Global Information Assurance Certification Paper

Jeff Langford
GIAC Security Essentials Certification (GSEC)
Option 1 Version 1.4b
July 5, 2003

Implementing Least Privilege at your Enterprise

Introduction

Enterprise security involves people, process and technology. The principle of least privilege can and should be applied to all of those areas

An expansion of the topic of 'least privilege' has some importance because, those responsible for information security, have had some past difficulty explaining it or gaining acceptance for this important principle. It is often referenced and occasionally supported with a brief definition, but rarely is the principle supported with any significant examples or rationale. It is a principle that touches many aspects of the organization or enterprise, and since it is not really well explained or understood it is difficult to achieve acceptance. This paper will provide some background, offer some rationale to help develop support for it's acceptance, and identify ways it can be implemented at your enterprise.

What is 'least privilege'?

The principle of least privilege is routinely described today as a basic security principle. It is interesting to note that the 'Principle of Least Privilege' was first published over 30 years ago, as a secure system design principle, in a paper by J.H. Saltzer and M.D. Schroeder

> Least Privilege: Every program and every user of the system should operate using the least set of privileges necessary to complete the job. Primarily this principle limits the damage that can result from an accident or error. [1]

In 1975 when this principle was published it was one of 8 design principles that were originally intended to implement appropriate security restrictions on operating systems. The underlying consideration for the principle of least privilege was that 'user computations have to be protected from each other'. [2]

Prior to that a task force was assembled by the Department of Defence to address computer security safeguards that would protect classified information. Their work resulted in the publication of the Department of Defence Trusted Computer System Evaluation Criteria (TCSEC) in December of 1985. The published work, identified as DOD 5200.28-STD, produced a variation of the

Saltzer and Schroeder design principle, and introduced additional rationale to justify the implementation of 'least privilege'.

> Least Privilege - This principle requires that each subject in a system be granted the most restrictive set of privileges (or lowest clearance) needed for the performance of authorized tasks. The application of this principle limits the damage that can result from accident, error, or unauthorized use. [3]

In this definition the target of the principle is generalized so it can be easily applied to different aspects of the enterprise. As well, this particular definition specifically identifies the potential of unauthorized use as further rationale for introducing this principle.

In some other variations of the principle, the concept of time is introduced into the definition. The concept of limiting the amount of time that the privilege is available is an important part of risk mitigation. In a guideline document for developers, Gary McGraw and John Viega presented a variation of the Saltzer and Schroeder design principle in the following definition:

> The principle of least privilege states that only the minimum access necessary to perform an operation should be granted, and that access should be granted only for the minimum amount of time necessary. [4]

The Saltzer and Schroeder design principles can be, and are often, applied to many aspects of enterprise security. The principle of least privilege, which is often equated with the military 'need to know' rule, is perhaps the most recognized and referenced of the all design principles. The 'need to know' rule is associated with classified data or information, and is often presented as a real life example of 'least privilege'. Another Saltzer and Schroeder design principle that is often used in conjunction with 'least privilege' is the principle of 'separation of privilege'. Similar to 'least privilege', the definition for separation has been modified and applied to many aspects of enterprise security. The original definition stated that:

> Where feasible, a protection mechanism that requires two keys to unlock it is more robust and flexible than one that allows access to the presenter of only a single key. [5]

At times the definition has been restated or presented to reflect that 'two conditions must be met to allow privilege' or 'access is never based on a single condition'. [17] The simplest example, and one that was part of the original definition, is the idea that two separate keys are required to open a bank safety deposit box. Another common example is the requirement for two different signatures on a check. In various applications this principle is renamed to 'segregation of duties' or 'separation of duties'. A National Institute of Standards

and Technology (NIST) publication focused on securing web servers offered examples of how it could be applied to various aspects of the enterprise:

> Separation of Privilege – Functions, to the degree possible, should be separate and provide as much granularity as possible. The concept can apply to both systems and operators/users. In the case of systems, such functions such as read, edit, write, and execute should be separate. In the case of system operators and users, roles should be as separate as possible. For example if resources allow, the role of system administrator should be separate from that of the security administrator. [6]

These two particular Saltzer and Schroeder design principles are effectively intertwined in almost all applications. In most instances the separation and granularity of function, provided by 'separation of privilege', is a necessary prerequisite or foundation that allows 'least privilege' to be implemented.

Least Privilege and Standards

Implementing 'least privilege' for people at your enterprise means that they will only get, and retain, the permissions and privileges required to do their jobs. As indicated, it will likely be implemented in conjunction with 'separation of duties'. Implementing least privilege will undoubtedly meet with some resistance from both management and staff. It some cases the restrictions associated with least privilege conflicts with the natural desire to be helpful and get the job done in a timely fashion. However in some cases the resistance can be answered by pointing to widely accepted best practices and information technology security standards.

