

A Framework for Measuring Hardening Capabilities of an Operating System

Thesis submitted in partial fulfillment of the requirements for the award
of degree of

Master of Engineering
in
Software Engineering



By:
Yugma Kumar Agrawal
(8053124)

Under the supervision of
Dr. Maninder Singh
Assistant Professor

MAY 2007

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

Certificate

I hereby certify that the work which is being presented in the thesis entitled, “**A Framework for Measuring Hardening Capabilities of Operating System**”, in partial fulfillment of the requirements for the award of degree of Master of Engineering in Software Engineering submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my research work carried out under the supervision of Dr. Maninder Singh.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

(Yugma Kumar Agrawal)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr. Maninder Singh)
Supervisor
Computer Science and Engineering Department
Thapar University
Patiala

Countersigned by

(Dr.(Mrs.) Seema Bawa)
Professor & Head
Computer Science & Engineering Department
Thapar University
Patiala

(Dr. R.K. Sharma)
Dean, Academic Affairs
Thapar University
Patiala

Acknowledgement

At the end of my thesis I would like to thank all those people who made this thesis possible and an enjoyable experience for me.

In the first place I would like to record my gratitude to my advisor, **Dr. Maninder Singh**, Assistant Professor, Computer Science & Engineering Department, Thapar University, Patiala for his supervision, advice, and guidance from the very early stage of this research as well as giving me extraordinary experiences through out the work. Above all and the most needed, he provided me unflinching encouragement and support in various ways. I am indebted to him more than he knows.

I also take the opportunity to thank **Dr Seema Bawa**, Professor & Head, Computer Science and Engineering Department, Thapar University, Patiala, and the entire faculty and staff for their help, inspiration and moral support, which went a long way in successful completion of my Thesis.

Collective and individual acknowledgments are owed to my colleagues whose presence is somehow perpetually refreshed, helpful, and memorable. Many thanks go in particular to Manmeet Singh, Parvinder Singh, Harpreet Kaur Bindra and Shailendra Sharma, for being great friends and giving me such a pleasant time when working together.

I am also thankful to the authors whose work I have consulted and quoted in this work.

Yugma Kumar Agrawal
(8053124)

Abstract

Today, computers are used in all walks of life: they are in homes, in various critical domains such as defense, education, finance, government, health care, and so on. This reliance of the world's infrastructure on computer systems, and the consequent pervasiveness of the latter, makes their "security" an issue of great importance. Communication networks are used to transfer valuable and confidential information for a variety of purposes. As a consequence, they also attract the attention of people who intend to steal or misuse the information, or to destroy the systems storing and communicating it.

Attackers break into a system by exploiting its weakness. These weaknesses may come from programming flaws or insecure configuration of the application programs or operating system. As the operating system is an essential part on any information system any vulnerability in it can make the whole system vulnerable. Most popular operating systems software in use today, were designed with some Information Assurance in mind. Many of them have sufficient security features and capabilities. However; their default configuration is wide-open with most security features not-enabled when installed in operational environments. This leaves the applications and networks using these operating systems exposed to attacks by computer hackers and unauthorized access to their resources. Proper configuration of the security features eliminates the basic vulnerabilities of the operating system and makes the whole information system more secure.

The idea of operating system hardening is to minimize a computer's exposure to current and future threats by configuring the security features of the operating system and removing unnecessary applications. Hardening of an operating system involves the removal of all non essential tools, utilities and other systems administration options, any of which could be used to ease a hacker's path to the systems. It also involves implementing the latest operating system patches and updates from the vendors. Following this, the hardening process will ensure that all appropriate security

features are activated and configured correctly. Countermeasures like firewalls or anti-anything (antivirus, anti-spam, anti-spyware, etc.) are all reactive security tools. They are necessary countermeasures and a part of a comprehensive security system, but proactive actions, such as daily vigilance and operating system hardening ensures the highest level of network security. By hardening operating systems, security risks are significantly reduced, in turn reducing the inconvenience and cost involved in downtime and system rebuilds, and also damage to the organization's reputation.

Different operating systems have different security features and thus different hardening capability. As the operating system represents the core of an information system it must be selected such that it fulfils the security requirements of the whole information system. Thus in the selection of the operating system the hardening capability present in it plays a crucial role. As there is large number of security features present in various operating systems, it will be quite difficult for an administrator to measure the hardening capability of the operating system qualitatively. Hence a method of quantitatively measuring the hardening capability of the operating system will be quite helpful in proper selection of the operating system.

This work presents a scoring scheme to systematically quantify the hardening capability present in an operating system. The scoring is based on a list of proposed security features for an operating system. The existence and appropriate setting of the security features increases the robustness of the operating system. All the security features are grouped into seven categories: Authentication, Network Access and Communication, System Right and Access Controls, Services, Data Access and File System, Patch Management, System Auditing

Each security feature is given a severity score by considering how much damage to the system can be prevented if that security feature is present in the operating system. A missing security feature provides the attacker an opportunity to violate the security of the system. This violation can be classified into four types: confidentiality violation, integrity violation, availability violation, and system compromised.

The hardening capability of an operating system is calculated according to the presence or absence of the security feature in the operating system. Scoring is done for each of the seven security hardening feature categories. If the security feature is present in operating system then its score is added to the score of its category, otherwise is it subtracted. To provide a concise view of the hardening capability an overall hardening capability score is also calculated.

For calculating the hardening capability score security hardening features in the operating system are matched with proposed security hardening features. Matching the natural language specification of the security features can result in incorrect matching due to the inherent ambiguity in natural language specification. To facilitate high-assurance hardening capability score, the proposed and operating system's security hardening features are formally specified. A formal specification enables to describe security features clearly and precisely. It also could benefit us in automated calculation of hardening capability score on formal specifications.

The windows 2000 professional and windows XP SP2 operating systems, due to there large deployment, are used to build a hardening capability profile to exemplify the scheme usefulness.

Table of Contents

Certificate	i
Acknowledgement	ii
Abstract	iii
Table of Contents	vi
List of Tables	xi
Chapter 1 Introduction	1
1.1 Security	
1.1.1 Integrity	2
1.1.2 Confidentiality	3
1.1.3 Availability	4
1.2 System Hardening	5
1.3 Operating System Hardening	7
1.3.1 Install Offline	8
1.3.2 Lock Down the Operating System	9
1.3.3 Lock Down the Services	9
1.3.4 Establish a Baseline	10
Chapter 2 Literature Survey	11
2.1 What is Operating System Hardening	11
2.2 Existing Operating System Hardening Methods	12
2.2.1 Center for Internet Security Scoring Tools	12
2.2.2 The Bastille Hardening Program	14
2.3 Hardening Windows XP	16
2.4 Hardening Linux	27
Chapter 3 Problem Statement	35
Chapter 4 Proposed Security Hardening Features	36
4.1 Proposed Security Hardening Features Categories	36

4.1.1 Authentication	36
4.1.2 Network Access and Communication	37
4.1.3 System Rights and Access Controls	38
4.1.4 Services	39
4.1.5 Data Access and File System	40
4.1.6 Patch management	40
4.1.7 System Auditing	41
4.2 Security Hardening Features	43
4.2.1 Authentication Hardening Features	43
4.2.1.1 Maximum password age	43
4.2.1.2 Minimum password age	44
4.2.1.3 Minimum password length	44
4.2.1.4 Enforce password history	46
4.2.1.5 Passwords must meet complexity requirements	47
4.2.1.6 Store password using reversible encryption for all users	48
4.2.1.7 Allow changing passwords only when required	48
4.2.1.8 Account lockout duration	49
4.2.1.9 Account lockout threshold	50
4.2.1.10 Reset account lockout counter after	51
4.2.1.11 Use of Blank Passwords to Console Logon	51
4.2.1.12 Machine account password changes	52
4.2.1.13 Do not display last user name	53
4.2.1.14 Number of previous logons to cache	54
4.2.1.15 Require Domain Controller authentication to unlock workstation	54
4.2.1.16 Do not allow storage of credentials or .NET Passports for network authentication	55
4.2.1.17 Pluggable different authentication for different Services	56
4.2.2 System Right and Access Controls Hardening Features	57
4.2.2.1 Load and unload device drivers	57
4.2.2.2 Change the system time	57
4.2.2.3 Manage auditing and security log	58

4.2.2.4 Log on as a service	59
4.2.2.5 Force shutdown from a remote system	59
4.2.2.6 Act as part of the operating system	60
4.2.2.7 Adjust memory quotas for a process	60
4.2.2.8 Back up files and directories	61
4.2.2.9 Debug programs	61
4.2.2.10 Access to this computer from the network	62
4.2.2.11 Lock pages in memory	63
4.2.2.12 Modify firmware environment values	63
4.2.2.13 Increase scheduling priority	63
4.2.2.14 Perform volume maintenance tasks	64
4.2.2.15 Profile single process	64
4.2.2.16 Profile system performance	65
4.2.2.17 Restore files and directories	65
4.2.2.18 Take ownership of files or other objects	66
4.2.2.19 Allowed to format and eject removable media	66
4.2.2.20 Restriction Policy for Software	67
4.2.2.21 Create a token object	68
4.2.3 Network Access and Communication Hardening Features	68
4.2.3.1 Digitally sign communications (four related settings)	68
4.2.3.2 Send unencrypted password to third-party SMB Servers	70
4.2.3.3 Do not allow anonymous enumeration of SAM accounts	71
4.2.3.4 Digitally encrypt or sign secure channel data (multiple related settings)	71
4.2.3.5 Strong key protection for user keys stored on the computer	72
4.2.3.6 Use FIPS compliant algorithms for encryption, hashing, and signing	73
4.2.3.7 Shares that can be accessed anonymously	73
4.2.3.8 Force logoff when logon hours expire	74
4.2.3.9 Disable remote Desktop Sharing	74

4.2.3.10 Make proxy settings per-machine (rather than per-user)	75
4.2.3.11 Do not allow users to add/delete sites	75
4.2.3.12 Do not allow users to change policies	76
4.2.3.13 Allow software to run or install even if the signature is invalid	77
4.2.3.14 Allow active content from CDs to run on user Machines	77
4.2.3.15 Allow third-party browser extensions	78
4.2.3.16 Check for server certificate revocation	78
4.2.3.17 Check for signatures on downloaded programs	79
4.2.3.18 Do not save encrypted pages to disk	79
4.2.3.19 Empty Temporary Internet Files folder when browser is closed	80
4.2.3.20 Consistent MIME Handling	80
4.2.3.21 Domain member: Require strong (Windows 2000 or later) session key	81
4.2.3.22 Wireless Networking Security	82
Chapter 5 Design Using Formal Specification	83
5.1 Formal Specification of Security Feature.	83
5.2 Security Hardening Feature Matching	85
5.3 Signature Match	86
5.4 Property Match	87
5.5 Algorithms for Matching the Security Hardening Features	88
5.5.1 Algorithm 1 signature match (LSIG, TSIG)	88
5.5.2 Algorithm 2 construct-name-tables (LSIG,TSIG)	90
5.5.3 Algorithm 3 property-match (LP, TP, nametable)	91
5.5.4 Algorithm 4 declaration-match (LD, TD)	92
5.5 Measure of Similarity between Specifications	93
Chapter 6 Implementation of the Proposed Framework	97
6.1 Severity Score of Security Hardening Features	98

6.2 Hardening Capability Scoring Technique	103
6.2.1 Scoring by Hardening Option Categories	104
6.2.2 Overall System Scoring	105
6.3 Hardening Capability Scoring	10
Chapter 7 Conclusion and Future Work	110
7.1 Conclusion	110
7.2 Future Work	111
References	113
List of Papers Published	115

List of Tables

Table 6.1. Description of damages by Types and Severity	98
Table 6.2 Security Feature Severity Score	100
Table 6.3 Security Feature Categories Weights	105
Table 6.4 Hardening Capability Score by Security Feature Categories	106
Table 6.5 Final Hardening Capability Score by Security Feature Categories	109
Table 6.6 Overall Hardening Capability Comparison	109

1.1 Security

Security is an important area in today's IT-related headlines. Frequent system vulnerabilities and security patches as well as viruses and worms are common ground for everyone giving primary consideration to computer security. As nearly every computer system is interconnected to other computers or to the Internet, securing these computers is critical to mitigate break-ins, data theft or loss, misuse, or even liabilities to third parties.

Even securing a stand-alone computer, one not connected to a network, is not trivial. Applications have to be installed from trustworthy sources, such as from a verified and virus-checked CD-ROM. Administrators have to be equally cautious with application data. For example, software packages (office suites, etc.) can execute powerful macro languages or present malformed data and can be used to execute arbitrary code by exploiting software flaws. Therefore, application data has to be checked for integrity before they are copied it to the computer. Access to the system can be controlled by placing the data in a secured area (disregarding attacks from authorized personnel, of course).

Things get even more difficult when a system is connected to a network and offers services to other computers – intentionally or unintentionally. In that case, the system administrator might not be the only source of data, as client programs make use of the offered services, and system vulnerabilities might allow an intruder to take control of the computer.

That's why dealing with security is essential throughout the entire life cycle of a system, from the planning stages until it is dismantled.

In general, data security and system security can be separately distinguished. Data security is commonly understood to refer to all efforts taken to ensure:

- Confidentiality
- Integrity
- Availability

Taken together, these are referred to as the "CIA" of data stored on a computer. Protection of configuration data such as `/etc/passwd` can be subsumed under data security. *System security* refers to the computing platform itself. The U.S. National Information Systems Security Glossary defines system security as follows:

System security. The protection of information systems against unauthorized access to or modification of information, whether in storage, processing or transit, and against the denial of service to authorized users or the provision of service to unauthorized users, including those measures necessary to detect, document, and counter such threats.

It is important to realize that system security enforces an iterative process consisting of applying security patches, regular audits, and controls, as well as having a secure system configuration to begin with. In this context, it is impossible to guarantee absolute security, as it is impossible to provide a 100% secure service. The goal is rather to find a tradeoff between security, usability of the system, and efforts required to maintain this security level. This compromise is influenced by the importance of the data stored on the computer to be secured and its intended usage scenario [1].

1.1.1 Integrity

Data with *integrity* is valid and has not been changed accidentally or maliciously. Integrity should be considered when data is stored or exchanged. Data received at the destination must be an exact copy of the source data. This means that, on the one hand, physical transmission and storage media must be reliable, so that data is transferred correctly without bit errors. On the other hand, data must not be altered by unauthorized entities having access without being detected. The scope of integrity

begins after the user -- the ultimate authority -- entrusts data to the system. User errors are therefore out of scope of integrity.

For network connectivity, special care has to be taken ensure integrity, regardless of whether the network is secured (for example, by encrypting transmission) or not. Third parties with access to the transport medium may re-route or alter data in transit. Measures taken to protect data integrity on a physical level include, among other things, restricting access to computers, keeping transmission media (such as cables and connectors) protected, and avoiding power outages and electrostatic discharges [2].

1.1.2 Confidentiality

Confidentiality is maintained when protected data can only be read or modified by authorized people or systems. It is a distinct concept from integrity: when data is sent over a network, it might be transmitted correctly without changes so that integrity is ensured, but it would no longer be confidential if intercepted by a third party. Integrity is not sufficient when unauthorized persons gain access to the data transmitted and extract valuable information from it. The confidentiality of data brings up three further questions:

- Who wants to access the data? (Authentication)
- What data can be accessed? (Authorization)
- How is data protected from unauthorized access?

Operating systems have several approaches to ensure that the entity trying to access data is the one it claims to be. Such as in Linux via Pluggable Authentication Modules (PAM), several authentication strategies can be implemented, from simple username/password combinations stored on the local machine over a centralized directory (NIS, Kerberos, LDAP, etc.) to hardware tokens or biometric scans. File access authorization can be issued using the classical (coarse) UNIX file permissions: read, write, and execute permissions on user, group, or world level. Newer fine-grained approaches -- Access Control Lists -- allow granting or denying specific rights to specific users.

The standard operating system security concept is implemented in software and relies on the kernel disallowing access to resources a user is not authorized to use. However kernel bugs (privilege elevation/escalation, unchecked parameters, etc.) may allow a user to access previously inaccessible memory regions, disk space, networks, or other resources. Physical access to the hardware allows users to bypass software checks by installing keyloggers, taking away hard drives and reading them in other machines, sniffing network traffic, and so on. Thus, further efforts, such as encryption of file systems, of individual files, of network traffic, and/or on the application level, have to be taken to secure confidential data.

1.1.3 Availability

Even if integrity and confidentiality are ensured, data is useless if it cannot be accessed and is not available. *Availability* measures ensure that data is never lost and is accessible at a predefined performance level when it is requested. Availability can be compromised in various ways:

- Destructive attacks -- called Denial-of-Service (DoS) attacks -- which target availability. They aim at making a computer or service unavailable or unusable by consuming all available network, CPU, storage, or operating system resources (such as file handles).
- Attacks that aim at the data itself by trying to delete or overwrite it.
- Accidental destruction of data. In most cases, it is not possible to prevent accidental destruction of data, because a user with the appropriate rights is seen as the ultimate authority. Organization can only be forearmed by having a backup/restore infrastructure in place.
- Last, but not least, bugs, configuration errors, physical environment, hardware failures, power outages, unwanted system reboots, and more can also affect availability.

To address the first two items in the list above, measures include firewalling/filtering of network packets, protecting file system integrity, and planning for additional resources that can be added on demand.

1.2 System Hardening

System hardening is a step by step process of securely configuring a system to protect it against unauthorized access, while also taking steps to make the system more reliable. Generally anything that is done in the name of system hardening ensures the system is both secure and reliable.

System hardening is necessary since "out of the box", some operating systems tend to be designed and installed primarily to be easy to use rather than secure. Most but not all systems can have security measures enabled that will make them suitable for high security, high reliability environments.

There are many reasons why taking the steps to harden the system is worth it.

- more confidence in the integrity of the data;
- performance improvements can be experienced since unnecessary services are removed, and inefficiencies in system configuration are detected;
- if there is a system failure, organization can recover faster;
- the company's reputation is protected;
- clients are happier as a result of fewer system failures or delays;
- to prevent lawsuits. The organization may have a legal liability to secure the private information of its employees, customers or research subjects.

If the system is not hardened then there are quite good chances that something bad will happen with the system, this can include the following

- Hackers move quickly. Most unprotected systems are compromised within 72 hours from the time they are installed according to results from the honeynet project [10];
- The system might be hijacked without user's knowledge, and then used to attack another system, or spread viruses, or distribute illegal content such as pornography or software;
- Company's proprietary information could be stolen;
- Money could be wasted paying employees to sit around, unable to do their work while the system is down;

Following are the main steps that administrators or user can follow to harden their system.

Step 1. Ensure that the hardware is robust

- Is it new enough to be considered reliable
- Identify the weak links and strengthen them (redundant disks, server clustering etc.)
- Ensure the environment is computer friendly (climate, location etc.)
- Provide physical security to eliminate tampering or theft

Step 2. Select and install a solid operating system

- New operating systems have not been massively probed by hackers. Mature operating systems are a known quantity. While the risks are known, so are the fixes.
- Features that are important include the ability to support fault tolerant measures such as uninterruptible power supply support (UPS), RAID disk arrays, logging, and access control measures including log on authentication and file protection.
- Strip down the OS to support only essential services
- Disable unnecessary protocols and subsystems
- Remove, disable, or rename known “target” accounts
- Require strong local and remote authentication for access
- Strictly manage users and groups to control inappropriately powerful rights and memberships (Least Privilege)
- Enable auditing to track important events
- Install a 3rd party firewall and monitor the logs
- Apply all relevant hot-fixes, patches and service packs

Step 3. Install and configure the file system

- Configure Access Control Lists (ACL) to eliminate inappropriately powerful rights and permissions (Least Privilege);
- Enable auditing to track important events;

- Begin by fully locking down all directories and then providing controlled access to user groups;
- Access to specific users should only be made on an exception basis.

