customer, financial, pricing, or personnel data; merger/acquisition, product, or marketing plans; new product designs, proprietary processes and systems.

WHIRLPOOL INTERNAL − All forms of proprietary information originated or owned by Whirlpool, or entrusted to it by others. Examples: Organization charts, policies, procedures, phone directories, some types of training materials.

WHIRLPOOL PUBLIC − Information officially released by Whirlpool for widespread public disclosure. Example: Press releases, public marketing materials, employment advertising, annual reports, product brochures, the public web site, etc.

The policy also states that privacy markings are allowable. Specifically:

For WHIRLPOOL INTERNAL, additional markings or caveats are optional at the discretion of the information owner.

For WHIRLPOOL CONFIDENTIAL, add additional marking or caveats as necessary to comply with regulatory or heightened security requirements. Examples: MAKE NO COPIES, THIRD PARTY CONFIDENTIAL, ATTORNEY−CLIENT PRIVILEGED DOCUMENT, DISTRIBUTION LIMITED TO ____, COVERED BY A NON−ANALYSIS AGREEMENT.