In general, the first best practices that should be applied to any operation is the concept of 'due care' and 'due diligence'. 'Due care' suggests that one 'ensures that a minimum level of protection is in place in accordance with the best practices of the industry'. [7] 'Due diligence' is essentially the maintenance program that supports 'due care', and has been described as 'a protection plan (deployed) to prevent abuse, fraud and the means to detect them'. [7] What would appear to be just plain common sense to someone responsible for any operation also has legal backing. There are many examples of organizations that have been found negligent and held liable in courts of law, because they have failed to follow the best practices of the industry. These concepts are applicable to the protection of information assets in the same way that they are applicable to other aspects of an operation.

In the area of information security, there are many standards and interpretations that can be used. The ISO17799 'Information Technology - Code of Practice for Information Security Management' is a widely used and referenced standard in the area of information security. It was originally developed by the British

Standards Institution and called the BS7799. It was subsequently accepted by the International Organization for Standards and published under their title in December 2000. Many organizations have used the BS7799 or the ISO17799 as a framework or reference standard to address information security. The National Institute of Standards and Technology, or NIST, is a United States Federal government agency with a mandate to develop standards in a number of areas. The standards developed by that group in the area of information technology and information security, are also widely recognized and referenced by practitioners in the field.

The ISO17799 Standard provides recommendations for information security management. The ISO17799 Standard identifies 'access control' as one of the controls that should be implemented to mitigate risks. The principle of least privilege is reflected in the recommendations for access control. One of the overall objectives is to limit access to what is appropriate and defined by 'business and security requirements'. The concept of only what is needed, when it is needed, is covered in the category of Privilege Management and specifically stated in the following fashion:

> 9.2.2 b) Privileges should be allocated to individuals on a need-to-use basis and on an event by-event basis, i.e. the minimum requirement for their functional role only when needed. [8]

The ISO17799 Standard also identifies the concept of 'zero access' to start. That suggests that no access or virtually no access is the default, so that all subsequent access and the ultimate accumulation can be traced back through an approval process.

In a 1996 NIST publication (SP 800-17) titled, 'Generally Accepted Principles and Practices for Securing Information Technology Systems', the principles of separation of duties and least privilege were described as security rules that must be applied when granting access. This publication highlights the importance of these two principles when considering the responsibilities associated with a position or job function, as well as their importance in the area of access control.

In June 2001, a division of the NIST organization released a special publication (SP 880-27), titled 'Engineering Principles for Information Technology Security'. It identified a number of security principles that needed to be considered in the design, development, and ongoing operation of an information system. The publication was intended for a wide audience including those charged with developing functional requirements, developers, project managers, and Information Security Officers. This publication was built off the 1996 NIST publication but has a more system-level perspective. It specifically recommends the implementation of 'least privilege' as one of the security principles that should

be considered during the initiation, development/acquisition, implementation and operation/maintenance system life-cycle phases. The expansion of 'least privilege' in this publication suggests an organization should limit the number of people with administrator style access, look to separate administrative functions wherever possible, and limit the permissions associated with a process to only what is required. In a related principle, it is also recommends that identifying and removing excessive privileges from programs is a common vulnerability that should be addressed.

Insider Security Breaches

In some cases, today's reality or real-life examples can be offered when trying to gain acceptance or disarm resistance. The old adage that 'it will take something bad to happen before anything gets done' is often very applicable in organizations. In reality a significant number of security breaches or incidents are caused or initiated by insiders. In some cases this is done unknowingly or by accident. In other cases an insider with some motivation intentionally causes the security incident and this often involves an abuse of privileged access. In this situation the least privilege principle is applied to limit the potential for damage through accident, error, or unauthorized use of an information system.

A UK survey, Information Security Breaches Survey (ISBS 2002) focused on malicious security breaches and quantified some aspects of insider breaches. The ISBS 2002 identified that a significant number of the worst security incidents were caused by an internal activity. In the case of large businesses, an insider was responsible for the worst security incident 48% of the time. When small and medium size businesses were factored in, the overall number was reduced to 32%. In one specific example that was cited by a financial services provider, the insider security incident involved computer based fraud carried out by an employee who had accumulated system access privileges over a number of years. [9] As cited in this example and in most cases, fraudulent activities are usually supported by opportunity and excessive privileges. The implementation of 'least privilege' would have helped in this example. In another part of the survey, companies in all size categories identified the threats from employees as their second greatest concern. [10]

A 2002 survey conducted by the Computer Security Institute and the FBI suggests that security incidents from the inside is even more significant and is in fact on the rise. The 2002 CSI/FBI survey reported that 64% of enterprises identified that they had security incidents from the inside. This is up from the 59% figure that was reported by the same group in 2001. [11]

The emergence of the Internet and the fact that most companies now have a web presence has increased the number of external threats (i.e. hackers). The old axiom that the majority of your threats will come from insiders has had to be revised, with the changing paradigm called the Internet. In addition, a survey

result that is summarised as 'insider security incidents' will undoubtedly include a number of different types of incidents. Applying the principle of least privilege to people at your enterprise will not completely remove the potential of insider threat. However, that mitigation is part of a defence in depth approach that is designed to effectively strengthen the security posture of your enterprise.