Step 4. Configure applications/services

- Install only essential applications and services;
- Install only tested and approved software;
- Remove or disable any unneeded applications and services that are installed by default – remove the files where possible;
- Set access control within applications/services where applicable;
- Apply all relevant hot-fixes, patches and service packs;
- Remove any sample data (scripts, sample web pages, etc).

Step 5. Configure server side applets/scripts

- Install only essential applications, applets and scripts;
- Install only tested and approved software;
- Verify that applets and scripts perform only their intended function;
- Apply all relevant hot-fixes, patches and service packs.

1.3 Operating System Hardening

The first point that every computer user needs to understand is that no information system is 100% secure 100% of the time. Once this fact is ingrained, the user is ready to take the second and final step right into the "continuous security process." While numerous security vendors show that their product is the only way to a secure system, the truth is that there is no need to purchase any program or device to keep attackers out of the system. A computer can be hardened by ensuring that the operating system is patched, services and third-party programs are locked down, and the system is secure from physical attacks. Unfortunately, this type of control only exists in an ideal world where all of the following are true:

- Vendors know about vulnerabilities before malicious hackers do.
- The vendors release patches.

- The patches get applied immediately.
- The patches work as advertised.
- System administrators have complete control over all users' actions.

To help fill the gap between this ideal world and reality, IDS, firewalls, and other products and procedures build a bridge of useful tools and techniques that help to harden information systems.

If there's one best place to start the hardening process, it's the operating system. This is because the operating system represents the core of the information system. Ultimately, the operating system is responsible for ensuring that users are securely authenticated and controlled. Fortunately, hardening the operating is not a difficult task, although the initial implementation can be tedious. From the moment an operating system install is planned, security should be a top priority. Unfortunately, this isn't often the case, which results in immature installations and ultimately leads to a compromised system.

The following sections outline some measures that administrator or user can follow to harden their operating system.

1.3.1 Install Offline

During the installation of an operating system, it should remain physically disconnected from any network—especially the Internet. While statistics vary depending on who provides them, it's estimated that a default operating system installation of either Windows or Linux will be scanned and/or hacked within an hour of being connected to the Internet. Unfortunately, it often takes longer than this to get all the correct security patches installed. Ironically, most users need the Internet connection to download the necessary security patches, which presents a catch-22 situation. So, while this is a good practice, it's not always practical for home users who don't have a spare computer to which they can download the required patches.

1.3.2 Lock Down the Operating System

While a default installation may result in a usable computer, it can almost be guaranteed that it's susceptible to at least one vulnerability that could be exploited by an attacker to gain unauthorized access to the system. To prevent this, the administrator should be prepared to install all applicable patches and updates—which means knowing ahead of time what major updates are available for that operating system, often in the form of service packs or updated releases. Fortunately, most major operating systems provide these patches and updates in a downloadable format, which means that the organization can prepare the necessary updates on a CD before installation.

Once the operating system is installed with current updates, it is needed to ensure that the system stays up to date. To assist, most major operating systems have a built-in update program that keeps track of available "hot fixes" and the patches that are actually installed. For example, Microsoft uses the Windows Update tool and Red Hat uses a similar up2date program to keep operating system updates current.

1.3.3 Lock Down the Services

The core operating system is only a small part of an information system. Built on top of this fundamental code are the extras that define how and where the computer will be used. For example, one of the most well-known services for the Windows Server operating system is Internet Information Server (IIS). This piece of software is notorious for giving attackers an open welcome to the host computer, due to numerous programmer errors.

Just like the operating system, all services and third-party programs on the computer should be checked to ensure that they're the most current versions and that they're safe to use. While many administrators understand this principle and make efforts to remove unwanted services, many small office/home office users may not know which services their systems are running. For example, an older Microsoft operating system,

Windows NT 4, included in its default installation a fully-functional setup of IIS. Unfortunately, that version of IIS was vulnerable to numerous exploits, and was lucky to survive even 10 minutes connected to the Internet before becoming host to a hacker or worm.

To prevent from becoming a victim the command `netstat -an` can be used to see what communication ports are accepting incoming data. In addition to this check, tools such as Nessus, Nmap, and Stealth can quickly give a snapshot of the system and the potentially vulnerable services that are running in the background. Using a tool like this helps speed up the lockdown process because it is learned that which services or programs are responsible for the open port.

1.3.4 Establish a Baseline

Once the computer is patched and locked down, an important step that should be taken before opening it to the world is to establish a baseline for that system. This key ingredient to the system's hardening does nothing to actually protect the system from intrusion or an attack. Instead, it's used to determine what files and/or settings were altered as a result of any change—authorized or not.

While the establishment of a baseline doesn't technically fall into operating system hardening, it's a necessary step in setting up the security systems that will surround the core operating system and its applications. Due to the risk associated with placing a system online where it's accessible to attacks, the baseline must be created *before* the system has any interaction with the outside world, with the exception of load and resource usage.

Chapter 2

Literature Survey

Within this work there are a few areas that need to be investigated before the true scope of the task at hand can be fully recognized. The concept of operating system hardening needs to be understood and it will be helpful to explore the background concept. As this work is aimed at producing a method of scoring the hardening capability present in an operating system that will aid the system administrator it will also be useful, and indeed necessary, to research current such methods available. Areas of difficulty that administrators currently experience should also be investigated to identify whether the proposed method would be a worthwhile product.

2.1 What is Operating System Hardening

Operating system hardening is the process of eliminating basic vulnerabilities on the operating system. It is a kind of setup checklist and is one of the first and most important considerations when securing a public access workstation or server. The list of possible steps that can be taken to do this is very long, and the procedure can vary for each installation environment. Furthermore, each operating system requires different steps.

Most computers offer network security features to limit outside access to the system. Software such as antivirus programs and spyware blockers prevent malicious software from running on the machine. Yet, even with these security measures in place, computers are often still vulnerable to outside access. System hardening, also called Operating System hardening, helps minimize these security vulnerabilities.

The purpose of operating system hardening is to eliminate as many security risks as possible. This is typically done by removing all non-essential software programs and utilities from the computer. While these programs may offer useful features to the

user, if they provide "back-door" access to the system, they must be removed during system hardening.

Advanced operating system hardening may involve reformatting the hard disk and only installing the bare necessities that the computer needs to function. The CD drive is listed as the first boot device, which enables the computer to start from a CD or DVD if needed. File and print sharing are turned off if not absolutely necessary and TCP/IP is often the only protocol installed. The guest account is disabled, the administrator account is renamed, and secure passwords are created for all user logins. Auditing is enabled to monitor unauthorized access attempts. While these steps are often part of operating system hardening, system administrators may choose to perform other tasks that boost system security.

2.2 Existing Operating System Hardening Methods

The findings of my research so far have indicated that methods and tools are available to check whether the operating system configuration matches the accepted security standard but no method or tool is available to evaluate the hardening capability of the operating system. However examination of the available methods and functionality of tools has provided valuable information and established a base in designing the hardening capability scoring scheme.

2.2.1 Center for Internet Security Scoring Tools

The Center for Internet Security (CIS) is a forerunner at defining consensus best-practice security configurations for Internet-connected computers. CIS works with organizations, information security professionals and auditors until they all agree on security configuration specifications that represent a prudent level of due care. Their free Benchmark and Scoring Tools [3] provide a quick and easy way to evaluate the system and compare its level of security against their minimum due care security benchmarks. Various reports shows how to harden both new and active systems while

monitoring them to ensure that security settings continuously conform to the configuration specified in the benchmark.

CIS Benchmarks [4] enumerates security configuration settings and actions that "harden" the system. The CIS Level-I Benchmarks set a prudent level of minimum due care, while CIS Level-II Benchmarks vary depending on architecture and server function. The Level-I Benchmark settings can be applied with little security knowledge, as they are unlikely to cause an interruption of service to the operating system or the applications that run on it. However, the CIS Level-II Benchmarks should be used by system administrators who know how to apply them to the operation systems and applications running in their environment. The Benchmarks are developed through a unique global consensus process. The process is effective because it pools the security knowledge of technical professionals, and cost-efficient because it taps voluntary involvement by security specialists in organizations that share the benefits of the technical work. The benchmarks are open, that is, publicly available to everyone.

The Center's Internet security benchmarks are intended to:

- Provide managers, business partners and insurance underwriters with a security 'ruler', where each increment on the ruler represents a set of security-enhancing actions. This security ruler will enable an organization to select the level of security deemed appropriate for that enterprise and implement the specific technical actions associated with the security level chosen;
- Include interventions that can be implemented before, during, and after attacks to reduce losses; and

CIS Scoring Tools enable end users to verify the security configuration of systems prior to network deployment, monitor systems and network devices for ongoing conformity with the benchmarks, and demonstrate to auditors and business partners their compliance with the internationally accepted standard for security configuration. The Tools are host based and produce reports that guide users and system administrators to secure both new installations and production systems. CIS Scoring Tools are available on the CIS web site for most of the CIS Benchmarks.

The CIS Next Generation Scoring Tool (“**NG Tool**”) [5] is written in JAVA. It is designed to be platform independent, allowing it to run in almost any environment and to score the systems compared to all of the CIS benchmarks. The NG Tool reads Benchmark files and configuration check files. Both files are in XML format. The Benchmark files express the consensus configuration recommendations. They instruct the NG Tool to check a system’s technical security controls and report on the compliance of those controls with the Benchmark recommendations. The configuration check files express the method the NG Tool uses to check the system for the recommended technical controls. They instruct the NG Tool on how to execute the configuration checks.

The detailed data reports generated by the compliance checking process enable users to compare the configuration of their systems with the Benchmark recommendations (defined as Benchmark “items”). For system configurations that are in compliance with the Benchmark recommendations, the NG Tool reports those Benchmark items as “pass.” For system configurations NOT in compliance with the Benchmark recommendations, the NG Tool reports those Benchmark items as “fail.” The reports also guide users on how to enact the recommended controls on “failed” items, thereby improving the security configuration of the scanned system and bringing it into closer compliance with the consensus configuration standard.

2.2.2 The Bastille Hardening Program

The Bastille Hardening program "locks down" an operating system, proactively configuring the system for increased security and decreasing its susceptibility to compromise. Bastille can also assess a system's current state of hardening, granularly reporting on each of the security settings with which it works. Bastille was originally conceived of by a group of concerned system administrators at a conference organized by the SANS Institute, an ally of the Bastille Project [6]. Jay Beale wrote the initial program and now leads a number of developers, beta-testers and concept-creators in Bastille's development.

Bastille currently supports the Red Hat (Fedora Core, Enterprise, and Numbered/Classic), SUSE, Debian, Gentoo, and Mandrake distributions, along with HP-UX. Bastille's focus is on letting the system's user/administrator choose exactly how to harden the operating system. In its default hardening mode, it interactively asks the user questions, explains the topics of those questions, and builds a policy based on the user's answers. It then applies the policy to the system. In its assessment mode, it builds a report intended to teach the user about available security settings as well as inform the user as to which settings have been tightened.

Bastille runs in two modes -- Assessment and Hardening. Assessment mode scans the system for the current settings of the configuration options that Bastille hardens. It records the state of these options and scores the security of the host based on that state. A score out of ten is returned together with a HTML and text report detailing the state of each option. The higher the score, the more secure is the host.

In Hardening mode, Bastille prompts the user to answer questions and at the end of the process applies the requested configuration changes. Hardening mode can run either from the command line or via a GUI. Bastille also has a revert function that can undo the Hardening process -- although it's only really effective if changes to the configuration files haven't manually made. If this is the case, then Bastille will warn that changes may not be able to be reverted. The Hardening process can be executed multiple times. For example, if the host has had some security controls reverted and needs to be re-hardened.

The application can harden a variety of elements in the operating system and some of the applications, most notably Apache. It can apply more granular controls to SUID and harden network daemons, including management tools and NFS/Samba. It can also harden the LILO or GRUB installation and help in configuring a firewall for the host. In all, there are approximately fifty hardening features in the Bastille application.

Installing Bastille is easy, since many distributions come with Bastille available as a package, like Debian, Gentoo and Red Hat Fedora. It can also be downloaded in the form of an RPM or as a source file.

Bastille can be a powerful tool, particularly for ensuring a consistent security baseline on Linux hosts. It doesn't guarantee that the host is secured against all threats, but it does take care of a lot of configuration weaknesses and security configuration that can be time-consuming and complicated

2.3 Hardening Windows XP

This checklist covers XP Professional and XP Home Edition. Unfortunately, Windows XP Home Edition doesn't have all of the security features of XP Professional, so not all of the options are available for both versions. When implementing these recommendations, the trade off between increased security levels and usability for any Operating System must be considered. The organization should assess its potential security risks, determine the value of its data, and balance its needs accordingly [11].

Provide Physical Security for the machine

The simple fact is that most security breaches in corporate environments occur from the inside. Keep the workstation in an office that locks, install a lock on the CPU case, keep it locked, and store the key safely away from the computer at a secure location.

Use NTFS on all your partitions

The FAT16/FAT32 file systems that were shipped with Windows 95/98/ME offered no security for data and left the system wide open to attacks. The NTFS file system is faster than FAT32 and allows setting permissions down to the file level. In addition, using NTFS on Windows XP Professional allows encrypting files and folders using the Encrypting File System (EFS).

Disable Simple File Sharing

Both Windows XP Home Edition and XP Professional workstations that are not part of a domain, use a network access model called "Simple File Sharing," where all attempts to log on to the computer from across the network are forced to use the Guest account (to prevent them from using a local Administrator account that wasn't

configured with a password) This means that if the system is connected to the internet and don't use a secure firewall, the files contained within those shares are available to just about anybody.

To disable Simple File Sharing on XP Professional:

- Click **Start > My Computer > Tools > Folder Options**
- Select the **View** tab
- Go to **Advanced Settings,**
- Clear the Use **Simple File Sharing** box
- Click **Apply**

Unfortunately, XP Home Edition doesn't allow to disable Simple File Sharing and is unable to join a domain, so the best that can be done is to set the shared folders to be read only, hide the file shares by using a \$ sign after the folder name, or if the NTFS file system is being used, use the "Make Private" option in the folder properties. For Windows XP Professional workstations that are part of a domain or that have Simple File Sharing disabled, use the "Classic" NT security model that requires all users to authenticate before granting access to shared folders.

Use passwords on all user accounts

Both Windows XP Professional and Home Edition allow user accounts to utilize blank passwords to log into their local workstations, although in XP Professional, accounts with blank passwords can no longer be used to log on to the computer remotely over the network. Obviously, blank passwords are a bad idea if the organization cares about security. Make sure that passwords are assigned to all accounts, especially the Administrator account and any accounts with Administrator privileges. By the way, in XP Home Edition all user accounts have administrative privileges and no password by default. Make sure that this hole is closed as soon as possible

Use the Administrator Group with care

It's very common for home users and small business administrators to simply give all local accounts full Administrator privileges in order eliminate the inconvenience of

logging into another account. However this practice gives a hacker the opportunity to try to crack a greater number of administrator level accounts and increases his/her chance for success. It also increases the odds that malicious code executed via an e-mail attachment or other vector can do more damage to the files. In a workgroup consider placing local users with a greater need for control in the local Power Users group, instead of the Administrators group. And avoid the temptation of using the local administrator account as the default login account.

Disable the Guest Account

The guest account has always been a huge hacker hole, and should be disabled as soon as workstation is installed. Unfortunately, this setting recommendation only applies to Windows XP Professional computers that belong to a domain, or to computers that do not use the Simple File Sharing model.

Use a firewall if there is a full time internet connection

Having instant, high speed access to the internet is a real convenience but it also puts your data at risk. Although XP comes with a built in firewall (called ICF), it is not enabled by default, and it only filters incoming traffic without attempting to manage or restrict outbound connections at all. While this may be fine for most users, it is highly recommended to use a third party personal firewall such as BlackIce. For corporate users already behind a firewall, consider using Group Policy to enable ICF and disable specific ports when users are not connected to the corporate network.

Use a router instead of ICS

The Internet Connection Sharing feature within XP allows a user to connect one PC to the internet and then share that connection with the rest of the computers within his home or small office network. While it was generally a good idea when it was conceived, if there is a high speed connection a real router is a faster, easier to configure, and more secure.

Install AntiVirus Software on all workstations

Viruses and other forms of malicious software have been around for years, but today's malware utilizes the internet and e-mail systems to spread globally in a matter of

hours. Installing AntiVirus software is a basic step in protecting the data, but it's near useless if the definitions aren't updated.

Keep up to date with hotfixes and service packs

Windows XP is a complex operating system and is not immune to its own bugs and security holes. Its common tactic for hackers to use the latest known security hole to break into a system and work backward from there until they find an open door that gives them full access. In fact 99% of system breaches are executed using known security vulnerabilities that were never patched. Use the Windows Update feature or automatic update to keep the system up to date. The Microsoft Baseline Security Analyzer can also be used to check the system for known vulnerabilities.

Password protect the screensaver

Once again this is a basic security step that is often circumvented by users. Make sure all of the workstations have this feature enabled to prevent an internal threat from taking advantage of an unlocked console. For best results, choose the blank screensaver or logon screensaver. Avoid the OpenGL and graphic intensive program that eat CPU cycles and memory.

Secure your wireless network

The new 802.11 wireless standard allows to roam freely without cables and make anywhere the virtual office. This also gives hackers another open door to the system's data if is not locked. A recent survey in the U.K found that of 5,000 wireless networks that were discovered by simply driving around the city with a wireless enabled laptop, 92% were wide open. As "drive by" hacking and warchalking are becoming common practice, any hacker with a laptop and a Pringles can could potentially compromise the organizations network.

Secure your Backup tapes

It's amazing how many organizations implement excellent platform security, and then don't encrypt and/or lock up their backup tapes containing the same data. It's also a good idea to keep the Emergency Repair Disks locked up and stored away from the workstations as well.

Use the Security Configuration Manager and templates provided with XP Professional

The Security Configuration Manager (SCM) set of tools allows security administrators to define security templates that can be applied to individual machines or any number of machines via group policy. Security templates can contain password policies, lockout policies, Kerberos policies, audit policies, event log settings, registry values, service startup modes, service permissions, user rights, group membership restrictions, registry permissions and file system permissions. Microsoft provides a number of predefined security templates to help in locking down the PC via Group Policy. These templates represent low, medium, and high security configurations, which can be customized to meet the company's specific security needs. The security relevant registry values configurable by SCM appear under Local Policies\Security Options when using SCM tools such as the security templates snap-in, the security configuration and analysis snap-in, or the security settings extension to Group Policy. Note: This feature is not available on Windows XP Home Edition

Password Security

A good password policy is essential to the network security, but is often overlooked. In large organizations there is a huge temptation for lazy administrators to create all local Administrator accounts (or worse, a common domain level administrator account) that uses a variation of the company name, computer name, or advertising tag line. i.e. *%companyname%#1*, *win2k%companyname%*, etc. Even worse are new user accounts with simple passwords such as "welcome", "letmein", "new2you", that aren't required to change the password after the first logon. Use complex passwords that are changed at least every 60 -90 days. Use Group Policy or the local computer policy to set restriction on password age, length, complexity, lockout duration, and number of bad attempts. (Click **Start** > **Run** > type **GPEDIT.MSC** > Go to **Computer Configuration** > **Windows Settings** > **Security Settings** > **Local Policy** > **Security Options**) Passwords should contain at least eight characters, and preferably nine (recent security information reports that many cracking programs are using the eight character standard as a starting point). Also, each password must follow the standards set for strong passwords. The basic goal is that the password should be complex enough to foil hacker attempts, and not so complex that users will

have difficulty remembering their passwords and end up writing them on sticky notes pasted to the bottom of their keyboards.