Least Privilege applied to People

As indicated earlier, the application of 'least privilege' applies to people, process and technology. For people at your organization this application typically targets the role or function that they have, and looks to implement 'separation of duties'. In most cases, this idea is understood and well implemented in financial organizations because they must follow strict guidelines and have their controls regularly audited. Similarly the financial department at any organization will often implement 'separation of duties' for financial functions. For example most organizations have separate functions associated with initiating a payment and authorizing payment, as a common practice. The goal is to reduce opportunities and the risk for intentional or accidental misuse of systems or information.

The basic premise used in finance departments can be applied to information security at your enterprise, and specifically to typical information technology functions. The objective is to define the user functions, consider the responsibilities and privileges that are necessary to do those functions, and enforce the separation of those functions. In information technology the two most common examples referenced are the system administrator function and the role of developers.

For system administrators there are four common practices that are recommended to support 'least privilege' and separation:

- The role of system administrator (i.e. root in Unix environments, Enterprise Admin in Windows 2000), which is very powerful, should be limited to as small a group as possible.
- Implement fine grained access privileges when a specific task requires elevated privileges
- Separate system administration from regular account requirements
- Separate the system administrator and audit/logging functions.

The system administrator account in any system is typically the most powerful account, and effectively has full privileges to the entire system. In reality it is very difficult to implement restrictions on this type of account. Limiting the number of people with this account access or privilege is one mitigation option.

While limiting the access to the powerful system administrator account is a recommended best practice, very often an organization is challenged because many smaller functions or tasks need elevated privileges. For example, some

functions such as password resets, desktop support, executing backups and restores, and network administration functions require elevated privileges to run. In a Unix environment there are various products that can be used to establish a graduated privileged authority. The *sudo* facility is one such product that is publicly available. It can be used to allow specific users to run commands with root access without having access to the root account and password. The *sudo* facility uses a configuration file */etc/sudoers* to identify and match specific commands to specific users. [12] In this implementation there is a conscious decision by the Unix system administrator to isolate and delegate certain commands. For example the ability to reset a password or administer a print queue can be isolated and delegated to HelpDesk staff very easily. In a Windows 2000 environment it is relatively easy to implement a separation of function and avoid a wider distribution of the domain or enterprise administrator level accounts by taking advantage of supplied default roles. For example the HelpDesk, or those in charge of desktop support, can be made local administrator of the local desktop operating system. In some cases support for various servers can be isolated and delegated to various people using the local administrator account on the server. Support personnel responsible for backups can be given the built-in backup operator role. [20]

For a variety of reasons, it is recommended that users with system administrator style accounts use and maintain a regular user account to perform routine, non-administrative tasks. One reason is that viruses are typically introduced by routine tasks such as browsing the Internet and reading e-mail. Viruses initiated with system administrator level privileges have the potential to cause more damage than those initiated from a regular user account. This separation of duties helps ensure that if a user is compromised, the impact is minimized by the limited privileges held by that user. This practice also facilitates log analysis and review because a great deal of irrelevant information is effectively removed from any review of system administrator activities. In a Unix environment restricting direct login access to the powerful root command and mandating the use of SU (switch user) will support this recommendation.

As indicated earlier it is very difficult to implement restrictions on a system administrator account. Properly implemented, logging and auditing can be another mitigation option. Part of the implementation requires that the functions of system administration and audit are separated and given to different people. In addition, access to the logs used for audit must also be restricted to ensure the integrity. In this situation the system administrator would not have the responsibility to review the logs, or have ability to alter or delete the logs. The ISO17799 Standard recognized the requirement for this specific separation of duties in the recommendations for access control.

9.7.2.3 Logging and reviewing events

> When allocating the responsibility for log review, a separation of roles should be considered between the person(s) undertaking the review and those whose activities are being monitored. [8]

In addition to the system administrator function, and the principle of least privilege is often applied to the role of developers. The emphasis is on separating the responsibilities associated with development, testing and production support. Most organizations recognize the requirement to separate the production and development environments and typically implement physically separate platforms and separate directories or libraries to support this. In most cases the two environments have different modes of operation, and the separation is recommended because the practices associated with development do not mesh well with the requirement to preserve integrity and availability in a production environment. The objective of separating the development and testing function is designed to improve the quality and integrity of changes. The recommendation to separate the production support and development function supports the concept of least privilege in the production environment and helps ensure proper change control.

In the ISO17799 Standard, the operations management control area recommends the implementation of segregation of duties and separation of environments as part of the mitigation for intentional or accidental misuse of systems or information. These best practices reduce the potential for one individual to perpetrate fraud or introduce malicious code and helps perverse the confidentiality, integrity and availability of production systems and information.

Least Privilege and Process

The principle of least privilege is at the same time, dependant on and supportive of other information security management control processes. To properly implement 'least privilege' at your organization it is important to recognize that other control processes must be implemented and maintained.

Preventative control processes, which include access control and authentication, have been described as 'methods, tools, practices and techniques used to ensure that systems remain secure and highly available'. [7] Authentication and the resultant individual identification are used by most access control mechanisms to link activities on a system to specific individuals and thereby establish accountability. Various authentication mechanisms can be used, but two common criteria are necessary to support 'least privilege'. First, the authentication procedure must be strong enough to provide assurance and that a specific identification (user id) is only available to the anticipated end-user. Obviously two factor authentication provides a greater level of assurance than a single factor authentication (userid and password). However, userid and password can also provide a reasonable level of assurance as long as criteria such as strong passwords, password expiry, policy prohibiting password sharing,

and encryption of passwords for storage and transit are respected. Second, there must be singular ownership of system identifiers (userid) so that activities associated with a userid could be traced back to one individual. It would not be possible to implement any restriction or granularity for access to information or processes, if there was not a reliable authentication process in place.

In order to ensure that access controls adequately protect all of the organization's resources, it will be necessary to properly categorize the resources. This is an important step that must be done as early as possible, and would be a collaborative effort between the identified information owners and the Information Technology Department. The categorization of resources can involve determining the appropriate classification of the information or data, and providing a clear definition and delineation of roles and responsibilities associated with the application. This is not an insignificant effort and it requires the active participation of the business units of an enterprise, since they are typically the information owners. Once the internal structure is established and documented it is possible to institute a consistent system for assigning access rights to those data resources or system functions

Certainly there are other key control processes and best practices that help support 'least privilege'. Change management processes and auditing for compliance also help support 'least privilege'. However the preventative control processes of access control and authentication are the most important ones. Within the access control process, the organizational commitment to information classification, and proper delineation of roles and responsibilities for the information system, are the most important requirements for 'least privilege'.

Least Privilege and Technology

Applying the principle of least privilege to today's technology involves establishing and maintaining zones of control and utilizing a number of methods to limit or disable services and accesses. These practices can be applied successfully to infrastructure, operating systems and development standards to strengthen the security posture of your enterprise. Applying least privilege to today's technology, particularly the web architecture, is important because it will limit the organization's exposure to a variety of threats and the damage they can cause.

Today most organizations leverage the Internet or web technologies as part of their day-to-day business. There are inherent risks associated with exposing valuable corporate systems to this new type of end-users. Similar to any user group in your organization, the access and privileges for Internet users must also be defined and minimized as much as possible. To that end technology solutions such as segmentation of networks, defining zones, firewalls, and removing services must be implemented in your web architecture.

The first practical implementation of 'least privilege' in your web architecture involves establishing segmentation between various parts of your network. For most organizations this segmentation initially involves establishing three zones; the internal network, a buffer zone typically called a demilitarized zone, and the outside network or Internet. The main purpose of separating your network infrastructure into zones is to establish boundaries to facilitate control limitations. Access requirements, applications and offered services will be different in each one of these traditional zones. Firewalls, and in some cases authentication mechanisms, will be used to control access between the zones.

There are basic rules and best practices that are applied in establishing the demilitarized zone. If only one demilitarized zone (DMZ) has been established then it should contain the public accessible systems, such as the web servers and the mail servers. In a multi-tier or e-commerce application, the application and database servers should not be in the one demilitarized zone (DMZ). In this traditional 'three zone' example, the application and database servers are contained in the internal network. The benefit of this design is that the scope of any attack from the external network or Internet can be limited to the systems in the publicly accessible DMZ. The design supports 'least privilege' because it will allow unauthorised and unexpected service requests to be easily controlled and blocked. In this example Internet users would only be allowed to interact with the web server in the DMZ, because that would be the only other zone that they could access.

In this particular example there is an implied separation of services. There are a variety of reasons why it is recommended to separate services, but the two most commonly cited reasons are performance and security. From a security perspective, the separation of services facilitates the practice of limiting access and limits the potential for damage if one service is compromised. Isolating services to separate and dedicated host computers is a recommended best practice that extends the protection introduced with network segmentation. In this example the web server, which is available to the Internet community, is separated from the confidential information of the organization, which is on the database server. The levels of access for the two platforms are very different. For example the web server may be available to the entire Internet and allow access without authentication, the database server would likely only support connection from a specific host platform, and require authentication. Implementing appropriate access control and protection for these two different services is facilitated if the services are separated. The separation of services across two different zones and three different host platforms effectively creates multiple conditions required for the 'separation of privilege' principle.

Firewalls are designed to control access. Firewalls are a combination of hardware and software that acts as a gateway between two networks or zones and can be configured with a set of rules to allow or deny communications. A properly configured firewall in a properly segmented network is tailor made to

implement ‘least privilege’. Implementing ‘least privilege’ in this case means that only the smallest set of services required for an application is allowed to pass or traverse the firewall. That limitation is sometimes referred to a ‘permitted services’ or ‘user permit rules’. Allowing only HTTP or HTTPS to pass through the firewall to the public web server would be an example of implementing ‘least privilege’ with a firewall. Firewalls are also capable of implementing a form of access control by restricting services to particular systems and users. Packet based firewalls will filter packets based protocol, IP addresses of the sender and destination, and source destination ports. [18] In the simplest form the firewall access rule restricts types of network outbound traffic, and restricts inbound destinations when packets match selected rules. When a match is found, the firewall will take action and either allow or deny the packet to pass. In this simple ‘three zone’ example, an access rule could be established so that only the web server in the DMZ is allowed to communicate with the application server. In another example, most guidelines for Firewall configuration will recommend that the port number 53(TCP) be restricted to know secondary domain name servers, to so that intruders cannot learn about systems connected on the internal network. [19] Another rule could be established that would only allow the Webmaster’s workstation access to the web server in the DMZ from the internal systems. The concept of ‘zero access’, or ‘forbidden unless expressly permitted’ should always be applied to Firewall configuration.