Use software restriction policies

Using a software restriction policy, unwanted programs can be prevented from running; this includes viruses and Trojan horses, or other software that is known to cause conflicts when installed. Software restriction policies can also be used on a standalone computer by configuring the local security policy, or can integrate with Group Policy and Active Directory. (Click Start > Run > type GPEDIT.MSC > Go to Computer Configuration > Windows Settings > Security Settings > Software Restriction Policies) Note: This feature is not available on Windows XP Home Edition

Limit the number of unnecessary accounts

Eliminate any duplicate user accounts, test accounts, shared accounts, general department accounts, etc., Use group policies to assign permissions as needed, and audit the accounts regularly. These generic accounts are famous for having weak passwords (and lots of access) and are at the top of every hacker's list of accounts to crack first. This can be a big problem at larger companies with understaffed IT departments. An audit at a Fortune 10 company revealed that 3,000 of their 15,000 active user accounts were assigned to employees who no longer worked for the company.

Rename the Administrator Account

Many hackers will argue that this won't stop them, because they will use the SID to find the name of the account and hack that. Renaming the Administrator account will stop some amateur hackers cold, and will annoy the more determined ones. Remember that hackers won't know what the inherit or group permissions are for an account, so they'll try to hack any local account they find and then try to hack other accounts as they go to improve their access. If the accounts are renamed, try not to use the word 'Admin' in its name. Pick something that won't sound like it has rights to anything.

Consider creating a dummy Administrator account

Another strategy is to create a local account named "Administrator", then giving that account no privileges and impossible to guess +10 digit complex password. This should keep the script kiddies busy for a while. If a dummy Administrative account is created, auditing should be enabled to know when it is being tampered with.

Replace the "Everyone" Group with "Authenticated Users" on file shares

"Everyone" in the context of Windows XP security, means anyone who gains access to the network can access the data. "Everyone" Group should never be assigned to have access to a file share on the organization's network, use "Authenticated Users" instead. This is especially important for printers, who have the "Everyone" Group assigned by default.

Prevent the last logged-in user name from being displayed

When Ctrl-Alt-Del is pressed, a login dialog box appears which displays the name of the last user who logged in to the computer, and makes it easier to discover a user name that can later be used in a password-guessing attack. This can be disabled via the Group Policy snap in. (Click Start > Run > type GPEDIT.MSC > Go to Computer Configuration > Windows Settings > Security Settings > Local Policy > Security Options) [20].

Make sure that Remote Desktop is disabled

Remote Desktop is a new feature in Windows XP Professional that allows to connect to the computer remotely and work as though the person sitting at the console. While this may be convenient for some users, it also makes it easier for a hacker who has compromised one of the user accounts to log in directly to the machine from a remote location. Fortunately, remote desktop is not enabled by default on Windows XP Professional, and is not available for Windows XP Home Edition.

Disable unnecessary services

An unnecessary service is an unnecessary hacker hole, as well as a drain on system resources [17]. Services can be disabled via Control Panel > Administrative Tools > Services

Disabling the following services can be considered

- **Disable IIS** - Luckily, IIS is not installed by default in Windows XP. If you enabled it during installation, and aren't using it, it should disable it. If IIS is being used on the workstation, there is need to take extra precautions to lock it down and stay on top of security vulnerabilities specific to web services.
- **Netmeeting Remote Desktop Sharing**
- **Remote Desktop Help Session Manager** - If you haven't disabled this via Group Policy already
- **Remote Registry**
- **Routing & Remote Access**
- **SSDP Discovery Service** - this disables the Universal PNP Service, which leaves TCP Port 5000 wide open.
- **Universal Plug and Play Device Host** - This is designed to allow the computer to automatically connect to network-enabled appliances. Although there are no practical uses for this technology yet, several severe security flaws have already been discovered. Use the UnPlug and Pray utility from Gibson Research to disable "Universal Plug and Play". Gibson's web site has additional information about why this is necessary
- **Telnet**

Enable EFS (Encrypting File System)

Windows XP Professional ships with a powerful encryption system that adds an extra layer of security for drives, folders, or files. This will help prevent a hacker from accessing the files by physically mounting the hard drive on another PC and taking ownership of files. Be sure to enable encryption on Folders, not just files. All files that are placed in that folder will be encrypted.

If Offline Folders are used, encrypt the local cache

With Windows XP, any shared folder that is available on the network (or any Web page) can be marked to be made available offline. The contents of these shared folders (or pages) are copied to an Offline Files database that is known as the client-side cache, where you can access them when not connected to the network. To safeguard offline files against theft, the client-side cache can be encrypted. To encrypt the Offline Files database on a local computer: Click **Start > Folder Options > select**

the **Offline Files** tab > If Offline Files are not already enabled, click the **Enable Offline Files** option > Click the **Encrypt offline files to secure data** option > Click **OK**.

Encrypt the Temp Folder

Applications such as Microsoft Office use the temp folder to store copies of files while they are being updated or modified, but they don't always clean the folder when you close the program. Encrypting the temp folder provides an extra layer of security for the files.

Clear the page file at shutdown

The Windows XP Page file can occasionally contain passwords and other sensitive data that the system has stored into memory. The operating system can be forced to clear the page file by using the Local Computer Policy via the MMC, or via Group Policy

Enable Auditing on the Workstations

While this is a fairly normal practice for servers, it isn't usually performed on workstations unless there is a high risk of data theft. Our philosophy is that the time to fix the roof is before it starts to rain. By selectively auditing a few key actions, there will be a place a place to start investigating theft or destruction of data if someone ever does compromise the workstation. Auditing the following actions are recommend:

Event	Level of Auditing
Account logon events	Success, failure
Account management	Success, failure
Logon events	Success, failure
Object access	Success
Policy change	Success, failure

System events	Success, failure
---------------	------------------

Disable default shares

Windows XP automatically creates a number hidden administrative shares that the operating system uses to manage the computer environment on the network. These default shares can be disabled via the Computer Management console in the Control Panel, but they are re-enabled by the system after restarting the computer. The default hidden shares are:

Path and Functions	
C\$ D\$ E\$	Root of each partition. For a Windows XP Professional computer, only members of the Administrators or Backup Operators group can connect to these shared folders.
ADMIN\$	%SYSTEMROOT% This share is used by the system during remote administration of a computer. The path of this resource is always the path to the Windows XP system root (the directory in which Windows XP is installed: for example, C:\Winnt).
FAX\$	This used by fax clients in the process of sending a fax. The shared folder temporarily caches files and accesses cover pages stored on the server.
IPC\$	Temporary connections between servers using named pipes essential for communication between programs. It is used during remote administration of a computer and when viewing a computer's shared resources
NetLogon	This is used by the Netlogon service to process log on requests
PRINT\$	%SYSTEMROOT%\SYSTEM32\SPOOL\DRIVERS Used during remote administration of printers.

To prevent these shares from being created at startup, open RegEdit and edit the following key: **HKeyLocal**

Machine\SYSTEM\CurrentControlSet\Services\LanManServer\Parameters

Create a DWORD value called AutoShareWks and set the parameter to 0.

Disable Dump File Creation

A dump file can be a useful troubleshooting tool when either the system or application crashes and causes the infamous "Blue Screen of Death". However, they also can provide a hacker with potentially sensitive information such as application passwords. The dump file can be disabled by going to the **Control Panel > System > Advanced > Startup and Recovery** and change the options for "Write Debugging Information" to None. If it is needed to troubleshoot unexplained crashes at a later date, this option can be re-enable until the issue is resolved but it must be disabled again later and delete any stored dump files

Disable the ability to boot from a floppy or CD ROM on physically unsecured systems.

There are a number of 3rd party utilities that pose a security risk if used via a boot disk (including resetting the local administrator password.) If the security needs are more extreme, the floppy and CD drives should be removed entirely. Access to the floppy and CD-ROM drives in Windows XP Professional can be restricted via the Local Computer Policy in the MMC (Click **Start > Run > type GPEDIT.MSC > Go to Computer Configuration > Windows Settings > Security Settings > Local Policy > Security Options**)

Disable AutoRun for the CD-ROM

One of the easiest ways for a hacker with physical access to a company's PC's to distribute malicious code is via the CD-ROM. By creating a custom CD with a payload set to launch from the autorun feature in any machine, a hacker can affect any number of unlocked systems without ever leaving a fingerprint or touching a keyboard. Or he/she can simply leave a few of these lying around the office marked "MP3's", or "Payroll Data" and wait for an unsuspecting user to simply pick it up and insert it into their machine. This function in Windows XP Professional can be disabled by clicking **Start > Run > and type GPEDIT.MSC** Then go to **Computer**

Configuration > Administrative Templates > System > Locate the entry for Turn autoplay off

Consider using SmartCard or Biometric devices instead of passwords.

The more stringent the password policy is, the more likely the users will begin keeping paper password lists in their desk drawers, or taped to the bottom of their keyboard. Windows 2000 supports these devices, so the cost vs. risks of the most sensitive data is to be considered. When using smart cards make sure to configure the workstation to lock when the smart card is removed. Under **Local Policies > Security Options > Interactive logon > Smart card removal behavior > Lock Workstation setting**

Consider implementing IPSec

Basically, IPSec provides encryption for network sessions using the Internet Protocol (IP) and promises to offer transparent and automatic encryption of network connections.

2.4 Hardening Linux

After completing the preliminary planning and preparation and performing the minimal installation, several configuration steps are to be considered. These steps are generally referred to as hardening Linux [14]:

- Securing the boot process
- Securing services and daemons
- Securing local file systems
- Enforcing quotas and limits
- Enabling Mandatory Access Control
- Updating and adding security patches

Securing the boot process

The boot loader (LILO or Grub) has to be configured so that it does not allow any user intervention at boot time; this prevents users from passing kernel parameters at

boot prompt. Configured it to require a password unless it is need to perform remote reboots (such as in a remote datacenter). This is an additional precaution against people who have physical access to the machine; it prevents casual hacks of such events as booting with the parameters `single` or `init=/bin/sh` to obtain a root shell, etc.

For LILO, replace the parameter `prompt` with `password` in the `lilo.conf` configuration file (usually in `/etc`). For Grub, the relevant parameters are `hiddenmenu`, `default 0`, and `password` in the Grub configuration file (usually in `/boot/grub/grub.conf`). Ensure that the run-level configuration requires the root password when switching to single user mode by adding `sp:S:respawn:/sbin/sulogin` to `/etc/inittab` [21].

Prevent users from invoking a reboot using Ctrl-Alt-Del: disable the `ctrlaltdel` entry in `/etc/inittab` by commenting out the line with the `ctrlaltdel`. By adding a hash (#) sign similar to this -- `#ca::ctrlaltdel:/sbin/shutdown -t5 -rf now --` you can prevent that key combination from ever triggering a reboot.

Securing services and daemons

The first step to a secure configuration of services is to disable all services that are not needed. Services that aren't offered can't be exploited by potential intruders, effectively reducing your risk.

Disable insecure services and replace their functionality with more secure options. For instance, telnet is not encrypted, so use the encrypted ssh service instead of telnet

Consider these elements when securing services:

- `/etc/inittab`
- Boot scripts in `/etc/init.d`
- The `inetd/xinetd` daemons
- TCP wrappers
- Firewalls

`/etc/inittab`

During the boot process, the `init` process reads the entries in the file `/etc/inittab`. Each entry -- one per line -- defines which program is to be run in certain situations. Such programs are either services themselves or are used to start and stop services.

The init process distinguishes several states, so-called *run levels*, identified by a single character. When a run level is entered or certain events (such as power failure) occur, the entries are evaluated and the appropriate commands are executed.

For a secure Linux system, it is needed to understand the functions of all entries in `/etc/inittab` and disable potentially unwanted services by deleting the entry or commenting it out using the hash sign at the beginning of the line.

Two types of entries are common in all Linux systems. The first is used to launch a program called `/sbin/getty` (or similar), which is used to allow logins on the Linux virtual consoles or serial lines. The second type launches a script usually named `rc` in the `/etc/rc.d` directory with the current run level as the parameter. This script controls the starting and stopping of services.

Boot scripts in `/etc/init.d`

Boot scripts located in `/etc/init.d` are used to start or stop system services. For each run level, there is a `/etc/rcN.d` directory (with "N" being the identifier of the run level) that contains soft links to the boot scripts that are to be invoked on run-level change.

If the link name starts with an "S" the script is executed when entering the run level to start the corresponding service; if it starts with "K" the script is executed when exiting the run level to stop the service.

Most of the time the name of the boot script reveals which service it controls. To prevent a service from being started in a specific run level, delete the links from the run-level subdirectory to the corresponding boot script or replace the original boot script in `/etc/init.d` by a dummy script doing nothing.

The `inetd`/`xinetd` daemons

Services can also be invoked on demand when requested by a client. These requests are given to the super daemon `inetd` or `xinetd`. The super daemon then decides which service to start and passes the request to the corresponding daemon. Typically services like `telnet`, `ftp`, `rlogin`, etc., are started using `inetd` or `xinetd`.

The inetd daemon is configured in `/etc/inetd.conf`, which contains entries for each service to be offered by the super daemon. An entry configuring an FTP server could look like this -- `ftp stream tcp nowait root /usr/bin/ftpd in.ftpd -el` -- and you can disable it by commenting it out using a hash sign.

For security reasons, the use of xinetd is recommended. In contrast to inetd, xinetd is able to start rpc-based services and provides access control. xinetd can limit the rate of incoming connections, number of incoming connections from specific hosts, or total number of connections for a service. xinetd is configured by distinct configuration files for each subordinate daemon. These files are located in `/etc/xinetd.d/`. The example configuration file for the FTP server above would be called `/etc/xinetd.d/ftp` and would look like this:

Configuration file, `/etc/xinetd.d/ftp`

```
service ftp
{
    socket_type    = stream
    protocol       = tcp
    wait           = no
    user           = root
    server          = /usr/bin/ftpd
    server_args     = -el
    disable        = yes
}
```

To disable the service, the parameter `disable` is set to `yes` as in the previous example.

TCP wrappers

If it is decided to not use xinetd but to instead rely on `inted`, logging of requests and specifically permitting/denying certain networks can be achieved by using TCP wrappers. TCP wrappers evaluate `/etc/hosts.allow` and `/etc/hosts.deny` for authorization and logging purposes and wrap client requests without directly answering them. Once authorization is successful, the request is forwarded to the originally requested service.

Using TCP wrappers delivers two main advantages over using plain inetd:

- The requesting client does not recognize the TCP wrapper; therefore, good guys do not realize any difference, and bad guys do not get any information on why their request failed.
- A TCP wrapper works regardless of any wrapped services, enabling applications to share their configuration files and therefore easing up administration.

Firewall

To prevent unwanted communication to services that are not supposed to run or that are not supposed to be accessed from certain networks like the Internet, the installation of a firewall is recommended. A firewall provides controlled communication between networks based on trust levels and grants or denies access to certain services using a rule-based security policy leveraging the least-privilege principle.

Securing local filesystems

Securing the local filesystem deals with ownership of files and directories and access rights to them. To secure the filesystem, the protection bits of files and directories must set to grant minimum access only.

Special care should be taken with respect to inappropriate permissions regarding world-writable files and system directories and so-called setuid or setgid commands. These commands are executed with higher user privileges than the user running the command actually has. This might be necessary to access files that only root has access to (such as /bin/passwd needs to access /etc/passwd). For each of these commands, make sure that it really needs the setuid/setgid bits set. If this is not the case, disable it.

When all files on a partition definitely do not need the setuid/setgid bits, the nosuid option in /etc/fstab disables it for each file on the corresponding filesystem. Furthermore, for very sensitive data, it might make sense to encrypt that data and protect it with a passphrase. GnuPG provides a suitable package to accomplish this.

Enforcing quotas and limits

The Linux PAM (Pluggable Authentication Modules) can enforce some useful limits, which can be configured in the `/etc/security/limits.conf` file. These limits apply to a single session. The total limitation can be controlled with the `maxlogins` option. Entries within `limits.conf` have the following structure: `username|@groupname type resource limit`.

Groupnames must be preceded by the `@` to distinguish them from usernames. The type must be either `soft` or `hard`. Soft-limits can be exceeded and are usually warning marks, whereas hard-limits cannot be exceeded. A resource can be one of these keywords:

- `core` - Limits the size of a core file (KB).
- `data` - Maximum data size (KB).
- `fsize` - Maximum file size (KB).
- `memlock` - Maximum locked-in memory address space (KB).
- `nofile` - Maximum number of open files.
- `rss` - Maximum resident set size (KB).
- `stack` - Maximum stack size (KB).
- `cpu` - Maximum CPU time in minutes.
- `nproc` - Maximum number of processes.
- `as` - Address space limit.
- `maxlogins` - Maximum number of logins allowed for this user.

Quotas allow to restrict the number of inodes and the consumable space for users and groups. Note that a quota is defined per mount point, so if users have write access on several partitions, make sure to define quotas for each of them.

Quotas are an administrator's way of minimizing the risk of DoS attacks that rely on filling up all the available space of a hard drive (which prevents other processes from creating temporary files and thus makes them fail). Depending on which distribution is being used, shipped quota tools can be installed.

Quotas must be enabled in the kernel. Most distributions today come with quota support. Assigning quotas is done using the tool `edquota`.

Enabling Mandatory Access Control

Additional security can be achieved with Mandatory Access Control, or MAC, by implementing SELinux. With MAC, permissions in the operating system are governed by the user/group ID under which a process runs and the user/group IDs of the objects (files) being accessed. Also, with MAC, policies for each individual process are enforced by the Linux kernel, and they control what a process may do.

That way, in a properly configured system with MAC, a service that has been hijacked or hacked into cannot take over the system. Even though the user or group ID under which the service processes run (worst case scenario: root) may match the file permissions of critical system files such as `/etc/passwd`, the policy in place disallows access to them.

Updating and adding security patches

To keep the systems as secure as possible it is needed to stay informed of new fixes and patches for the software. There are several sources for this information, but naturally the software vendors and the Linux distributors should provide this information in a timely manner.

When a new update becomes available, it should be checked to see if it applies to the system and organization's security requirements. Installing an update can itself be the cause of security issues. Also consider that each update may introduce new vulnerabilities, or if the update fails, your system may be left in an unusable state.

Finding and disabling unnecessary processes

Ideally organization should know every process running on the system. To get a list of all processes, the command `ps -ef` (POSIX style) or `ps ax` (BSD style) can be used. Processes with square brackets around their names are kernel-level processes that perform helper functions (such as flushing the caches to disk); all other processes are user level. Even on your newly installed (minimal) system, a lot of processes are running.

Now let's look at those processes that open network connections; they hold the largest potential for attacks. To get a list of all TCP or UDP connections, use the command

netstat -atu (with name resolution, easier to read) or netstat -atun (without name resolution, quicker). Within this list, pay special attention to TCP connections with status LISTEN and to all UDP connections because these are servers that accept incoming connections.

Identify how these processes were started (via /etc/inittab, via boot scripts, etc.) and disable them as described earlier. This task can become a bit more challenging if the program was launched by another program

Audit preparation

Once a basic system is installed and securely configured, the ultimate goal is to keep the system secure. To identify unwanted changes to the system, auditing tools has to be used to record a fingerprint of the system in a hopefully known and secure state and detect changes to it.

This chapter provides detailed information about the various existing operating systems hardening tools and techniques. The chapter also highlights the existing method of hardening Windows XP and Linux and explains how to configure and enhance the security settings in these operating systems.

Chapter 3

Problem Statement

When an operating system is installed, by default many services are available; this can result in a configuration that is vulnerable to security exposures if the operating system is not configured appropriately. To minimize the number of possible security exposures, the system administrator must proactively configure the operating system for increased security and decreasing its susceptibility to compromise. Moreover, efficient security tools cannot work correctly or effectively when the operating system they operate under is vulnerable.

Operating System hardening is a global philosophy of system security that focuses strongly not only on detection, but also on prevention. It involves removing unnecessary services from the base operating system, restricting user access to the system, removing unnecessary applications and applying original patches from software vendors, enforcing password restrictions, controlling user and group rights, and enabling system accounting. Thus if the operating system provides strong hardening capability the security risks on the whole information system are significantly reduced, in turn reducing the inconvenience and cost involved in downtime and system rebuilds, and also damage to the reputation.

Every operating system has its own list of security features and thus provides different level of hardening capability. The security of information system, to a large extent, depends on the robustness of its operating system. After all, it is the operating system that is entrusted with preventing ill-behaved tasks from crashing the rest of the system, and providing each task with appropriate access to system resources. The operating system must be selected such that it meets the security needs the information system. Thus in selection of the operating system considering its hardening capability is quite important, so there is requirement of a method to quantitatively measure the hardening capability of the operating systems based on a common list of security features. Such a method will be quite helpful to administrators in properly selecting the operating system.

4.1 Proposed Security Hardening Features Categories

In our work we prepared a list of security hardening features for an operating system. Each hardening feature is related to some sort of security. All the suggested security options for an operating system can be categorized in following categories [7, 8]:

1. Authentication
2. Network Access and Communication
3. System Right and Access Controls
4. Services
5. Data Access and File System
6. Patch Management
7. System Auditing

4.1.1 Authentication

Authentication is any process by which it is determined that someone is who they claim they are. This usually involves a username and a password, but can include any other method of demonstrating identity, such as a smart card, retina scan, voice recognition, or fingerprints. Authentication is equivalent to showing your drivers license at the ticket counter at the airport. Authentication in Windows systems is accomplished by providing an account name and credentials that match the information stored in an account database. Possession of the necessary credentials enables both authorized and unauthorized individuals to access a system.

Organizations own users are often a major cause of authentication weaknesses. Creating a weak password, sharing credentials or writing down private account information are all ways in which sensitive information can be compromised, leading

to unauthorized logons and insider hacks. Proper logon policies must be in place and the authentication credentials hardened.

Weak Authentication such as LM, NTLM, NTLMv2 should be avoided. NTLM (NT LAN Manager) is a challenge/response form of authentication that was the default network authentication protocol in Windows NT 4.0. It's also used in Windows 2000 for compatibility with earlier Windows versions and to authenticate logons to standalone computers. Since the days of Windows NT, Microsoft has upgraded its default authentication protocol to Kerberos, a considerably more secure option than NTLM

Pluggable Authentication Modules (PAM) can be used to provide flexible authentication. PAM is an authentication mechanism that originated on Solaris, but is used on various systems, including Linux. The Linux PAM implementation allows a system administrator to choose how users authenticate to various services. New modules can be added by an administrator at any time, offering overall flexibility in how authentication happens.

4.1.2 Network Access and Communication

Three basic security processes can be used to harden network communications: authentication, integrity, and encryption. Computer *authentication* is essential in order to ensure that data is actually coming from and going to appropriate computers. If a communication can spoof its origination, or if a destination can be spoofed, then there is no way to know if the information is correct, and no way to avoid sending confidential information where it should not go. *Integrity* ensures that the data has not changed during transport. If integrity is not guaranteed, then an attacker might successfully change data. *Encryption* protects data by making the message useless to any but those possessing the key. While not every protocol designed for communications security does all three, the best protection for data communications will.

An additional security mechanism, *message signing*, can guarantee that a specific message came from the computer identified as the source of the message. As part of the negotiation process, the client and server are authenticated. If authentication fails, the communication does not proceed. If authentication is successful, each packet sent is signed by the source. Without message signing, session hijacking can occur. Session hijacking is an attack where communications are intercepted and modified en route.

The main reason to put a computer on a network is so that it can communicate with other computers. Computer security is often an afterthought when deploying a new server. Unfortunately, correctly configuring system security requires delicately balancing system access.

The best strategy to adopt when hardening any system is to limit machine-to-machine communication to just the necessary communications. The first step in limiting communication is to only allow a service to be enabled or running if it is fulfilling a requirement. The best time to configure services is right after installation.

4.1.3 System Rights and Access Controls

The reason for hardening operating systems is to provide secure platforms on which work can be done. Users of desktop systems must be able to run applications and to access and store data. Some users, such as those that work remotely, must also have the ability to modify some configuration settings. In hardening operating systems, there must be a balance between usability and security. To do so, users should be provided with the ability to do their job but restrict access to data, applications, and configuration settings.

System rights and Access control acts as the physical controls that prevent unauthorized access to systems. After a user is authenticated, operating system uses the authorization and access control technologies to determine if that user should be allowed to access a resource.

Microsoft defines access control technologies with five overriding principals: User-based authorization, discretionary access control, inheritance of permissions,

administrative privileges and auditing of system events. This access control model ensures authorized use of resources by security principals, which include users and groups that perform actions. In addition to security principals, access control components include security identifiers (SIDs), access tokens, permissions, user rights, security descriptors and access control lists (ACLs).

4.1.4 Services

Service Hardening restricts critical services from doing abnormal activities in the file system, registry, network, or other resources that could be used to allow malware to install itself or attack other computers. For example, the Remote Procedure Call (RPC) service can be restricted from replacing system files or modifying the registry.

Services are all more vulnerable because they typically start running when the operating system starts, and remain running until the operating system shuts down. Not only that, but many services run on pretty much every system, meaning that a worm has a huge number of computers available to it in order to propagate. In many ways, services represent the perfect attack surface for attackers. Services represent a large percentage of the overall attack surface in Windows. Today, many system and third-party services run in the LocalSystem account, where any breach could lead to unbounded damage to the local machine—including disk formatting, user data access, or driver installation.

Service Hardening reduces the damage potential of a compromised service. Windows Service Hardening provides an additional layer of protection for services based on the security principle of defense-in-depth. Windows Service Hardening cannot prevent a vulnerable service from being compromised, instead it limits how much damage an attacker can do in the unlikely event the attacker is able to identify and exploit a vulnerable service.

4.1.5 Data Access and File System

The major reason for hardening computers and networks is to protect the data that is stored on them. Sound perimeter, computer, and applications controls do assist in providing confidentiality and integrity for our data, hardening file system permissions and encryption can add another layer of defense.

Strong network access controls prevent unauthorized users from accessing a system. File access controls are important not only for preventing intruders from gaining too much access, but also to limit authorized users to their own data, and to maintain secure configuration and log files

File systems should be installed on secure machine with the highest form of file security. NTFS is a strong secure file system that let the administrator and user control access to files that have respective assigned permissions. This added level of control not only protects the system against potential intruder but also provides as a shield against viruses when they attempt to install them selves as a user that does not have rights to the hardware. It is a good idea to format all mobile machines with NTFS as this provides added security if the machine is stolen or misplaced.

4.1.6 Patch management

Patch management is the process of controlling the deployment and maintenance of interim software releases into production environments. It helps in maintaining operational efficiency and effectiveness, overcome security vulnerabilities, and maintain the stability of your production environment.

If an organization cannot determine and maintain a known level of trust within its operating systems and application software, it might have a number of security vulnerabilities, which, if exploited, could lead to a loss of revenue and intellectual property. Minimizing this threat requires to have properly configured systems, to use the latest software, and to install the recommended software updates.

There have been several widely-publicized attacks and vulnerabilities related to operating system or application software. Many organizations with proactive security patch management in place were not affected by these attacks, because they acted on information that vendors made available in advance of the attack. Proactive security patch management is an effective way to limit attacks that target known software vulnerabilities.

Patch management is a circular process and must be ongoing. The unfortunate reality about software vulnerabilities is that, after you apply a patch today, a new vulnerability must be addressed tomorrow. Develop and automate a patch management process that includes each of the following:

- **Detect.** Use tools to scan your systems for missing security patches. The detection should be automated and will trigger the patch management process.
- **Assess.** If necessary updates are not installed, determine the severity of the issue(s) addressed by the patch and the mitigating factors that may influence your decision. By balancing the severity of the issue and mitigating factors, you can determine if the vulnerabilities are a threat to your current environment.
- **Acquire.** If the vulnerability is not addressed by the security measures already in place, download the patch for testing.
- **Test.** Install the patch on a test system to verify the ramifications of the update against your production configuration.
- **Deploy.** Deploy the patch to production computers. Make sure your applications are not affected. Employ your rollback or backup restore plan if needed.
- **Maintain.** Subscribe to notifications that alert you to vulnerabilities as they are reported. Begin the patch management process again.

4.1.7 System Auditing

System auditing is an a posteriori technique for determining security violations. Auditing plays a major role in detection of security violations and in postmortem analysis to determine precisely what happened and how. This makes an effective

auditing subsystem a key security component of any system. By recording attempts to access objects on the system, and by recording security-sensitive changes to the system's configuration we can spot any unauthorized events or other security lapses. Using this information, any security holes can be plugged to prevent a recurrence.

System Auditing is the analysis of log records to present information about the system in a clear and understandable manner. With respect to computer security, logs provide a mechanism for analyzing the system security state, either to determine if a requested action will put the system in a nonsecure state or to determine the sequence of events leading to the system being in a nonsecure (compromised) state.

Auditing mechanism allows systems analysts to review patterns of usage in order to evaluate the effectiveness of protection mechanisms. These patterns can be used to establish expected patterns of resource usage, which are critical for some intrusion detection systems. Auditing mechanisms must record any use of privileges. A security control that would restrict an ordinary user may not restrict the empowered user. Finally, audit mechanisms deter attacks because of the record and the analysis, thereby providing assurance that any violation of security policies will be detected.

The decision of which events and actions should be audited requires knowledge of the security policy of the system, what attempts to violate that policy involve, and how such attempts can be detected. The question of how such attempts can be detected raises the question of what should be logged: what commands must an attacker use to (attempt to) violate the security policy, what system calls must be made, who must issue the commands or system calls and in what order, what objects must be altered, and so forth. Logging of all events implicitly provides all this information; the problem is how to discern which parts of the information are relevant, which is the problem of determining what to audit.

4.2 Security Hardening Features

The presence of the proposed security features in the operating system increases its hardening capability whereas their absence decreases the hardening capability. The proposed security features are for three categories: authentication, system rights and access control and Network Access and Communication are as follows.

4.2.1 Authentication Hardening Features

4.2.1.1 Maximum password age

This security feature determines the number of days that a password can be used before the user must change it.

The possible values for the **Maximum password age** setting are:

- A user-specified number of days between 0 and 999
- Not Defined

Vulnerability

Any password, even the most complex password, can be guessed (or "cracked") by an attacker with sufficient time and computer processing power. Some of the following policy settings can make it harder to crack passwords in a reasonable amount of time. The risk of a valid password being cracked can be reduced if users are frequently required to change their passwords, which can also mitigate the risk of unauthorized logon by someone who wrongfully acquired a password. The Maximum password age setting can be configured so that users are never required to change their passwords, but such a configuration would be a major security risk.

If the Maximum password age setting is too low, users will be required to change their passwords very often. Such a configuration may actually reduce security in the organization, because users may be more likely to write their passwords somewhere so that they do not forget them and then leave the information in an insecure location or lose it. If you configure the value for this policy setting too high, the level of security within an organization will be reduced because it will

allow potential attackers a much larger timeframe in which to crack user passwords or to use compromised accounts.

4.2.1.2 Minimum password age

This security feature determines the number of days that a password must be used before the user is allowed to change it. The Minimum password age value must be less than the Maximum password age value [19].

Configure this policy setting to a number that is greater than 0 if you want the Enforce password history setting to be effective. If you configure the Enforce password history setting to 0, the user is not required to choose a new unique password when they are prompted to change their password. If password history is used, users will have to enter a new unique password when they change their password.

The possible values for the Minimum password age setting are:

- A user-specified number of days between 0 and 998
- Not Defined

Vulnerability

It is ineffective to require users to change passwords regularly if they can cycle through passwords until they can reuse a favorite password. Use this policy setting with the Enforce password history setting to prevent the reuse of old passwords. For example, if you configure the Enforce password history setting to ensure that users cannot reuse any of their last 12 passwords, they could change their password 13 times in a few minutes and reuse the password they started with unless you configure the Minimum password age setting to a number that is greater than 0. You must configure this policy setting to a number that is greater than 0 for the Enforce password history setting to be effective.

4.2.1.3 Minimum password length

This security feature determines the least number of characters that may make up a password for a user account. There are many different theories about how to determine the best password length for an organization, but perhaps "pass phrase" is a better term than "password." In Microsoft Windows 2000 and later versions, pass

phrases can be quite long and they can include spaces, punctuation marks, and Unicode characters. Therefore, a phrase such as "I want to drink a \$5 beverage!" is a valid pass phrase. Such a phrase is considerably stronger than an 8 or 10 character string of random numbers and letters, but it is easier to remember.

The possible values for the Minimum password length setting are:

- A user-specified number between 0 and 14
- Not Defined

Vulnerability

There are several types of password attacks that can be performed to obtain the password for a particular user account. These attacks include dictionary attacks (which attempt to use common words and phrases) and brute force attacks (which try every possible combination of characters). Also, attackers sometimes try to obtain the account database so they can use utilities to crack the accounts and passwords.

This configuration will provide adequate defense against a brute force attack. Additional complexity requirements will help reduce the possibility of a dictionary attack. Complexity requirements are discussed in the next section of this chapter. You should also note that some countries have legal requirements with respect to password length.

Requirements for extremely long passwords can actually decrease the security of an organization, because users may be more likely to write their passwords somewhere so that they do not forget them and then leave the information in an insecure location or lose it. However, if users are taught that they can use pass phrases as described earlier, they should be able to easily remember them.

If short passwords are permitted, security will be reduced because they can be easily cracked with tools that perform either dictionary or brute force attacks. If very long passwords are required, mistyped passwords could cause account lockouts and subsequently increase the volume of help desk calls.

4.2.1.4 Enforce password history

This security feature determines the number of unique new passwords that must be associated with a user account before an old password can be reused.

The possible values for the Enforce password history setting are:

- A user-specified value between 0 and 24
- Not Defined

Vulnerability

Password reuse is an important concern in any organization. Many users will want to use or reuse the same password for their account over a long period of time. The longer the same password is in use for a particular account, the greater the chance that an attacker will be able to determine the password through brute-force attacks. Also, any accounts that may have been compromised will remain exploitable for as long as the password is left unchanged. If password changes are required but password reuse is not prevented, or if users can continually reuse a small number of passwords, the effectiveness of a good password policy is greatly reduced.

If you specify a low number for this policy setting, users will be able to use the same small number of passwords repeatedly. If you do not also configure the Minimum password age setting, users will be able to repeatedly change their passwords until they can reuse their original password.

The Enforce password history value should be set at a level that combines a reasonable maximum password age with a reasonable password change interval requirement for all users in your organization.

The major impact of this configuration is that users will have to create a new password every time they are required to change their old one. If users are required to change their passwords to new unique values, there is an increased risk of users who write their passwords somewhere so that they do not forget them. Another risk is that users may create passwords that change incrementally (for example, *password01*, *password02*, and so on) to facilitate memorization. Also, an excessively low value for

the Minimum password age setting will likely increase administrative overhead, because users who forget their passwords may require help desk assistance to reset them.

4.2.1.5 Passwords must meet complexity requirements

This security feature determines whether passwords must meet a series of guidelines that are considered important for a strong password.

If this policy setting is enabled, passwords must meet the following requirements:

- The password is at least six characters long.
- The password contains characters from three of the following four categories:
 - Uppercase characters (A, B, C, ...)
 - Lowercase characters (a, b, c, ...)
 - Numerals (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
 - Non-alphanumeric and Unicode characters (() ` ~ ! @ # \$ % ^ & * - + = | \ { } [] : ; " ' < > , . ? / € Γ f λ and space)
- The password does not contain three or more consecutive characters from the user's account name or display name.

These complexity requirements are enforced upon password change or creation of new passwords.

The possible values for the Passwords must meet complexity requirements setting are:

- Enabled
- Disabled
- Not Defined

Vulnerability

Passwords that contain only alphanumeric characters are extremely easy to crack with several publicly available utilities. To prevent passwords from being cracked, they should contain a wider range of characters.

When combined with a Minimum password length of 8, this policy setting ensures that the number of different possibilities for a single password is so great that it will be difficult (but not impossible) for a brute force attack to succeed. An

attacker who had enough processing power to test 1 million passwords per second could determine such a password in about seven and one-half days or less.

4.2.1.6 Store password using reversible encryption for all users

This security feature determines whether Microsoft Windows Server 2003, Windows 2000 Server, Windows 2000 Professional, and Windows XP Professional use reversible encryption when they store passwords.

The Store password using reversible encryption for all users setting provides support for application protocols that require knowledge of the user's password for authentication purposes. However, encrypted passwords that are stored in a way that is reversible can be decrypted. A knowledgeable attacker who was able to break this encryption could then log on to network resources with the compromised account.

Use of Challenge Handshake Authentication Protocol (CHAP) authentication through remote access or Internet Authentication Service (IAS) services requires that this policy setting be enabled. CHAP is an authentication protocol that can be used by Microsoft remote access and Network Connections.

The possible values for the Store password using reversible encryption for all users in the domain setting are:

- Enabled
- Disabled
- Not Defined

Vulnerability

This policy setting determines whether Windows Server 2003 will store passwords in a weaker format that is much more susceptible to compromise.

4.2.1.7 Allow changing passwords only when required

Users can change their passwords during the period between the minimum and maximum password age settings. However, the Specialized Security – Limited Functionality environment security design requires that users change their passwords only when prompted by the operating system after their passwords have reached the

maximum age of 42 days. To achieve this level of control, administrators can disable the Change Password... button in the Windows Security dialog box that appears when you press CTRL+ALT+DELETE.

This configuration can be implemented for an entire domain through Group Policy, or implement it for one or more specific users by editing the registry.

4.2.1.8 Account lockout duration

This security feature determines the number of minutes that a locked-out account remains locked out before it is automatically unlocked. The available range is from 1 to 99,999 minutes. To specify that the account will be locked out until an administrator manually unlocks it, configure the value to 0. If an account lockout threshold is defined, the Account lockout duration must be greater than or equal to the reset time.

The possible values for the Account lockout duration setting are:

- A user-defined value in minutes between 0 and 99,999
- Not Defined

Vulnerability

A denial of service (DoS) condition may be created if an attacker abuses the Account lockout threshold and repeatedly attempts to log on with a specific account. If you configure the Account lockout threshold setting, the account will be locked out after the specified number of failed attempts. If you configure the Account lockout duration setting to 0, then the account will remain locked out until an administrator unlocks it manually.

Configure the Account lockout duration setting to an appropriate value for the environment. To specify that the account will remain locked until an administrator manually unlocks it, configure the value to 0. When the Account lockout duration setting is configured to a non-zero value, automated attempts to guess account passwords must wait out this interval before they resume attempts against a specific account. If you use this setting in combination with the Account lockout threshold setting, such automated password guessing attempts can be made difficult or useless.

4.2.1.9 Account lockout threshold

This security feature determines the number of failed logon attempts that causes a user account to become locked out. A locked-out account cannot be used until it is reset by an administrator or until the lockout duration for the account expires. You can specify up to 999 failed logon attempts, or you can set the value to 0 to specify that the account will never be locked out. If you define an Account lockout threshold, then the Account lockout duration must be greater than or equal to the reset time.