To support ‘least privilege’ and separation the following general rules should be applied when configuring a firewall:

- Control connections from the Internet to the DMZ by only allowing protocols that are needed by the applications or services that are being offered (i.e. HTTP, HTTPS, perhaps SMTP if there is a mail server)
- Only allow Internet connections to the DMZ, by implementing a ‘deny all’ approach if connections to other zones are attempted.
- Control connections from the DMZ to the internal network by restricting those connections to the particular hosts, and only allowing protocols that are needed by the applications or services that are being offered. For example the web server in the DMZ can only initiate connections to the application server and is limited to the expected communication protocol.
- Control connections from the internal network to the DMZ by restricting those connections to the particular systems used to perform administrative functions and only allowing administrative protocols.

Firewall implementation and configuration is perhaps the most representative example of implementing ‘least privilege’ that exists in information technology. It is interesting to note that most organizations will accept the requirement and best practices of firewalls, which includes ‘least privilege’ and separation principles, but resist applying similar protections internally.

Minimizing and reducing unnecessary services is another important and very obvious application of ‘least privilege’. This application of ‘least privilege’ can be

applied to different components of your enterprise, such as servers, desktop operating systems and installed software. For servers that are publicly accessible or part of an organization's web architecture, the exercise to identify and remove unnecessary programs and services is often part of a larger exercise called 'server hardening'. This practice is operating system independent and offers two significant benefits, in that it reduces complexity of the system that is being supported and makes the system more secure by closing off possible avenues of attack. Understanding what programs and services are required takes some up front effort on the part of the implementation team, to identify which ones are not absolutely necessary for your server's everyday performance. However understanding which ones are unnecessary should facilitate future efforts related to patch management, because only the necessary programs and services will be present and need to be kept up to date. Given the numerous security advisories and vulnerabilities that are identified, and the variety of software that must be supported in today's enterprise, this upfront effort to reduce complexity could well save time in the long run.

Operating systems come with default software configurations that emphasize features, functions, and ease of use, at the expense of security. In addition to offering these features and functions, vendors try to facilitate the implementation and activation, by making many of them part of the default installation. However each of the new features and functions added to a host increases the risk of compromise for that host, because each service is another possible avenue of access for an attacker. For example the default installation of Windows 2000 server will install Internet Information Server (IIS) 5.0. This default installation also creates a Web server and an FTP server, and in their default state both are wide open for attack. It is interesting to note that Unix vulnerabilities identified in the SANS/FBI Twenty Most Critical Internet Security Vulnerabilities List [13] are for services that could be disabled on most systems in an enterprise. For example, nearly all versions of Unix allow the sendmail program, the email service for Unix servers, to be installed. The sendmail program, particularly older versions of the program, has known vulnerabilities that can be exploited. This is an unnecessary exposure because the sendmail daemon is not really required for any servers or workstation installations that are not identified as the organization's mail server or mail relay. In general, if services are not required then the minimalist approach associated with 'least privilege' is preferable and the services should be removed. If these services are implemented then there will be an immediate requirement for patch updates and re-configuration of these services. The group responsible for system support at your enterprise has a better understanding of the organization's security needs than the vendor, so a default installation should not be trusted. All new servers should reflect their organization's security requirements and the server configuration should be revisited as requirements change.

The specific exercise of identifying and removing unnecessary programs and services is variable depending on the base operating system, the function and

purpose the host platform and the security needs of your organization. Specific recommendations are beyond the scope of this paper but many operating system and function specific guideline papers exist on the Internet.

Another aspect of 'server hardening' that relates to 'least privilege' is the removal of unnecessary default accounts and groups. In some cases the default configuration of the operating system often includes a variety of accounts that are well known, and have known password defaults (i.e. guest account in Windows 2000). These accounts should be reviewed, and removed or disabled to eliminate their use, particularly by unwelcome intruders. If there is a requirement to retain a default account then the account should be obscured by renaming it, if possible, and the password should be changed.

Implementing 'least privilege' on servers that are publicly accessible, or part of an organization's web architecture, is important and should have a higher priority. They are available to a wider audience on the Internet that can include true end-users, as well as hackers. One of the goals is to provide appropriate services and accesses for the true end-user while removing possible entry points for unwanted intruders or hackers. The same principle can and should be applied to internal servers as well as desktop operating systems, to achieve a more secure enterprise.