Failed password attempts against workstations or member servers that have been locked using either CTRL+ALT+DELETE or password-protected screen savers do not count as failed logon attempts unless the policy setting Interactive logon: Require Domain Controller authentication to unlock workstation is enabled. If it is, then repeated failed password attempts to unlock the workstation will count against the account lockout threshold. The possible values for the Account lockout threshold setting are:

- A user-defined value between 0 and 999
- Not Defined

Vulnerability

Password attacks may use automated methods to try thousands or even millions of password combinations for any or all user accounts. The effectiveness of such attacks can be almost eliminated if you limit the number of failed logons that can be performed.

However, it is important to note that a DoS attack could be performed on a domain that has an account lockout threshold configured. A malicious attacker could programmatically attempt a series of password attacks against all users in the organization. If the number of attempts is greater than the account lockout threshold, the attacker may be able to lock out every account.

If this policy setting is enabled, a locked-out account will not be usable until it is reset by an administrator or until the account lockout duration expires. This setting will

likely generate a number of additional help desk calls. In fact, locked accounts cause the greatest number of calls to the help desk in many organizations.

If the Account Lockout Threshold is configured to **0**, there is a possibility that an attacker's attempt to crack passwords with a brute force password attack may go undetected if a robust audit mechanism is not in place.

4.2.1.10 Reset account lockout counter after

This security feature determines the number of minutes that must elapse before the counter that tracks failed logon attempts and triggers account lockouts is reset to **0**. If an Account lockout threshold is defined, this reset time must be less than or equal to the Account lockout duration setting configuration.

The possible values for the Reset account lockout counter after setting are:

- A user-defined number of minutes between 1 and 99,999
- Not Defined

Vulnerability

Users can accidentally lock themselves out of their accounts if they mistype their password multiple times. To reduce the chance of such accidental lockouts, the Reset account lockout counter after setting determines the number of minutes that must elapse before the counter that tracks failed logon attempts and triggers lockouts is reset to 0.

If this security feature is not set or if the value is configured to an interval that is too long, a DoS attack could occur. An attacker could maliciously attempt to log on to each user's account numerous times and lock out their accounts as described in the preceding paragraphs.

4.2.1.11 Use of Blank Passwords to Console Logon

This security feature determines whether remote interactive logons by network services such as Terminal Services, Telnet, and File Transfer Protocol (FTP) are allowed for local accounts that have blank passwords. If you enable this policy

setting, a local account must have a non-blank password to perform an interactive or network logon from a remote client.

The possible values for the Limit local account use of blank passwords to console logon only setting are:

- Enabled
- Disabled
- Not Defined

Vulnerability

Blank passwords are a serious threat to computer security and should be forbidden through both organizational policy and suitable technical measures. In fact, the default settings for Windows Server 2003 Active Directory® directory service domains require complex passwords of at least seven characters. However, if a user with the ability to create new accounts bypasses your domain-based password policies, they could create accounts with blank passwords. For example, a user could build a stand-alone computer, create one or more accounts with blank passwords, and then join the computer to the domain. The local accounts with blank passwords would still function. Anyone who knows the name of one of these unprotected accounts could then use it to log on.

4.2.1.12 Machine account password changes

This security feature determines whether or not a domain controller will accept password change requests for computer accounts.

The possible values for the machine account password changes setting are:

- Enabled
- Disabled
- Not Defined

Vulnerability

If you enable this policy setting on all domain controllers in a domain, domain members will not be able to change their computer account passwords, and those passwords will be more susceptible to attack.

4.2.1.13 Do not display last user name

This security feature determines whether the Log on to Windows dialog box displays the name of the last user to log on to the computer. If administrator enables this policy setting, the name of the last user to successfully log on does not display. If administrator disable this policy setting, the name of the last user to log on will display.

The possible values for the Do not display last user name setting are:

- Enabled
- Disabled
- Not Defined

Vulnerability

An attacker with access to the console (for example, someone with physical access or someone who is able to connect to the server through Terminal Services) could view the name of the last user who logged on to the server. The attacker could then try to guess the password, use a dictionary, or use a brute-force attack to try and log on.

4.2.1.14 Number of previous logons to cache

This security feature determines the number of different unique users who can log on to a Windows domain by using cached account information. Logon information for domain accounts can be cached locally so that if a domain controller cannot be contacted on subsequent logons, a user can still log on. This security feature determines the number of unique users whose logon information is cached locally.

The possible values for the Number of previous logons to cache (in case domain controller is not available) setting are:

- A user-defined number between 0 and 50
- Not Defined

Vulnerability

The number that is assigned to this policy setting indicates the number of users whose logon information the servers will cache locally. If the number is set to 10, then the server caches logon information for 10 users. When an eleventh user logs on to the computer, the server overwrites the oldest cached logon session.

Users who access the server console will have their logon credentials cached on that server. An attacker who is able to access the file system of the server could locate this cached information and use a brute force attack to attempt to determine user passwords.

To mitigate this type of attack, Windows encrypts the information and obscures its physical location.

4.2.1.15 Require Domain Controller authentication to unlock workstation

Logon information is required to unlock a locked computer. For domain accounts, the Require Domain Controller authentication to unlock workstation setting determines whether it is necessary to contact a domain controller to unlock a computer. If this setting is enabled, a domain controller must authenticate the domain account that is being used to unlock the computer. If this setting is disabled, logon information confirmation with a domain controller is not required for a user to unlock the computer. However, if the administrator configure the Number of previous logons to cache (in case domain controller is not available) setting to a value that is greater than zero, then the user's cached credentials will be used to unlock the computer.

The possible values for the Require Domain Controller authentication to unlock workstation setting are:

- Enabled
- Disabled
- Not Defined

Vulnerability

By default, the computer caches in memory the credentials of any users who are authenticated locally. The computer uses these cached credentials to authenticate anyone who attempts to unlock the console. When cached credentials are used, any changes that have recently been made to the account—such as user rights assignments, account lockout, or the account being disabled—are not considered or applied after the account is authenticated. User privileges are not updated, and (more importantly) disabled accounts are still able to unlock the console of the computer.

When the console on a computer is locked, either by a user or automatically by a screen saver time-out, the console can only be unlocked if the user is able to re-authenticate to the domain controller. If no domain controller is available, then users cannot unlock their workstations. If the Number of previous logons to cache (in case domain controller is not available) setting is configured to 0, users whose domain controllers are unavailable (such as mobile or remote users) will not be able to log on.

4.2.1.16 Do not allow storage of credentials or .NET Passports for network authentication

This security feature determines whether the Stored User Names and Passwords feature may save passwords or credentials for later use when it gains domain authentication. If this policy setting is enabled, the Stored User Names and Passwords feature of Windows does not store passwords and credentials.

The possible values for the Do not allow storage of credentials or .NET Passports for network authentication setting are:

- Enabled
- Disabled
- Not Defined

Vulnerability

Passwords that are cached can be accessed by the user when logged on to the computer. Although this information may sound obvious, a problem can arise if the user unknowingly executes hostile code that reads the passwords and forwards them to the attackers.

4.2.1.17 Pluggable different authentication for different services

The Linux PAM implementation allows a system administrator to choose how users authenticate to various services. New modules can be added by an administrator at any time, offering overall flexibility in how authentication happens. Many experts believe that having a single authentication mechanism for each service on the system (terminal logins, local logins, network logins, etc.) is too inflexible. Typically, each service needed its own authentication code or had to use the single mechanism available. PAM is the answer to inflexible single authentication mechanisms. PAM allows different modules to be added for authenticating new services and for adding new authentication mechanisms for old services. PAM can also be used to enable shadow file authentication for applications that don't natively support it [15,16].

Vulnerability

Traditionally, Linux and other Unix-like systems simply authenticated users against an entry in the file `/etc/passwd`. Everyone had read-only access to the password file, and the encrypted passwords were available to anyone with access to the system. This simple design made password files vulnerable to “dictionary attacks,” an attack where the attacker would encrypt common words and compare his encrypted words with what was in the password file. If a match was found, the attacker then would know the password.

4.2.2 System Right and Access Controls Hardening Features

4.2.2.1 Load and unload device drivers

This security feature determines whether users can dynamically load and unload device drivers. This user right is not required if a signed driver for the new hardware already exists in the Driver.cab file on the computer.

The possible values for the Load and unload device drivers setting are:

- A user-defined list of accounts
- Not Defined

Vulnerability

Device drivers run as highly privileged code. A user who has the Load and unload device drivers user right could unintentionally install malicious code that masquerades as a device driver. Administrators should exercise greater care and install only drivers with verified digital signatures.

4.2.2.2 Change the system time

This security feature determines whether users can adjust the time on the computer's internal clock. It is not required to change the time zone or other display characteristics of the system time.

The possible values for the Change the system time setting are:

- A user-defined list of accounts
- Not Defined

Vulnerability

Users who can change the time on a computer could cause several problems. For example, time stamps on event log entries could be made inaccurate, time stamps on files and folders that are created or modified could be incorrect, and computers that belong to a domain may not be able to authenticate themselves or users who try to log on to the domain from them. Also, because the Kerberos authentication protocol requires that the requestor and authenticator have their clocks synchronized within an administrator-defined skew period, an attacker who

changes a computer's time may cause that computer to be unable to obtain or grant Kerberos tickets.

The risk from these types of events is mitigated on most domain controllers, member servers, and end-user computers because the Windows Time service automatically synchronizes time with domain controllers in the following ways:

- All client desktop computers and member servers use the authenticating domain controller as their inbound time partner.
- All domain controllers in a domain nominate the primary domain controller (PDC) emulator operations master as their inbound time partner.
- All PDC emulator operations masters follow the hierarchy of domains in the selection of their inbound time partner.
- The PDC emulator operations master at the root of the domain is authoritative for the organization. Therefore it is recommended that you configure this computer to synchronize with a reliable external time server.

This vulnerability becomes much more serious if an attacker is able to change the system time and then stop the Windows Time service or reconfigure it to synchronize with a time server that is not accurate.

4.2.2.3 Manage auditing and security log

This security feature determines whether users can specify object access audit options for individual resources such as files, Active Directory objects, and registry keys. Object access audits are not performed unless you enable them through Audit Policy, which is located under Security Settings, Local Policies. A user who is assigned this user right can also view and clear the Security event log from Event Viewer.

The possible values for the Manage auditing and security log setting are:

- A user-defined list of accounts
- Not Defined

Vulnerability

The ability to manage the Security event log is a powerful user right and it should be closely guarded. Anyone with this user right can clear the Security log to erase important evidence of unauthorized activity.

4.2.2.4 Log on as a service

This security feature determines whether users can log on as a service.

The possible values for the Deny log on as a service setting are:

- A user-defined list of accounts
- Not Defined

Vulnerability

Accounts that can log on as a service could be used to configure and launch new unauthorized services, such as a keylogger or other malware. The benefit of the specified countermeasure is somewhat mitigated by the fact that only users with administrative privileges can install and configure services, and an attacker who has already attained that level of access could configure the service to run with the System account.

If the Deny log on as a service user right is assigned to specific accounts, services may not be able to start and a DoS condition could result.

4.2.2.5 Force shutdown from a remote system

This security feature determines whether a user can shut down a computer from a remote location on the network.

The possible values for the Force shutdown from a remote system setting are:

- A user-defined list of accounts
- Not Defined

Vulnerability

Any user who can shut down a computer could cause a DoS condition to occur. Therefore, this user right should be tightly restricted.

Restrict the Force shutdown from a remote system user right to members of the Administrators group or other specifically assigned roles that require this capability (such as non-administrative operations center staff).

4.2.2.6 Act as part of the operating system

This security feature determines whether a process can assume the identity of any user and thereby gain access to the resources that the user is authorized to access. Typically, only low-level authentication services require this user right. Note that potential access is not limited to what is associated with the user by default. The calling process might request that arbitrary additional privileges be added to the access token. The calling process might also build an access token that does not provide a primary identity for auditing in the system event logs.

The possible values for the Act as part of the operating system setting are:

- A user-defined list of accounts
- Not Defined

Vulnerability

The Act as part of the operating system user right is extremely powerful. Anyone with this user right can take complete control of the computer and erase evidence of their activities.

Restrict the Act as part of the operating system user right to as few accounts as possible—it should not even be assigned to the Administrators group under typical circumstances. When a service requires this user right, configure the service to log on with the Local System account, which has this privilege inherently. Do not create a separate account and assign this user right to it.

4.2.2.7 Adjust memory quotas for a process

This security feature determines whether users can adjust the maximum amount of memory that is available to a process. Although this capability is useful when you

need to tune computers, you should consider its potential for abuse. In the wrong hands, it could be used to launch a denial of service (DoS) attack.

The possible values for the Adjust memory quotas for a process setting are:

- A user-defined list of accounts
- Not Defined

Vulnerability

A user with the Adjust memory quotas for a process privilege can reduce the amount of memory that is available to any process, which could cause business-critical network applications to become slow or to fail.

4.2.2.8 Back up files and directories

This security feature determines whether users can circumvent file and directory permissions to back up the computer. This user right is effective only when an application attempts access through the NTFS backup application programming interface (API) through a backup utility such as NTBACKUP.EXE. Otherwise, standard file and directory permissions apply.

The possible values for the Back up files and directories setting are:

- A user-defined list of accounts
- Not Defined

Vulnerability

Users who are able to back up data from a computer could take the backup media to a non-domain computer on which they have administrative privileges and restore the data. They could take ownership of the files and view any unencrypted data that is contained within the backup set.

4.2.2.9 Debug programs

This security feature determines whether users can open or attach to any process, even those that they do not own. This user right provides access to sensitive and critical operating system components.

The possible values for the Debug programs setting are:

- A user-defined list of accounts

- Not Defined

Vulnerability

The Debug programs user right can be exploited to capture sensitive computer information from system memory, or to access and modify kernel or application structures. Some attack tools exploit this user right to extract hashed passwords and other private security information, or to effect rootkit code insertions. By default, the Debug programs user right is assigned only to administrators, which helps to mitigate the risk from this vulnerability.

4.2.2.10 Access to this computer from the network

This security feature determines whether users have the ability to connect to the computer from the network.

The possible values for the access to this computer from the network setting are:

- A user-defined list of accounts
- Not Defined

Vulnerability

Users who can log on to the computer over the network can enumerate lists of account names, group names, and shared resources. Users with permission to access shared folders and files can connect over the network and possibly view or modify data. You can explicitly deny this user right to high-risk accounts (such as the local Guest account and other accounts that have no business need to access the computer over the network) to provide an additional layer of protection.

Assign the Deny access to this computer from the network user right to the following accounts:

- ANONYMOUS LOGON
- The built-in local Administrator account
- The local Guest account
- The built-in Support account
- All service accounts

4.2.2.11 Lock pages in memory

This security feature determines whether a process can keep data in physical memory, which prevents the computer from paging the data to virtual memory on disk. If you assign this user right, significant degradation of computer performance can occur.

The possible values for the Lock pages in memory setting are:

- A user-defined list of accounts
- Not Defined

Vulnerability

Users with the Lock pages in memory user right could assign physical memory to several processes, which could leave little or no RAM for other processes and result in a DoS condition.

4.2.2.12 Modify firmware environment values

This security feature determines whether users can modify system environment variables by either a process through an API or by a user through System Properties.

The possible values for the Modify firmware environment values setting are:

- A user-defined list of accounts
- Not Defined

Vulnerability

Anyone who is assigned the Modify firmware environment values user right could configure the settings of a hardware component to cause it to fail, which could lead to data corruption or a DoS condition.

4.2.2.13 Increase scheduling priority

This security feature determines whether users can increase the base priority class of a process. (It is not a privileged operation to increase relative priority within a priority class.) This user right is not required by administrative tools that are supplied with the operating system but might be required by software development tools.

The possible values for the Increase scheduling priority setting are:

- A user-defined list of accounts

- Not Defined

Vulnerability

A user who is assigned this user right could increase the scheduling priority of a process to Real-Time, which would leave little processing time for all other processes and could lead to a DoS condition.

4.2.2.14 Perform volume maintenance tasks

This security feature determines whether non-administrative or remote users can perform volume or disk management tasks, such as defragment an existing volume, create or remove volumes, and run the Disk Cleanup tool. Windows Server 2003 checks for this user right in a user's access token when a process that runs in the user's security context calls SetFileValidData().

The possible values for the Perform volume maintenance tasks setting are:

- A user-defined list of accounts
- Not Defined

Vulnerability

A user who is assigned the Perform volume maintenance tasks user right could delete a volume, which could result in the loss of data or a DoS condition.

4.2.2.15 Profile single process

This security feature determines whether users can sample the performance of an application process. Typically, you do not need this user right to use the Microsoft Management Console (MMC) Performance snap-in. However, you do need this user right if System Monitor is configured to collect data through Windows Management Instrumentation (WMI).

The possible values for the Profile single process setting are:

- A user-defined list of accounts
- Not Defined

Vulnerability

The Profile single process user right presents a moderate vulnerability. An attacker with this user right could monitor a computer's performance to help identify critical processes that they might wish to attack directly. The attacker may also be able to determine what processes run on the computer so that they could identify countermeasures that they may need to avoid, such as antivirus software, an intrusion-detection system, or which other users are logged on to a computer.

4.2.2.16 Profile system performance

This security feature determines whether a user can sample the performance of computer system processes. This privilege is required by the MMC Performance snap-in only if it is configured to collect data through WMI. Typically, you do not need this user right to use the Performance snap-in. However, you do need this user right if System Monitor is configured to collect data through WMI.

The possible values for the Profile system performance setting are:

- A user-defined list of accounts
- Not Defined

Vulnerability

The Profile system performance user right a moderate vulnerability. An attacker with this user right could monitor a computer's performance to help identify critical processes that they might wish to attack directly. The attacker may also be able to determine what processes are active on the computer so that they could identify countermeasures that they may need to avoid, such as antivirus software or an intrusion detection system.

4.2.2.17 Restore files and directories

This security feature determines whether a user can circumvent file and directory permissions when they restore backed up files and directories and whether they can set any valid security principal as the owner of an object.

The possible values for the Restore files and directories setting are:

- A user-defined list of accounts
- Not Defined

Vulnerability

An attacker with the Restore files and directories user right could restore sensitive data to a computer and overwrite data that is more recent, which could lead to loss of important data, data corruption, or a denial of service. An attacker could overwrite executable files that are used by legitimate administrators or system services with versions that include malicious code to grant themselves elevated privileges, compromise data, or install backdoors for continued access to the computer.

4.2.2.18 Take ownership of files or other objects

This security feature determines whether a user can take ownership of any securable object in the computer, including Active Directory objects, NTFS files and folders, printers, registry keys, services, processes, and threads.

The possible values for the Take ownership of files or other objects setting are:

- A user-defined list of accounts
- Not Defined

Vulnerability

Any user with the Take ownership of files or other objects user right can take control of any object, regardless of the permissions on that object, and then make any changes they wish to that object. Such changes could result in exposure of data, corruption of data, or a DoS condition.

4.2.2.19 Allowed to format and eject removable media

This security feature determines who is allowed to format and eject removable media.

The possible values for the Allowed to format and eject removable media setting are:

- Administrators
- Administrators and Power Users
- Administrators and Interactive Users
- Not Defined

Vulnerability

Users may be able to move data on removable disks to a different computer where they have administrative privileges. The user could then take ownership of any

file, grant themselves full control, and view or modify any file. The fact that most removable storage devices will eject media by pressing a mechanical button diminishes the advantage of this policy setting.

4.2.2.20 Restriction Policy for Software

Software restriction policy provides administrators with a way to identify software and control its ability to run on local computers. This tool can help protect computers that run Microsoft® Windows® XP Professional against known conflicts and safeguard them against malicious software such as viruses and Trojan horse programs. Software restriction policy integrates fully with the Active Directory® directory service and Group Policy.

When implementing software restriction policy, the first decision that must be taken is to decide whether the default security level will be **Unrestricted** or **Disallowed**. If the default security level is **Unrestricted**, then all software will be allowed to run and additional rules are needed to be configured to block specific applications. The more secure approach is to configure the default security level to **Disallowed**, which means no software will be allowed to run, and then configure additional rules to allow specific applications. Software restriction policy can be applied to multiple computers through domain-based Group Policies or to individual computers through local Group Policy.

Software restriction policy provides a number of ways to identify software, as well as a policy-based infrastructure to enforce rules on how the identified software may run. Computer users must comply with the guidelines that are established in the software restriction policy by the administrator in their environment.

Administrator can use software restriction policy to accomplish the following:

- Control what software may run on the client computers in your environment.
- Restrict user access to specific files on multi-user computers.
- Decide who may add trusted publishers to client computers.
- Define whether the policies affect all users or a subset of users on the client computers.

- Prevent executable files from running on your local computers based on policies that are set at a computer, OU, site, or domain level.

4.2.2.21 Create a token object

This policy setting determines whether a process can create a token, which it can then use to gain access to any local resources when the process uses `NtCreateToken()` or other token-creation APIs.

The possible values for the **Create a token object** setting are:

- A user-defined list of accounts
- Not Defined

Vulnerability

The operating system examines a user's access token to determine the level of the user's privileges. Access tokens are built when users log on to the local computer or connect to a remote computer over a network. When you revoke a privilege, the change is immediately recorded, but the change is not reflected in the user's access token until the next time the user logs on or connects. A user with the ability to create or modify tokens can change the level of access for any currently logged on account. They could escalate their own privileges or create a DoS condition.

4.2.3 Network Access and Communication Hardening Features

4.2.3.1 Digitally sign communications (four related settings)

There are four separate settings that relate to digital signatures for Server Message Block (SMB) communications:

- Microsoft Network Client: Digitally Sign Communications (Always)
- Microsoft Network Server: Digitally Sign Communications (Always)
- Microsoft Network Client: Digitally Sign Communications (If Server Agrees)
- Microsoft Network Server: Digitally Sign Communications (If Client Agrees)

The possible values for each of these policy settings are:

- Enabled
- Disabled

- Not Defined

Vulnerability

Implementation of digital signatures in high security networks helps to prevent the impersonation of clients and servers. This type of impersonation is known as session hijacking, and uses tools that allow attackers who have access to the same network as the client or server to interrupt, end, or steal a session in progress. Attackers can potentially intercept and modify unsigned SMB packets and then modify the traffic and forward it so that the server might perform undesirable actions. Alternatively, the attacker could pose as the server or client after legitimate authentication and gain unauthorized access to data.

SMB is the resource sharing protocol that is supported by many Microsoft operating systems. It is the basis of NetBIOS and many other protocols. SMB signatures authenticate both users and the servers that host the data. If either side fails the authentication process, data transmission will not take place.

The Windows 2000 Server, Windows 2000 Professional, Windows Server 2003, and Windows XP Professional implementations of the SMB file and print sharing protocol support mutual authentication, which prevents session hijacking attacks and supports message authentication to prevent man-in-the-middle attacks. SMB signing provides this authentication by placing a digital signature into each SMB, which is then verified by both the client and the server. Implementation of SMB signing may negatively affect performance, because each packet needs to be signed and verified. If you configure computers to ignore all unsigned SMB communications, older applications and operating systems will not be able to connect. If you completely disable all SMB signing, computers will be vulnerable to session hijacking attacks.

4.2.3.2 Send unencrypted password to third-party SMB servers

This security feature allows the SMB redirector to send plaintext passwords to non-Microsoft SMB servers that do not support password encryption during authentication.

The possible values for the Send unencrypted password to third-party SMB servers setting are:

- Enabled
- Disabled
- Not Defined

Vulnerability

If you enable this policy setting, the server can transmit passwords in plaintext across the network to other computers that offer SMB services. These other computers may not use any of the SMB security mechanisms that are included with Windows Server 2003.

4.2.3.3 Do not allow anonymous enumeration of SAM accounts

This security feature determines what additional permissions will be granted for anonymous connections to the computer. Windows allows anonymous users to perform certain activities, such as enumerate the names of domain accounts and network shares. This capability is convenient, for example, when an administrator wants to grant access to users in a trusted domain that does not maintain a reciprocal trust. However, even if this setting enabled, anonymous users will still have access to any resources that have permissions that explicitly include the special built-in group ANONYMOUS LOGON.

The possible values for the Do not allow anonymous enumeration of SAM accounts setting are:

- Enabled
- Disabled
- Not Defined

Vulnerability

An unauthorized user could anonymously list account names and use the information to perform social engineering attacks or attempt to guess passwords. (Social engineering attacks try to deceive users in some way to obtain passwords or some form of security information.)

4.2.3.4 Digitally encrypt or sign secure channel data (multiple related settings)

The following security settings determine whether a secure channel can be established with a domain controller that cannot sign or encrypt secure channel traffic:

- Domain member: Digitally encrypt or sign secure channel data (always)
- Domain member: Digitally encrypt secure channel data (when possible)
- Domain member: Digitally sign secure channel data (when possible)

If you enable the Domain member: Digitally encrypt or sign secure channel data (always) setting, a secure channel cannot be established with any domain controller that cannot sign or encrypt all secure channel data.

To protect authentication traffic from man-in-the-middle, replay, and other types of network attacks, Windows-based computers create a communication channel through NetLogon called Secure Channels. These channels authenticate computer accounts, and they also authenticate user accounts when a remote user connects to a network resource and the user account exists in a trusted domain. This authentication is called pass-through authentication, and it allows a computer that has joined a domain to have access to the user account database in its domain and in any trusted domains.

If you enable the Domain member: Digitally encrypt or sign secure channel data (always) setting, the Domain member: Digitally sign secure channel data (when possible) setting is automatically enabled.

The possible values for this policy setting are:

- Enabled
- Disabled
- Not Defined

Vulnerability

When a Windows Server 2003, Windows XP, Windows 2000, or Windows NT computer joins a domain, a computer account is created. After it joins the domain, the computer uses the password for that account to create a secure channel with the domain controller for its domain every time that it restarts. Requests that are sent on the secure channel are authenticated—and sensitive information such as passwords are encrypted—but the channel is not integrity-checked, and not all

information is encrypted. If a computer is configured to *always* encrypt or sign secure channel data but the domain controller cannot sign or encrypt any portion of the secure channel data, the computer and domain controller cannot establish a secure channel. If the computer is configured to encrypt or sign secure channel data *when possible*, a secure channel can be established, but the level of encryption and signing is negotiated.

4.2.3.5 Strong key protection for user keys stored on the computer

This security feature determines whether users can use private keys, such as their S/MIME key, without a password.

The possible values for the Force strong key protection for user keys stored on the computer setting are:

- User input is not required when new keys are stored and used
- User is prompted when the key is first used
- User must enter a password each time they use a key
- Not Defined

Vulnerability

You can configure this policy setting so that users must provide a password that is distinct from their domain password every time they use a key. This configuration makes it more difficult for an attacker to access locally stored user keys, even if the attacker takes control of the user's computer and determines their logon password.

4.2.3.6 Use FIPS compliant algorithms for encryption, hashing, and signing

This security feature determines whether the TLS/SSL Security Provider will only support the strong cipher suite known as TLS_RSA_WITH_3DES_EDE_CBC_SHA, which means that the provider only supports the TLS protocol as a client and as a server, if applicable. It uses only the Triple Data Encryption Standard (DES) encryption algorithm for the TLS traffic encryption, only the Rivest-Shamir-Adleman (RSA) public key algorithm for the TLS key exchange and authentication, and only the Secure Hash Algorithm version 1 (SHA-1) hashing algorithm for the TLS hashing requirements.

When this setting is enabled, the Encrypting File System Service (EFS) supports only the Triple DES encryption algorithm for encrypting file data. By default, the Windows Server 2003 implementation of EFS uses the Advanced Encryption Standard (AES) with a 256-bit key. The Windows XP implementation uses DESX.

The possible values for the Use FIPS compliant algorithms for encryption, hashing, and signing setting are:

- Enabled
- Disabled
- Not Defined

Vulnerability

You can enable this policy setting to ensure that the computer will use the most powerful algorithms that are available for digital encryption, hashing and signing. Use of these algorithms will minimize the risk of compromise of digitally encrypted or signed data by an unauthorized user.

4.2.3.7 Shares that can be accessed anonymously

This security feature determines which network shares can be accessed by anonymous users.

The possible values for the Shares that can be accessed anonymously setting are:

- A user-defined list of shares
- Not Defined

Vulnerability

It is very dangerous to enable this setting. Any shares that are listed can be accessed by any network user, which could lead to the exposure or corruption of sensitive data.

4.2.3.8 Force logoff when logon hours expire

This security feature determines whether to disconnect users who are connected to the local computer outside their user account's valid logon hours. It affects the SMB component. If you enable this policy setting, client sessions with the SMB server will

be disconnected when the client's logon hours expire. If you disable this policy setting, established client sessions will be maintained after the client's logon hours expire.

The possible values for the Force logoff when logon hours expire setting are:

- Enabled
- Disabled
- Not Defined

Vulnerability

If you disable this policy setting, a user could remain connected to the computer outside of their allotted logon hours.

4.2.3.9 Disable remote Desktop Sharing

This security feature allows you to disable the remote desktop sharing feature of NetMeeting. You can enable this policy setting so that users cannot configure NetMeeting to automatically answer inbound calls and allow remote control of the local desktop.

The possible values for the Disable remote Desktop Sharing setting are:

- Enabled
- Disabled
- Not Configured

Vulnerability

When this security setting is enabled, users will not be able to use the remote desktop sharing feature of NetMeeting. Users will be unable to configure remote desktop sharing through NetMeeting, although they may still be able to use the Windows Remote Assistance and Remote Desktop features if you have left them enabled.

4.2.3.10 Make proxy settings per-machine (rather than per-user)

If you enable this security setting, users will not be able to alter user-specific proxy settings. They must use the zones that are created for all users of the computers they access.

The possible values for the Make proxy settings per-machine (rather than per-user) setting are:

- Enabled
- Disabled
- Not Configured

Vulnerability

If you disable or do not configure this policy setting, users of the same computer will be able to establish their own proxy settings. This policy setting is intended to ensure that proxy settings remain uniformly in effect on the same computer and do not vary from user to user.

4.2.3.11 Do not allow users to add/delete sites

This security feature allows you to disable the site management settings for security zones. (To see the site management settings for security zones, select Tools and then Internet Options from the menu bar in Internet Explorer. Click the Security tab, and then click **Sites**.) If you disable or do not configure this setting, users will be able to add or remove Web sites in the Trusted Sites and Restricted Sites zones. They will also be able to alter settings in the Local Intranet zone.

The possible values for the Do not allow users to add/delete sites setting are:

- Enabled
- Disabled
- Not Configured

Vulnerability

If you do not configure this policy setting, users will be able to add or remove sites from the Trusted Sites and Restricted Sites zones at will and change settings in the Local Intranet zone. This configuration could allow sites that host malicious mobile code to be added to these zones, which users could execute.

Users will not be able to change site management settings for security zones that have been established by the administrator. When users need to add or remove

sites from these Internet Explorer security zones, an administrator will have to configure them.

4.2.3.12 Do not allow users to change policies

This security feature allows you to effectively disable the **Custom Level** button and Security level for the zone slider on the **Security** tab in the **Internet Options** dialog box. If this policy setting is disabled or not configured, users will be able to change the security zone settings. This policy setting can be used to prevent changes to the security zone policy settings that are established by the administrator.

The possible values for the **Security Zones: Do not allow users to change policies** setting are:

- Enabled
- Disabled
- Not Configured

Vulnerability

Users who change their Internet Explorer security settings could enable the execution of dangerous types of code from the Internet and Web sites that were listed in the Restricted Sites zone in the browser.

4.2.3.13 Allow software to run or install even if the signature is invalid

This security setting allows managing whether downloaded software can be installed or run by users if the signature is invalid. An invalid signature might indicate that someone tampered with the file. If this policy setting is enabled, users will be prompted to install or run files with an invalid signature. If this policy setting is disabled, users cannot run or install files with an invalid signature.

The possible values for the **Allow software to run or install even if the signature is invalid** setting are:

- Enabled
- Disabled
- Not Configured

Vulnerability

Microsoft ActiveX® controls and file downloads often have digital signatures attached that certify the file's integrity and the identity of the signer (creator) of the software. Such signatures help ensure that unmodified software is downloaded and that you can positively identify the signer to determine whether you trust them enough to run their software. The validity of unsigned code cannot be ascertained.

4.2.3.14 Allow active content from CDs to run on user machines

This policy setting allows you to determine whether active content on CDs can run on users' computers. Security-conscious organizations may wish to prevent the execution of ActiveX controls or other active content that is delivered on a CD.

The possible values for the Allow active content from CDs to run on user machines setting are:

- Enabled
- Disabled
- Not Configured

Vulnerability

Users may accidentally circumvent an organization's security policy if they run content that was delivered on a CD instead of over the network.

4.2.3.15 Allow third-party browser extensions

Users may install third-party browser extensions, which are known as browser helper objects (BHOs). The Allow third-party browser extensions setting controls whether installed BHOs will load when Internet Explorer starts.

The possible values for the Allow third-party browser extensions setting are:

- Enabled
- Disabled
- Not Configured

Vulnerability

Third-party browser extensions are potentially dangerous, because they may contain vulnerabilities or even malicious code. Also, their installation may violate organizational security policies.

4.2.3.16 Check for server certificate revocation

When a Secure Sockets Layer (SSL) connection is established between the browser and a remote server, the server presents a certificate to the client computer to use in the initial security negotiation. When the Check for server certificate revocation setting is enabled, Internet Explorer will determine whether the presented certificate is on the issuing authority's certificate revocation list.

The possible values for the Check for server certificate revocation setting are:

- Enabled
- Disabled
- Not Configured

Vulnerability

Users may accidentally communicate with a server whose certificate has expired or been revoked by the issuing authority. Such an occurrence may lead to information disclosure or even active attacks if the remote server has been compromised.

4.2.3.17 Check for signatures on downloaded programs

Downloaded programs may be signed with Microsoft Authenticode® technology, which binds a digital signature to executable objects such as programs and ActiveX controls. When the Check for signatures on downloaded programs setting is enabled, Internet Explorer will check the digital signature of executable programs and display their identities before they are downloaded.

The possible values for the Check for signatures on downloaded programs setting are:

- Enabled
- Disabled
- Not Configured

Vulnerability

Users might download inappropriate or malicious content without realizing it.

4.2.3.18 Do not save encrypted pages to disk

When Internet Explorer retrieves pages from a remote server, it stores the pages in its temporary file cache. This capability improves performance and provides the ability to go forward or backward in the browse history list without a reconnection to the host.

The possible values for the Do not save encrypted pages to disk setting are:

- Enabled
- Disabled
- Not Configured

Vulnerability

Cached pages from SSL-secured connections may contain sensitive information, like passwords or credit card numbers. Pages that are retrieved over SSL connections will not be cached. This configuration may increase the use of network bandwidth as users' browsers re-download pages that would have been cached if this policy setting were not in effect.

4.2.3.19 Empty Temporary Internet Files folder when browser is closed

Pages that are retrieved by Internet Explorer are stored in its temporary file cache. Internet Explorer manages this cache according to the settings in the Temporary Internet Files Settings dialog box. After a file or object is downloaded, it remains in the cache until Internet Explorer removes it.

The possible values for the Empty Temporary Internet Files folder when browser is closed setting are:

- Enabled
- Disabled
- Not Configured

Vulnerability

Sensitive information may remain in the \Temporary Internet Files folder after a user quits Internet Explorer and logs off. Another user on the same computer may be able to gain access to these files.

Internet Explorer uses the \Temporary Internet Files folder as a cache to improve browser performance. If you disable this feature you may increase the amount of time and bandwidth that users need to browse the Web.

4.2.3.20 Consistent MIME Handling

Internet Explorer uses Multipurpose Internet Mail Extensions (MIME) data to determine how to handle files that are downloaded from a Web server. The Consistent MIME Handling setting determines whether Internet Explorer requires that all file-type information that is provided by Web servers be consistent. For example, if the MIME type of a file is text/plain but the MIME data indicates that the file is really an executable file, Internet Explorer changes its extension to reflect this executable status. This capability helps ensure that executable code cannot masquerade as other types of data that may be trusted.

If you enable this policy setting, Internet Explorer examines all received files and enforces consistent MIME data for them. If you disable or do not configure this policy setting, Internet Explorer does not require consistent MIME data for all received files and will use the MIME data that is provided by the file.

Vulnerability

A malicious Web server could deliver executable content by using a non-executable MIME type, and a user who opened the content could be tricked and cause the content to execute.

The existence of the above listed security features in the operating increases its hardening capability. A missing security hardening feature will leave some vulnerability in the system, which can be exploited by the attacker to compromise the system. The vulnerabilities discussed above are analyzed to determine the level of the security violation an attacker can achieve. According to this level the hardening capability score of the operating system is calculated.

4.2.3.21 Domain member: Require strong (Windows 2000 or later) session key

This security feature determines whether a secure channel can be established with a domain controller that cannot encrypt secure channel traffic with a strong, 128-bit, session key. If the administrator enable this policy setting, a secure channel cannot be established with any domain controller that cannot encrypt secure channel data with a strong key. If this policy setting is disabled, 64-bit session keys are allowed.

The possible values for the **Domain member: Require strong (Windows 2000 or later) session key** setting are:

- Enabled
- Disabled
- Not Defined

Vulnerability

Session keys that are used to establish secure channel communications between domain controllers and member computers are much stronger in Windows 2000 than they were in previous Microsoft operating systems.

Whenever possible, advantage of these stronger session keys must be taken to help protect secure channel communications from attacks that attempt to hijack network sessions and eavesdropping. (Eavesdropping is a form of hacking in which network data is read or altered in transit. The data can be modified to hide or change the sender, or be redirected.)

4.2.3.22 Wireless Networking Security

Wi-Fi Protected Access version 2 (WPA2) is the latest version of security for wireless networks. It's more secure than Wired Equivalent Privacy (WEP) and easy to set up using the features built into Microsoft Windows XP Service Pack 2 (SP2).

Microsoft has released an update for Windows XP SP2 that provides operating system support for WPA2. WPA2-Personal is backwards compatible with the original WPA. Routers and access points may have selectable settings for WPA2-Personal only or a mixed mode, which means that you can run both WPA2- and WPA-capable wireless devices on the same network. Most users need to select a mixed setting if they have computers or devices that are not WPA2-Personal compliant. Computers equipped with WPA2-Personal-capable wireless cards will use the stronger WPA2 Advanced

Encryption Standard (AES) encryption, while WPA only-capable devices will fall back to WPA.

Vulnerability

If WEP is used instead of WPA2 the system may vulnerable to Denial of service attack. After the WEP encryption mechanism was released, it was proved to be vulnerable to multiple forms of attack. WEP uses the symmetric cryptography system called RC4 with a user-specified key (64 bits and 128 bits) to protect the data. As a result, WEP alone is not enough to protect your data, and coming sections will address this fact with practical solutions such as dynamic WEP, IPSec, and 801.1x authentication.

The attack methodology of an attacker is as follows:

- Footprint the wireless network Locate and understand your target.
- Passive attack Analyze the network traffic or break the WEP.
- Authentication and authorization Determine what methods are enforced and how they can be circumvented.
- Active attack Launch denial of service (DoS) attacks.

For calculating the hardening capability score the specification of the operating system security feature is to be compared with the specification of the proposed security features, to decide whether the security feature is present in the operating system. The specifications of the security features in the operating system are available in natural language; analyzing these specifications will be quite ambiguous and misleading. It is also possible that specification of the proposed security features matches to a certain level to the specification of the security features in the operating system. If this matching is above a threshold value the feature is considered to be present in the operating system.