Implementing security, and particularly 'least privilege' offers a unique challenge in today's modern operating systems as a result of the prevalence of malicious code, and a design weakness in the operating systems. A National Security Agency paper titled ' The Inevitability of Failure: The Flawed Assumption of Security in Modern Computing Environment' describes the inevitability of malicious code.

> Substantial increases in connectivity and data sharing have increased the risk to systems such that even a careful and knowledgeable user running on a single-user system is no longer safe from the threat of malicious code. Because the distinction between data and code is vanishing, malicious code may be introduced, without a conscious decision on the part of a user to install executable code, whenever data is imported into the system. [14]

In addition, the paper goes on to suggest that 'mainstream commercial operating systems rarely support the principle of least privilege even in their discretionary access control architecture'. [14] The design weakness in modern operating systems, which violates the principle of least privilege, may be best exampled by the evolution and current state of the Windows 2000 operating system. In an effort to continually provide more and better features to the consumer, some basic principles such as the principle of least privilege are sacrificed. For example to improve performance the concept of isolating services, which was a

primary design principle from Saltzer and Schroeder is set aside in Windows 2000. One representative example is described in the following text:

> Subsystems are no longer isolated from each other to avoid expensive context switches during execution. For example, the graphics subsystem of Windows is largely contained within the kernel's address space (!) to reduce the cost of invoking common drawing routines. [2]

This combined with the reality that most desktop operating systems are configured to run in the context of a single user, means that any code or application that is invoked, runs with full privileges of that user. Two infamous viruses demonstrated this vulnerability: the Melissa Virus in 1999 and the Love Bug Virus in 2000. These mobile code viruses were identified as the first and second fastest spreading viruses as of May 2000 [15]. In the case of the Melissa Virus a script contained in an email message executed and allowed the viruses to propagate. The script would be invoked by the mail-viewer when the message was opened. In this case the virus is invoked at the application level, Microsoft Outlook, and as a result the operating system kernel had no opportunity to take control. In this situation the script ran in the context of a single user who received the message and was allowed to read the address book and forward copies of a message. The Love Bug virus utilized a Visual Basic script and Outlook mail to execute and spread in a similar fashion. Unfortunately, as these examples attest it is very possible for users to unknowingly or intentionally execute malicious code in today's modern operating systems.

The failure of modern operating systems to implement the principle of least privilege has spawned a number of different research initiatives to develop a more secure model. Unfortunately, these models do not offer an immediate solution to an enterprise that supports modern operating systems, such as Windows 2000. In that case re-enforcing some other 'defence in depth' principles and practices such as separation of duties, employee awareness, vigilant virus defence and operating system maintenance can help mitigate the weakness in today's operating systems.

The concern that malicious code can run with all the privileges of a single user is magnified if that single user context was someone with Windows 2000 Administrator privileges. As part of Windows 2000 Server Documentation, Microsoft recommends the practice of separating the administrator and user functions. The recommendation is that administrator level access be used only when required, and that most of the common activities, such as e-mail, Internet surfing and legacy applications be done from an account that is part of the default Users or Power Users group. Since the Administrator role would have full control of the operating system (i.e. the ability to install system components, perform upgrades, etc) it should not be used for common activities to avoid the potential of having malicious code run with extra privileges. This would necessitate a second account for those charged with Windows 2000

Administrator functions and those individuals would effectively implement a 'separation of duties' within their own job functions. Microsoft facilitates this requirement with the 'run as' feature. The Windows 2000 'runas' feature allows a person charged with system administrator responsibilities to switch to the privileged account Administrator context without a logoff login process. For example:

> To start an instance of the Windows 2000 command prompt as an administrator on the local computer, type:
> runas /user:*localmachinename*\administrator cmd
> When prompted, type the administrator password. [16]

A second mitigation strategy, which is always a good practice, would be to improve employee awareness about this issue for staff in general, and for the Information Technology group in particular. Educating staff on the potential for code exploitation because of an inherent weakness in today's operating systems would, at the very least, give them a greater appreciation for need for virus defence. Educating support staff with administration responsibilities should increase the level of attention and caution exercised by that key group. As well, it will provide a rationale to facilitate the implementation and adoption of the 'separation of duties' concept, which would separate administrator and user job functions across two accounts.

The third and fourth mitigation strategies in this area would be maintaining current virus defence and keeping products current in terms of service maintenance. Virus defence and product currency are part of the 'defence in depth' approach to information security. In many cases new viruses take advantage of old or known vulnerabilities in desktop operating systems. Keeping all software product suites, but particularly the desktop operating systems, as current as possible will in itself close down a large number of viruses because the vulnerability will be removed. As indicated previously even security conscious end-users can be victims of malicious code, so keeping your enterprise virus defence current is very important. A gateway e-mail scanning solution, desktop virus scanning software and current virus definition signature files combine to provide a comprehensive virus defence.

Applying the principle of least privilege to the development standards at your enterprise represents another best practice that will strengthen the security posture of your enterprise. The principle could be restated for developers in the following fashion: programs should be designed to operate with the least amount of privilege possible to accomplish the intended function. In order to achieve that objective it is important that developers adopt the habit of using lesser-privileged accounts during the development process. The personal account for a developer typically has elevated privileges, particularly for their desktop operating system or in the development environment. This is to allow the developer some extra autonomy for such things as software installations or other administrator

activities. In this situation the developer's elevated privilege status is not representative of minimum requirement for the end-user in production. As a result it is important that the developer have access to, and make use of, a regular lesser-privileged account during development. In creating the lesser-privileged account or test account the concept of 'zero-access' to start should be applied. In most cases the 'zero access' start point in terms of privilege will equate to the enterprise's standard account default. Any additional privilege or access requirements can be recorded as they are applied to this test account in the development environment. When the development package is promoted to the next environment (i.e. quality assurance or test) the accumulation of access and privilege requirements are readily available, and can easily be quantified for this part of the life cycle. This requires some additional effort on the part of the developer but there is some assistance available. In an online publication, entitled 'How to develop code as a non-admin', Keith Brown offers some tips on how to facilitate the co-existence of the two types of accounts in Windows development setting.

http://www.develop.com/kbrown/book/html/howto_runasnonadmin.html

The first and most important step towards applying the principle of least privilege to the development standards is to realize that the best way to build software that can be run by non-privileged users is to run as a non-privileged user while you write the code.

Conclusion

'Least privilege' is a foundation principle for information security. It needs to be applied to all aspects of information security at your enterprise. Since proper application is so widespread and diverse, many different parts of the organization can make or break its application. The principle of least privilege has a direct dependence on other foundation principles, as well as, a dependence on well-accepted control processes. Proper application of the principle requires an initial investment and an ongoing discipline on the part of all management and staff at an organization. The examples of modern operating systems demonstrate that the responsibility even extends to vendors outside the organization.

It is difficult to identify one area of the application of 'least privilege' as more important than another. Depending on the business requirements of your organization strict adherence to 'least privilege' in a web architecture design, or establishing least privilege through separation of responsibilities for financial applications, or control processes for system administrators may seem to have more importance than something else. In this reality no one aspect of implementing 'least privilege' is really any more important than another because a weakness in one area can effectively compromise the best efforts in another area. In order to achieve acceptance of this principle or any other information security principle, it is essential to raise awareness and understanding. This paper provided some background, offered some rationale to help develop

support for it's acceptance, and identified ways 'least privilege' can be implemented at your enterprise.

References

[1] SCHNEDIER, Fred B. Cornell University. 'Least Privilege and More'. URL: http://www.cs.cornell.edu/fbs/publications/leastPriv.pdf

[1] SALTZER, J.H. and SCHROEDER, M.D. 'The Protection of information in computer systems in computer systems', Proceeding of IEEE, vol. 63, no.9 (Sept. 1975), pp. 1278-1308

[2] SCHNEDIER, Fred B., MORRISETT, Greg, and HARPER, Robert. 'A Language-Based Approach to Security' URL: http://www-2.cs.cmu.edu/~rwh/papers/langsec/dagstuhl.pdf

[3] Department of Defense - Trusted Computer System Evaluation Criteria (TCSEC) DOD 5200.28-STD (DECEMBER 1985). URL: http://www.radium.ncsc.mil/tpep/library/rainbow/5200.28-STD.html

[4] McGraw, Gary and Viega, John. IBM developerWorks, 'Software Security Principles: Part 3' (October 2000) URL: http://www-106.ibm.com/developerworks/security/library/s-priv.html?dwzone=security

[5] SALTZER, J.H. and SCHROEDER, M.D. 'Basic Principles of Information Protection' (1975) URL: http://web.mit.edu/Saltzer/www/publications/protection/Basic.html

[6] Miles, Tracy, Jansen, Wayne, and McLamon, Mark. National Institute of Standards and Technology (NIST) 'Guidelines on Securing Public Web Servers' (September 2002) URL: http://csrc.nist.gov/publications/nistpubs/800-44/sp800-44.pdf

[7] SANS Institute, 'SANS Online Security Essentials Course Track 1 (2003)' Section 1.4.6 Operations Security

[8] International Standard ISO/IEC 17799:2000(E), 'Information Technology – Code of Practice for information security management' Switzerland: ISO, 1st Edition 2000-12-01

[9] PricewaterhouseCoopers on behalf of the UK Department of Trade and Industry, 'Information Security Breaches Survey 2002 – Figure 22' http://www.pwcglobal.com/Extweb/ncsurvres.nsf/0cc1191c627d157d8525650600609c03/845a49566045759e80256b9d003a4773/$FILE/ATT7PG80/DTI%20Security%20Survey%202002.pdf

[10] PricewaterhouseCoopers on behalf of the UK Department of Trade and Industry, 'Information Security Breaches Survey 2002 – Figure 52' http://www.pwcglobal.com/Extweb/ncsurvres.nsf/0cc1191c627d157d8525650600609c03/845a49566045759e80256b9d003a4773/$FILE/ATT7PG80/DTI%20Security%20Survey%202002.pdf