The ambiguity in the natural language specification can result in incorrect matching of specification of the security features and eventually in an incorrect hardening capability score. To facilitate high-assurance hardening capability score, the proposed and operating system's security features are formally specified. A formal specification [12] enables to describe security features clearly and precisely. It also could benefit us in automated calculation of hardening capability score on formal specifications.

5.1 Formal Specification of Security Feature.

The formal specification of a security feature is described by schema declarations. A schema contains a unique name for the security feature, a signature, and optionally a property over the signature [18]. A signature is represented by a set of <name, type> pairs, and the property of the signature is represented by a predicate. This predicate is expressed as a logical conjunction of well-formed formulas. The signature consists of at least three variables feature, attribute and value. Feature variable specifies the feature of the operating system which is affected by the security feature, the attribute variable specifies the property of the feature that is considered in the security feature

and value variable specifies the possible values for the attribute variable. As examples, schema of the some security feature is given below.

Minimum Password Length	_____
Feature : Password	
Attribute : Length	
Values : Not defined, Range numeric	

Here the signature consists of three declarations

Feature : Password

This declaration specifies that the password property of the operating system will be affected by the security feature

Attribute : Length

This declaration specifies that the length of the password will be affected by this security feature.

Values : Not defined, Range

This declaration specifies that the values for the length attribute of the password property are not defined or range numeric.

Account lockout duration	_____
Feature : Account	
Attribute : LockDuration	
Values : Not defined, Range numeric	

Feature : Account

This declaration specifies that the user accounts on the operating system will be affected by the security feature

Attribute : LockDuration

This declaration specifies that the duration of locking of user accounts is affected by this security feature.

Values : Not defined, Range

This declaration specifies that the values for the lock duration attribute of the password property are not defined or range numeric.

5.2 Security Hardening Feature Matching

This method of matching security hardening features is derived from the method used in software reuse for matching the formal specifications of software component [13]. The definitions and the algorithms are modified to suite the requirement for matching the security hardening features.

Definition 1 *A declaration d is defined to be an ordered pair $d : (\sigma ; P)$ where σ represents the signature of d , and P refers to the property over σ .*

Since the property P is always associated with the signature σ , we use the notation $P(\sigma)$ to denote the P component of d .

Definition 2 *A signature σ is defined as a partial function $\sigma : n \rightarrow t$, where n represents a set of identifiers and t represents a set of type names.*

While matching declarations, we have to consider two cases: exact match and partial match. This classification applies individually to signatures and properties.

Definition 3 *A declaration $d1$ exactly matches with another declaration $d2$ if and only if the signature $\sigma1$ of $d1$ exactly matches with the signature $\sigma2$ of $d2$ and the property $P(\sigma1)$ exactly matches with the property $P(\sigma2)$.*

Definition 4 *A declaration $d1$ partially matches with a declaration $d2$ if one or both of the following conditions is true:*

- The signature $\sigma1$ of $d1$ partially matches with the signature $\sigma2$ of $d2$.
- The property $P(\sigma1)$ partially matches with the property $P(\sigma2)$.

5.3 Signature Match

We now define exact and partial matches of signatures in detail.

Definition 5 *A signature $\sigma1$ exactly matches with another signature $\sigma2$ if and only if for every name $n1$ in the domain of $\sigma1$, there is exactly one name $n2$ in the domain of $\sigma2$ such that $\sigma1(n1) = \sigma2(n2)$. The ordering of names in either signature is immaterial.*

A necessary condition for an exact match of two signatures is that the number of names in both signatures must be the same (i.e., cardinality of domain of $\sigma1$ = cardinality of domain of $\sigma2$). A consequence of definition 5 is that there may be more than one possibility for an exact match between two signatures. See the two signatures given below for an example:

User	Book
Id : Numeric	Title : String
Name : String	Year : Numeric
Age : Numeric	Pages : Numeric

In this case, id of User can match with either pages or year of Book, giving two possible exact matches.

Definition 6 *A type $\tau1$ is compatible with a type $\tau2$ if the result of performing some operation on a member of the carrier set of $\tau1$ is the same as the result of performing one or more operations on a unique member of the carrier set of $\tau2$.*

Informally, type compatibility between two types indicates that the functional behavior of both types as determined by their carrier sets and operations on carrier sets must be equivalent. A necessary condition for ensuring type compatibility is that the carrier set of τ_1 must be a subset of the carrier set of τ_2 , both sets being derived from the same semantic universe. Thus, a type τ_1 can be ensured to be compatible with a type τ_2 by taking an element x from the carrier set of τ_1 , performing one of the operations associated with τ_1 on x and showing that the result y is the same as performing one or more operations associated with τ_2 on x .

The following properties hold regarding type compatibility:

- Every type is compatible with itself.
- A subtype is compatible with its supertype. Type compatibility is transitive, but not symmetric.

Using type compatibility we can now define a partial match between two signatures.

Definition 7 : *A signature σ_1 partially matches with another signature σ_2 if there exists a bijection f from the domain of σ_1 to the domain of σ_2 such that for every x in the domain of σ_1 , $\sigma_1(x)$ is type-compatible with $\sigma_2(f(x))$.*

A necessary condition for partial match is that the number of components in both the signatures must be the same.

As seen from the definition of partial match, if a signature σ_1 exactly matches with another signature σ_2 , then σ_1 partially matches with σ_2 as well. This is a consequence of the definition for type compatibility, and is also consistent with similar definitions for signature match given in.

5.4 Property Match

Definition 8 : *A property $P(\sigma_1)$ associated with the signature σ_1 exactly matches with a property $P(\sigma_2)$ associated with the signature σ_2 if and only if $P(\sigma_1)$ is equivalent to*

$P(\sigma_2)$ after renaming the names in the domain of σ_1 to match with the names in the domain of σ_2 .

A necessary condition for exact match of the properties $P(\sigma_1)$ and $P(\sigma_2)$ is that σ_1 must exactly match with σ_2 . As an example, consider the two following schemas.

User
Id : Numeric Name : String Age : Numeric
name ≤ 20

User
Title : String Year : Numeric Pages : Numeric
#title ≤ 20

In this case, the signature of User exactly matches with the signature of Book; this results in the following renaming table: (id $\leftarrow$ pages, name $\leftarrow$ title, age $\leftarrow$ year). The property of the signature of User exactly matches with the property of signature of Book after renaming.

Definition 9 : *Given two signatures σ_1 and σ_2 , the property $P(\sigma_1)$ of σ_1 partially matches with the property $P(\sigma_2)$ of σ_2 if $P(\sigma_1)$ implies $P(\sigma_2)$ after renaming the names in the domain of σ_1 to match with the names in the domain of σ_2 .*

As seen from this definition, partial match between two signatures is not symmetric. This is expected; otherwise, partial match will converge to exact match.

5.5 Algorithms for Matching the Security Hardening Features

In this section, we give algorithms to implement the matching method.

5.5.1 Algorithm 1 signature match (LSIG, TSIG)

Purpose: This algorithm implements partial match of signatures LSIG and TSIG. The two signatures must have the same number of components. The type of LSIG(i) must be the same or compatible with the type of TSIG(i), $1 \leq i \leq \#LSIG$.

Input: LSIG, TSIG: Signature

/* LSIG represents library signature and TSIG represents target signature. The type 'Signature' is a sequence of <name, type>. */

Output: table: Name Table

/* Name Table contains pairs of names <l,t>, $l \in \text{LSIG}$ and $t \in \text{TSIG}$. If any name in LSIG is not matched, the table will be empty. */

Local variable: table: Name Table; ag: Boolean

/* The type Name Table is defined as a sequence of ordered pairs of names */

Procedure:

```
table ← {};  
/* Initialize table to an empty sequence */  
flag ← true;  
if (#LSIG = #TSIG) return table  
  while (LSIG ≠ {}) and flag do  
    if (type_compatible (type (head(TSIG)), type (head(LSIG))))  
      table ← table ^ {name(head(LSIG)) → name(head(TSIG))};  
    else  
      table ← {};  
      flag ← true;  
    endif  
    LSIG tail(LSIG);  
    TSIG tail(TSIG);  
  endwhile  
endif  
return table;  
endproc
```

Given two signatures, there might be more than one possible way of matching them. This is because the ordering among the entries (pairs of names and types) in a signature is immaterial. Therefore, we store all possible combinations of matches for a

given pair of signatures in a table, called T. Each entry in T itself is a table that contains one set of matching names.

Algorithm 2 describes the construction of all name tables from a pair of signatures.

5.5.2 Algorithm 2 construct-name-tables (LSIG,TSIG)

Purpose : To construct all possible name tables from the two signatures LSIG and TSIG.

Input : LSIG, TSIG: Signature

/* LSIG represents library signature and TSIG represents target signature. */

Output : result: seq Name Table;

Local variables : permut: _ Signature; current: Signature

Procedure :

```

result  $\leftarrow$  {};
  if (#LSIG = #TSIG)
    permut = permute-signatures (TSIG);
    /* compute all possible permutations of names in TSIG */
    while (permut  $\neq$   $\Phi$ ) do
      current  $\leftarrow$  head(permut);
      if (signature-match (LSIG, current)  $\neq$  {})
        result  $\leftarrow$  resultsignature-match (LSIG, current);
      endif
      permut  $\leftarrow$  tail (permut);
    endwhile
  endif
return result;
endproc

```

Algorithm 3 compares two predicates LP and TP representing properties of library signature and target signature, respectively. The predicates are input in conjunctive normal form, which includes only three operators: $\vee$, $\wedge$ and $\neg$. At first, the two predicates are rewritten into prefix form. If the heads of LP and TP have the same operators, then their respective operands are compared for logical equivalence.

If a match is found, Algorithm 3 is invoked recursively. Otherwise, one of the predicates is commuted and then the algorithm is exercised once again for the operands. On the other hand, if LP and TP are simple predicates without logical operators (the termination condition for recursion), then they are directly compared for logical equivalence.

5.5.3 Algorithm 3 property-match (LP, TP, nametable)

Purpose: This algorithm describes partial match of properties represented by the predicates LP and TP. The properties match as long as LP and TP are logically equivalent.

Input: LP, TP: Predicate; name-table: Name Table;

/ LP represents library predicate and TP represents target predicate, both in conjunctive normal form. 'name table' contains the matched names from LSIG and TSIG which are used by LP and TP. */*

Output: result: boolean Local variable: Lnew ;Tnew; new1; new2: Predicate;

Procedure :

```

declare Op =  $\{\wedge, \vee\}$ ;
result  $\leftarrow$  false;
/* Obtain the prefix expressions for both the predicates */
Lnew  $\leftarrow$  prefix expression (LP);
Tnew  $\leftarrow$  prefix expression (TP);
/* If the first element in each expression is not an operator,
then they are simple predicates, which can directly be
compared for logical equivalence. */
if (head(Lnew)  $\notin$  Op  $\wedge$  head(Tnew)  $\notin$  Op)
    result  $\leftarrow$  (logically-equivalent (Lnew ;Tnew, name-table))
else
/* If both expressions have the same operator at the front,
then check for logical equivalences of their respective
operands by recursively calling this function. If they don't

```

*match, try for logical equivalence once again after commuting the target expression. */*

```

    if (head(Lnew ) = head(Tnew ))
        result ← (property-match (first-operand(LP),
                                first-operand(TP), name-table)
                and
                property-match (second-operand(LP),
                                second-operand(TP), name-table))
        or
        (property-match (first-operand(LP),
                        second-operand(TP), name-table)
            and
            property-match (second-operand(LP),
                            first-operand(TP), name-table))
    endif
endif
return result;
endproc

```

Deciding logical equivalence of two predicates is not a simple task, since the same property can be expressed in more than one way. For example, the two predicates $x > y$ and $y < x$ denote the same property, and hence are logically equivalent. Automating this process requires term rewriting rules. There are a number of tools available for checking logical equivalence of predicates. In our current implementation, we manually check the predicates for logical equivalence. This is done by (i) substituting the names in TP by the matching names in LP using the name-table (that is why the name-table is passed as a parameter to Algorithm 3), and (ii) displaying the modified TP and unmodified LP to the user to check for logical equivalence manually.

Algorithm 4 describes the process of declaration match. This algorithm ensures structural compatibility.

5.5.4 Algorithm 4 declaration-match (LD, TD)

Purpose: This algorithm matches the two declarations LD and TD. They match if their respective signatures and properties match. The algorithm outputs all possible

declaration matches based on the number of possible permutations among the signatures.

Input: LD, TD: Declaration;

/* LD represents library declaration and TD represents target declaration. */

Output: result: seq Name Table

/* A sequence of name tables containing matching names from LD and TD for which both signature and property match. */

Local variables: LSIG, TSIG: Signature; LP, TP: Predicate; tables: seq Name Table;

Procedure :

```
LSIG ← declaration-of (LD);
TSIG ← declaration-of (TD);
LP ← property-of (LD);
TP ← property-of (TD);
tables ← construct-name-tables (LSIG,TSIG);
result ← {};
while (tables ≠ {}) do
    if property-match (LP, TP, table(i))
        result ← result ^ tables(i);
    endif
    tables ← tail (tables);
endwhile
return result;
endproc
```

5.5 Measure of Similarity between Specifications

In the case of partial match the similarity between the security features is calculated, on the basis of which the decision is made that whether the security feature is present in the operating system or not. A simple evaluation method for computing similarity is given. The similarity between a pair of security feature, X and Y, is denoted by $s(X,Y)$. Similarity is symmetric thus for any two security features, X and Y

$$s(X,Y) = s(Y, X)$$

In addition, similarity s is said to be normalized if $0 \leq s(X,Y) \leq 1$. Each predicate formula of the property $P(\sigma)$ is expressed in DNF.

If $s(X,Y)$ represents the similarity between two security features X and Y , then it is assumed that X and Y are of the following forms.

$$X = x_1 \vee x_2 \vee \dots \vee x_m \text{ and}$$

$$Y = y_1 \vee y_2 \vee \dots \vee y_n$$

The disjuncts x_i and y_i are defined in terms of conjuncts, that is,

$$x_i = p_{i_1} \wedge p_{i_2} \dots \wedge p_{i_{u_i}} \quad 1 \leq i \leq m \text{ and}$$

$$y_i = q_{i_1} \wedge q_{i_2} \dots \wedge q_{i_{v_i}} \quad 1 \leq i \leq n$$

where u_i and v_i are the number of conjuncts within disjunct x_i and y_i , respectively. The disjuncts of each object are ordered from left to right in a non decreasing order according to the number of conjuncts in each disjunct. Given that u_i and v_i represent the number of conjuncts for disjuncts x_i and y_i , respectively, the following inequalities are true: $u_{i-1} \leq u_i$ and $v_{i-1} \leq v_i$, for all i . Moreover, each conjunct p_{i_k} is associated with an equivalence class eq class. For example, if $p_{i_k} = \text{greater}(a, b)$ and greater is in the equivalence class for comparison, then $eq \text{ class}(p_{i_k}) = \text{comparison}$.

The equivalence classes may be specified by the users or by the system, defined. The number of equivalence classes in the security feature list is assumed to be a known value, say T . Using the above definitions, a matrix $X_{m \times (T+1)}$ is constructed for every security feature X . The matrix $X_{m \times (T+1)}$ derived from security feature X has m rows and $T + 1$ columns. $X(i, j)$ represents the entry in row i and column j . Row i represents the i^{th} disjunct of X as follows, where there are u_i conjuncts in disjunct x_i

$$X(i,0) = u_i \quad 0 \leq i \leq m-1 \text{ and}$$

$$X(i, j) = 1, \quad x_i \text{ has } l \text{ terms in } eq \text{ class } j$$

Similarly, for security feature Y containing v_i conjuncts in disjunct y_i the corresponding matrix is defined by

$$\begin{aligned} Y(i,0) &= v_i & 0 \leq i \leq n-1 \text{ and} \\ Y(i,j) &= 1, & y_i \text{ has } l \text{ terms in eq class } j \end{aligned}$$

From the derived matrices $X_{m \times (T+1)}$ and $Y_{n \times (T+1)}$ the similarity matrix $s'_{m \times n}$ is constructed. The following expression defines $s'_{m \times n}$

$$\begin{aligned} &\text{for all } i, j \text{ if } X(i, 0) = Y(j, 0) \\ &\text{then } s'(i, j) = \frac{\sum_{t=1}^T 2 * \min(X(i, t), Y(j, t))}{\sum_{t=1}^T (X(i, t) + Y(j, t))} \\ &\text{else } s'(i, j) = 0 \end{aligned} \tag{1}$$

where $s'(i, j)$ is the similarity of the i^{th} disjunct of X and the j^{th} disjunct of Y. The similarity between two conjunctive expressions from two security feature is calculated according to the minimum number of common occurrences of a given equivalence class. Since the results from the clustering process are purely based on syntactic similarities, only the disjuncts with the same number of conjuncts are selected for comparison. Assume N is the number of nonzero entries in $s'_{m \times n}$. The similarity between software security feature X and Y is calculated as follows

$$s(X, Y) = \frac{\sum_{i=1}^m \sum_{j=1}^n s'(i, j)}{N} \tag{2}$$

Here $s(X, Y)$ is a normalized similarity since $0 \leq s(X, Y) \leq 1$.

Example: Suppose the similarity of two components X and Y is to be computed where, both specifications are in DNF. Let $X = (C1 \wedge C2) \vee (C2 \wedge C3 \wedge C3) \vee (C3 \wedge C4 \wedge C5)$ and $Y = (C3) \wedge (C2 \wedge C3) \wedge (C3 \wedge C3 \wedge C5) \wedge (C2 \wedge C5 \wedge C5)$, where C_i refers to the term that corresponds to the i^{th} equivalence class. There are 5 equivalence classes in this case, so $T = 5$. X has 3 disjuncts and Y has 4 disjuncts. The

corresponding matrices for X and Y are shown in Figures and where the vertical axis represents the disjuncts in each security feature and the horizontal axis refers to the equivalence classes. From Formula (1) the similarity matrix $s'(X,Y)$ can be computed yielding results shown in Figure c where the vertical axis represents the disjuncts in component X and the horizontal axis refers to the disjuncts in the Y component. From Formula (2) the similarity $s(X, Y) = \frac{2/4+4/6+2/6+4/6+2/6}{5} = \frac{1}{2}$ is obtained.

This value is used to determine whether the security feature is to be considered present in the operating system or not.

	1	1	2	3	4	5
1	2	1	1	0	0	0
2	3	0	1	2	0	0
3	3	0	0	1	1	1

Figure 5.1a

	0	1	2	3	4	5
1	1	0	0	1	0	0
2	2	0	1	1	0	0
3	3	0	0	2	0	1
4	3	0	1	0	0	2

Figure 5.1b

	1	2	3	4
1	0	2/4	0	0
2	0	0	4/6	2/6
3	0	0	4/6	3/6

Figure 5.1c

If the value of $s(X,Y)$ is equal to or greater than 0.5 the security hardening feature is considered to be present in the operating system.

Implementation of the Proposed Framework

Each proposed security hardening feature prevents some sort of security violation. A missing security feature in the operating system provides the attacker an opportunity to violate the security of the information system. The absence of security features decreases the hardening capability of the operating system. The security violations due to a missing security feature can be classified into four types:

1. Confidentiality violation
2. Integrity violation,
3. Availability violation,
4. System compromised.

Confidentiality Violation

Confidentiality addresses the secrecy and privacy of information [9]. Confidentiality violation allows an attacker to directly steal information from the system.

Integrity Violation

Integrity provides for the correctness of information. It allows users of information to have confidence in its correctness [9]. Integrity violation allows an attack that can directly change the information residing on or passing through a system.