[11] Richardson, Robert. Computer Security Institute, 'Eighth Annual 2003 CSI/FBI Computer Crime and Security Survey' CSI (2003)

[12] Frisch, AEleen., 'Essential System Administration 2nd Edition', Sebastopol: O'Reilly & Associates (1995)

[13] SANS Institute, 'SANS/FBI Top 20 The Twenty Most Critical Internet Security Vulnerabilities' Version 3.23 May 29, 2003 URL: http://www.sans.org/top20/

[14] Loscocco, Peter A et al., National Security Agency, 'The Inevitability of Failure: The Flawed Assumption of Security in Modern Computing Environments', Proceedings of the 21st National Information Systems Security Conference, pages 303-314, October 1998 http://www.nsa.gov/selinux/doc/inevitability.pdf

[15] McGraw, G. and Morrisett, G., Infosec Research Council, 'Attacking Malicious Code: A report to the Infosec Research Council' (May 2000) URL : http://www.cigital.com/irc/malicious_code.pdf

[16] Microsoft Online Documentation. 'Runas command description' URL: http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/windowsserver2003/proddocs/standard/runas.asp

[17] Bishop, Matt, 'Computer Security: Art and Science' Addison-Wesley Professional, 2002 - Chapter 13 Design Principles http://www.aw-bc.com/samplechapter/0201440997.pdf

[18] Wack, John, Cutler, Ken, and Pole, Jamie, National Institute of Standards and Technology (NIST) Special Publication SP 800-41, ' Guidelines on Firewalls and Firewall Policy (January 2002) – Appendix C' URL: http://csrc.nist.gov/publications/nistpubs/800-41/sp800-41.pdf

[19] CERT Coordination Center, 'Packet Filtering for Firewall Systems' Last Revision February 2002 URL: http://www.cert.org/tech_tips/packet_filtering.html

[20] Microsoft Online Documentation. 'Microsoft Windows 2000 Security Common Criteria Secure Configuration Guide - Appendix D User and Group

Accounts' Microsoft Technet Security. URL:
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/issues/W2kCCSCG/W2kSCGcd.asp

Other References

Swanson, M. and Guttman, B., National Institute of Standards and Technology (NIST) Special Publication SP800-14 'Generally Accepted Principles and Practices for Securing Information Technology Systems'
http://csrc.nist.gov/publications/nistpubs/800-14/800-14.pdf

Stoneburner, G., Hayden, C., and Feringa, A., National Institute of Standards and Technology (NIST) Special Publication SP880-27, 'Engineering Principles for Information Technology Security (EP-ITS)' URL:
http://csrc.nist.gov/publications/nistpubs/800-27/sp800-27.pdf

Sokol, Marc S. & Curry, David A. Internet Security Systems, 'Security Architecture and Incident Management for E-Business' (2000) URL:
http://documents.iss.net/whitepapers/secarch.pdf

SANS Institute, 'SANS Online Security Essentials Course Track 1', (2003)

Birkholz, Erik Pace. Foundstone Inc., 'Special Ops Host and Network Security for Microsoft, Unix and Oracle', Rockland: Syngress Publishing Inc. 2003

Naidu, Krishni. 'Security Consensus Operational Readiness Evaluation - Firewall Checklist V1.0' URL:
http://www.sans.org/score/checklists/FirewallChecklist.doc

Garfinkel, Simson, Spafford, Gene, 'Web Security, Privacy and Commerce 2nd Edition', Sebastopol: O'Reilly & Associates Inc, 2002

Vossen, J.P., 'Securing (Hardening) Windows Servers'
Revised January 22, 2002 URL:
http://www.speakeasy.org/~vossenjp/Hardening_Windows_Servers.pdf

Parmar, S.K. "An Introduction to Security - Security Manual" URL:
http://downloads.securityfocus.com/library/index.html

Microsoft Online Documentation. 'Microsoft Windows 2000 Server Online Documentation' URL:
http://www.microsoft.com/windows2000/en/server/help/default.htm?id=0

Microsoft Online Documentation. 'Microsoft Windows 2000 Resource Kit - Privileges and Attack Prevention' URL:

http://www.microsoft.com/windows2000/techinfo/reskit/en-us/default.asp?url=/windows2000/techinfo/reskit/en-us/iisbook/c09_privileges_and_attack_prevention.asp

Microsoft Online Documentation. 'Microsoft Windows 2000 Resource Kit – IIS 5.0 Resource Guide – Security' URL: http://www.microsoft.com/windows2000/techinfo/reskit/en-us/default.asp?url=/windows2000/techinfo/reskit/en-us/iisbook/c09_security.asp

Brown, Keith. 'A .NET Developer's Guide to Windows Security – Item 1: What is a non-privileged user?' URL: http://www.develop.com/kbrown/book/html/whatis_anonprivilegeduser.html

Brown, Keith. 'A .NET Developer's Guide to Windows Security – Item 2: How to develop code as a non-admin' URL: http://www.develop.com/kbrown/book/html/howto_runasnonadmin.html