Availability Violation

Availability simply means that when a legitimate user needs the information, it should be available [9]. Availability violation results in an attack that directly inhibits a user (human or machine) from accessing a particular system resource. Denial of service attacks are availability violations by this definition.

System compromised

System compromised gives the attacker the privilege to access the system on four different levels: run an arbitrary code, elevate privilege, account break-in, and root break-in.

6.1 Severity Score of Security Hardening Features

The vulnerabilities due to a missing security feature are examined to determine that which security violations can occur due to that missing security feature. Each security violation result in some damage to the information system. The severity (extensiveness) of the damage when a system violation occurs is given a severity level of high, medium and low. The conditions on damage types and severity levels are described in Table 5.1.

Table 6.1. Description of damages by Types and Severity

Damage Type	Severity Level	Condition
Confidentiality	High	Disclosure of information and system configuration in root/super user level
	Medium	Disclosure of system information and configuration in user level
	Low	Disclosure of some non-relevant information.
Integrity	High	Information and system configuration changed in root/super user level
	Medium	Information changed in user level
	Low	Non-relevant information changed in other user level
Availability	High	Whole system crash or unavailable
	Medium	Some services unavailable –System temporary unavailable
	Low	Some services temporary slow down with flooding
System Compromised	High	Root break-in –Account break-in –Run arbitrary code by root/super user privilege

	Medium	Privilege gain in some domain – Run arbitrary code by user privilege
	Low	Run arbitrary code by other user privilege

Each proposed security feature is given a severity score by analyzing the severity of the violations that can occur because of nonexistence of the security feature in the operating system. The vulnerabilities that will occur in the system if a security feature is missing or if it is not securely configured are discussed in the fourth chapter “proposed security features”. For calculating the severity score the severity levels are given numerical value: 3 for high, 2 for medium, and 1 for low severity. The severity score is calculated by first deciding the severity level for each violation due to the missing security feature and then adding the numerical values of the severity levels. If the security feature does not lead to a violation then the numerical value of 0 is given for that violation.

The following are examples of calculating the severity score

Example 1 : Minimum Password Length

If this hardening option is missing the password can be of any length and small password can be easily cracked. The attacker can compromise the system and can get root/super user access. Properly setting this option can make password cracking quite difficult and prevents the attacker from gaining system access. The severity levels of violations if this security feature is mission are

Violation	Severity Level
Confidentiality	N/A
Integrity	N/A
Availability	N/A
System Compromised	High

If the attacker is able to crack the password of a root/super user account then the complete system or network is on his full exposure. The other three violations

confidentiality, integrity and availability become irrelevant in this condition, that's why there severity level is not applicable.

The severity score for this security feature will be $0 + 0 + 0 + 3 = 3$

Example 2 : Deny Force shutdown from a remote system

Any user who can shut down a computer could cause a Denial of Service condition to occur. The system can be made unavailable for a small duration if this setting is not properly set or missing from the operating system. The severity levels of violations for this security feature are.

Violation	Severity Level
Confidentiality	N/A
Integrity	N/A
Availability	Medium
System Compromised	N/A

Thus the severity score for this security option will be $0 + 0 + 2 + 0 = 2$.

The severity score for all of the hardening options explained in the fourth chapter are as follows

Table 6.2 Security Feature Severity Score

	Authentication Hardening security features	C	I	A	SC	Severity Score
1	Maximum Password Age	0	0	0	3	3
2	Minimum Password Age	0	0	0	2	2
3	Minimum Password Length	0	0	0	3	3
4	Enforce Password History/Uniqueness	0	0	0	2	2
5	Password must meet complexity requirements	0	0	0	3	3
6	Store password using reversible encryption for all users	0	0	0	3	3

7	Allow changing passwords only when required	0	0	0	2	2
8	Account lockout duration	0	0	2	0	2
9	Account lockout threshold	0	0	3	0	3
10	Reset account lockout counter after	0	0	1	0	1
11	Use of bank passwords to console logon	0	0	0	3	3
12	Do not allow Machine account password changes	0	0	0	3	3
13	Display of Last User Name	0	0	0	2	2
14	Number of Previous Logons to Cache	0	0	0	3	3
15	Require Domain Controller Authentication to Unlock Workstation	0	0	0	1	1
16	Do not allow Storage of Credentials or .NET Passports for Network Authentication	2	0	0	2	4
17	Pluggable different authentication for different services	0	0	0	3	3

	System Right and Access Controls Hardening security features	C	I	A	SC	Severity Score
1	Load and unload device drivers	3	3	2	3	11
2	Manage auditing and security log	3	0	0	0	3
3	Log on as a service	2	2	0	2	6
4	Force shutdown from a remote location	0	0	3	0	3
5	Act as part of the operating system	3	3	0	3	9
6	Adjust memory allocated to a process	0	0	3	0	3

7	Back up files and directories	2	2	0	0	4
8	Change the system time	0	2	2	0	4
9	Debug programs	2	2	0	2	6
10	Access to this computer from the network	2	2	0	2	6
11	Lock pages in memory	0	0	3	0	3
12	Modify system environment variables	0	2	2	0	4
13	Increase scheduling priority	0	0	3	0	3
14	Perform disk management tasks	0	2	2	0	4
15	Profile single process	2	0	0	0	2
16	Profile system performance	1	0	0	0	1
17	Restore files and directories	3	3	2	2	10
18	Take ownership of files or other objects	3	3	3	3	12
19	Ability to format and eject removable media	3	0	0	3	9
20	Restriction policy for software	3	3	3	3	12
21	Create Token Objects	3	2	3	3	11

	Network Access and Communication Hardening Security Features	C	I	A	SC	Severity Score
1	Digitally sign communications	2	2	2	2	8
2	Send unencrypted password to third-party servers	3	0	0	0	3
3	Do not allow Anonymous enumeration of SAM accounts	3	0	0	0	3
4	Digitally encrypt or sign secure channel data	3	0	0	0	3
5	Require strong session key	3	0	0	0	3
6	Strong key protection for user keys stored on the computer	3	0	0	0	3

7	Use FIPS compliant algorithms for encryption, hashing, and signing	3	0	0	0	3
8	Shares that can be accessed anonymously	3	2	0	0	5
9	Force logoff when logon hours expire	2	2	2	2	8
10	Disable remote desktop sharing	2	2	0	2	6
11	Proxy settings per-machine instead of per user	2	0	1	2	5
12	Do not allow users to add/delete sites	2	0	1	2	5
13	Do not allow users to change security zone policies	2	0	2	2	6
14	Allow software to run or install even if the signature is invalid	0	0	1	2	3
15	Allow active content from CDs to run on user machines	0	0	0	2	2
16	Third-party browser extensions	0	0	0	2	2
17	Check for server certificate revocation	2	0	0	2	4
18	Check for signatures on downloaded programs	0	0	0	2	2
19	Do not save encrypted pages to disk	2	0	0	0	2
20	Empty Temporary Internet Files folder when browser is closed	2	0	0	0	2
21	Consistent MIME Handling	2	0	0	2	4
22	Wireless Networking security	3	2	3	3	11

6.2 Hardening Capability Scoring Technique

After deciding the severity score for the security features it is now straightforward to compute the hardening capability score for an operating system. The scoring is done by the security hardening feature categories: authentication, network access and communication, system right and access controls, services, data access and file

system, patch management, system auditing. Calculating hardening capability score for each of the security feature categories provides the administrator a summarized view of the level of hardening capability present in the operating system for each of the category. This helps the administrator in selecting the operating system which provides the desired hardening capability in the required security category. For example if the security requirements of the information system needs a high level of security in network access and communication then an operating system having a high score in this category will be a preferred choice. Also an overall system scoring was attempted to provide a useful index to the overall hardening capability of the operating systems under consideration.

6.2.1 Scoring by Hardening Option Categories

For each type of security hardening feature category, the scoring is done by considering each of the suggested security features in that category. The severity score of each hardening option is calculated and if that security feature is found in operating system then its score is added to the score of its category and if the security feature is missing from the operating system then its score is subtracted from the score of its category.

Now for calculating the hardening capability score the operating system, for each of the security feature, is assigned the positive severity score of the security feature if it is present in it and if the security feature is missing then its score is negated and then assigned to the operating system. For example the minimum password length security hardening feature prevents the attacker from gaining access into the system. This security feature has a severity score of 3, so the operating system is assigned a score of 3 for this severity feature and the score of the authentication category is added by 3. In case if this security hardening feature is missing from the operating system then the severity score of the security hardening feature is negated and assigned to the operating system, that is, the operating system is assigned a score of -3 for this security feature and the score of authentication category is subtracted by 3.

6.2.2 Overall System Scoring

In order to provide a concise view of the hardening capability, we attempted an overall hardening capability scoring scheme as follows. Each of the hardening option categories is given a weight on the scale of 1 to 5, 1 for high and 5 for low. This weighting is based on the extent to which the options in the category are important to the security of the system. Lower weight is assigned to a category if the security hardening features in it are very important for the system's security, that is, lower the weight assigned larger the affect of the category on system's security. For example the authentication category is given a weight of 2 because of its importance as a first line of defense, system rights and access control is given a weight of 1 as System rights and Access control acts as the physical controls that prevent unauthorized access to systems.

The overall system hardening capability score is calculated by dividing the hardening capability score of the categories by their respective weight and adding the result. The severity level of each subcategory is given in Table 5.3. For example if the score of the authentication category is 50, then this score is divided by the weight of the authentication category and result is added to the overall hardening capability score of the operating system.

Table 6.3 Security Feature Categories Weights

Hardening Option Categories	Weight
Authentication	2
Network Access and Communication	1
System Right and Access Controls	1
Services	4
Data Access and File System	3
Patch Management	2
System Auditing	3

6.3 Hardening Capability Scoring

Armed with the severity-based hardening capability scoring scheme, the hardening capability score of the windows 2000 professional is calculated against windows XP SP2 operating system. Windows XP SP2 is having more strong security hardening options. The results of the scoring are shown in Table 5.4 to 5.6.

Table 6.4 Hardening Capability Score by Security Feature Categories

	Authentication Hardening security features	Severity Score	Windows XP SP2	Windows 2000
1	Maximum Password Age	3	3	3
2	Minimum Password Age	2	2	2
3	Minimum Password Length	3	3	3
4	Enforce Password History/Uniqueness	2	2	2
5	Password must meet complexity requirements	3	3	3
6	Store password using reversible encryption for all users	3	3	3
7	Allow changing passwords only when required	2	2	2
8	Account lockout duration	2	2	2
9	Account lockout threshold	3	3	3
10	Reset account lockout counter after	1	1	1
11	Use of bank passwords to console logon	3	3	3
12	Do not allow Machine account password changes	3	3	3
13	Display of Last User Name	2	2	2
14	Number of Previous Logons to Cache	3	3	-3
15	Require Domain Controller Authentication to Unlock Workstation	1	1	1
16	Do not allow Storage of Credentials or .NET Passports for Network Authentication	4	4	-4
17	Pluggable different authentication for	3	-3	-3

	different services			
	Total Score →		37	23

	System Right and Access Controls Hardening security features	Severity Score	Windows XP SP2	Windows 2000
1	Load and unload device drivers	11	11	11
2	Manage auditing and security log	3	3	3
3	Log on as a service	6	6	6
4	Force shutdown from a remote location	3	3	3
5	Act as part of the operating system	9	9	9
6	Adjust memory allocated to a process	3	3	3
7	Back up files and directories	4	4	4
8	Change the system time	4	4	4
9	Debug programs	6	6	6
10	Access to this computer from the network	6	6	6
11	Lock pages in memory	3	3	-3
12	Modify system environment variables	4	4	4
13	Increase scheduling priority	3	3	3
14	Perform disk management tasks	4	4	4
15	Profile single process	2	2	2
16	Profile system performance	1	1	1
17	Restore files and directories	10	12	12
18	Take ownership of files or other objects	12	12	12
19	Ability to format and eject removable media	9	6	6
20	Restriction policy for software	12	12	-12
21	Create token objects	11	11	11
	Total Score →		125	95

	Network Access and Communication Hardening Security Features	Severity Score	Windows XP SP2	Windows 2000
1	Digitally sign communications	8	8	8
2	Send unencrypted password to third-party servers	3	3	3
3	Anonymous enumeration of user accounts	3	3	3
4	Digitally encrypt or sign secure channel data	3	3	3
5	Require strong session key	3	3	3
6	Strong key protection for user keys stored on the computer	3	3	3
7	Use FIPS compliant algorithms for encryption, hashing, and signing	3	3	3
8	Shares that can be accessed anonymously	5	4	4
9	Force logoff when logon hours expire	8	8	8
10	Disable remote desktop sharing	6	6	6
11	Proxy settings per-machine instead of per user	5	5	5
12	Do not allow users to add/delete sites	5	5	-5
13	Do not allow users to change security zone policies	6	6	6
14	Allow software to run or install even if the signature is invalid	3	3	-3
15	Allow active content from CDs to run on user machines	2	2	-2
16	Third-party browser extensions	2	-2	-2
17	Check for server certificate revocation	4	-2	-2
18	Check for signatures on downloaded programs	2	-2	-2
19	Do not save encrypted pages to disk	2	-2	-2
20	Empty Temporary Internet Files folder when browser is closed	2	-2	-2
21	Consistent MIME Handling	4	4	-4
22	Wireless Networking security	11	12	-12

	Total Score →		71	19
--	---------------	--	----	----

Table 6.5 Final Hardening Capability Score by Security Feature Categories

Security Feature Categories	Hardening Capability Score	
	Windows 2000 Pro	Windows XP SP2
Authentication	23	37
Network Access and Communication	95	125
System Right and Access Controls	19	71

Table 6.6 Overall Hardening Capability Comparison

	Windows 2000 Pro	Windows XP SP2
Overall Hardening Capability Score	125.5	214.5

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Network security is the effort to create a secure computing platform, designed so that agents (users or programs) cannot perform actions that they are not allowed to perform, but can perform the actions that they are allowed to. The actions in question can be reduced to operations of access, modification and deletion.

Security is an ever changing topic with new vulnerabilities and more sophisticated attacks coming out on a daily basis. Security Vulnerabilities are a matter of one's unawareness and can be avoided or properly managed as long as we are equipped with good information and a set of best of breed tools.

To minimize the number of possible security exposures, the system administrator must proactively configure the operating system for increased security and decreasing its susceptibility to compromise. Moreover, efficient security tools cannot work correctly or effectively when the operating system they operate under is vulnerable. The security of a computing system, to a large extent, depends on the robustness of its operating system. Evaluating the hardening capability present in an operating system helps, to a great extent, in making the complete system more secure by making the selection of the operating system more precise according to the security needs of the system.

This work attempts to quantify the hardening capability present in an operating system. A number of security settings are listed, which can make an operating system harder to security violations. Windows, due to its large deployment, is used as a case study. Our severity-based hardening capability scoring scheme is derived from the listed security hardening options for an operating system. The windows 2000 professional and windows XP SP2 operating systems are used to build a hardening

capability profile to exemplify the scheme usefulness.

The use of formal specification for matching the security features can increase the efficiency of the scoring process. But the main problem with using formal specification is in writing the formal specification for the proposed security features and of the operating systems, which requires a sound knowledge and expertise of formal specification language. This limitation can be relaxed to a limit by using a formal specification translator. This translator will translate the specification written in natural language to formal language specification. But a specification written in natural language can lead to multiple formal specifications because of the inherent ambiguity in natural language. Therefore the input language for the translator cannot be purely natural, that is, it must have a specified grammar.

From the scoring results, it can be readily seen, quantitatively, that enhancing the security options in the operating system increases its hardening capability and in turn lower the level of vulnerability of the system drastically.

7.2 Future Work

In this work, the scoring by severity level of 3 for high, 2 for medium, and 1 for low of the extensiveness of violation is quite arbitrary. Although this gives us a significant quantitative view of hardening capability, a more detailed weighted scoring should be more desirable. List of the Security features is purposed for three categories of security hardening features : authentication, system rights and access control and network access and communication. An extensive list of security features is for all the security feature categories is required to provide a detailed scoring of the hardening capability of the operating system. The list should be prepared by analyzing the security features present and missing in the operating system commercially available operating systems.

The categories of proposed security features for the operating can be further divided in to subcategories. This subcategorizing will further refine the overall system

hardening capability scoring by assigning weights to each subcategory instead of main categories.

Also, our hardening capability scoring scheme still needs rigorous mathematical analysis in order to guarantee a trustworthy operating system for system administrators' peace of mind. This will require a formal model for the hardening configuration and the scoring scheme. The operating system specifications are written in natural language which cannot be analyzed by any tool. For an automated tool to calculate the hardening capability score a formal specification of the operating system security features are required. This formal specification of the security features will in turn allow methodical proof and verification using automated tools later on. This scoring technique can be expanded to include other types of operating systems and servers.

References

- [1] Bruce Schneier , “Secrets and Lies”, John Wiley & Sons, 2000
- [2] Seth Friedman, “Building the Ideal Web Hosting Facility: A Physical Security Perspective”, SANS institute 2001.
- [3] Center for Internet Security ;
www.cisecurity.org
- [4] Windows XP Professional Operating System Legacy, Enterprise, and Specialized Security Benchmark Consensus Baseline Security Settings, Version 2.01, August, 2005, The Center for Internet Security
- [5] Next Generation Scoring Tool, Users Manual, Public Version 1.0, The Center for Internet Security
- [6] The Bastille Hardening Program ;
www.bastille-linux.org
- [7] Windows XP Security Guide, April 13, 2006, available at:
<http://go.microsoft.com/fwlink/?linkid=14840>
- [8] Windows2000 Security Hardening Guide, May 15, 2003, available at:
<http://www.microsoft.com/downloads/details.aspx?FamilyID=15E83186-A2C8-4C8F-A9D0-A0201F639A56&DisplayLang=en>
- [9] Michael Gregg, “Certified Ethical Hacker Exam Prep”, Que, April 07, 2006
- [10] The Honeynet Project
<http://project.honeynet.org>
- [11] The Labmice.net, “Windows XP hardening Checklist”
<http://labmice.techtarget.com>
- [12] B. Potter, J. Sinclair and D. Till. “An Introduction to Formal Specification and Z” Prentice Hall International Series in Computer Science, 1991.

- [13] Jun-Jang Jeng and Betty H.C. Cheng, "Using Formal Methods to Construct a Software Component Library", Proc. of 4th Eur. Soft. Eng. Conf., Lect. Notes in Comp. Science, Vol. 717, pp. 397-417, Sept. 1993.
- [14] Mario Eberlein, Rene Auberger, Wolfram Andreas Richter, "Securing Linux, Part 3: Hardening the System", IBM DeveloperWorks.
<http://www-128.ibm.com/developerworks/linux/library/l-seclnx3/>
- [15] John H. Terpstra, Paul Love, Ronald P. Reck and Tim Scanlon, "Hardening Linux", McGraw-Hill/Osborne © 2004
- [16] The Linux PAM resource page at kernel.org:
<http://www.kernel.org/pub/linux/libs/pam/modules.html>
- [17] Roberta Bragg, "Hardening Windows Systems", McGraw-Hill/Osborne © 2004
- [18] J.M. Spivey, "The Z Notation: A Reference Manual", Second Edition, Prentice Hall International Series in Computer Science, 1992
- [19] PC Magazine® Windows® XP Security Solutions, Wiley Publishing, Inc.
- [20] Jonathan Hassell, "Hardening Windows", Second Edition, Apress
- [21] James Turnbull, "Hardening Linux", Second Edition, Apress

List of Papers Published

- [1] Yugma Kumar Agrawal, Maninder Singh, “Hardening Capability Scoring Scheme for Operating System”, in National Conference on Information Technology: Emerging Engineering Perspectives & Practices (April 6-7, 2007), ITEEPP’07