Developing and Implementing

an Operating Systems Security Course with Labs

Harry Bulbrook

bulbrookh@durhamtech.edu

Durham Technical Community College

1637 East Lawson Street

Durham, NC 27703

Developing and Implementing

an Operating Systems Security Course with Labs

Abstract

A core component of any curriculum in modern information security is the security of the operating systems that reside on the workstations and servers of a network. Effective information security depends on addressing all facets of how information is stored, moved, and modified. Since the operating system of a computer is the primary means of implementing the security of the information on that computer, it must be configured to minimize the risks of losing or compromising the data being processed.

Durham Technical Community College, as part of its new Information Systems Security curriculum, is developing a security course based on securing operating systems. This course will instruct students in the fundamentals of designing security architectures and provide an overview of security administration of several operating systems, focusing primarily on Windows and Linux. Additionally, students will also learn the design of basic security defenses and the use of network analysis tools. Topics covered will be essential foundation for later courses which will cover intrusion detection, Defense-in-Depth, attack methodologies, and firewall security and configuration.

This paper will present a list of course objectives and an outline for the course. In addition, several lab exercises will be developed and presented, including auditing and monitoring (through log files), and locking down access (including implementation of password policies.)

Introduction

Durham Technical Community College serves a two-county area (Durham and Orange) of North Carolina that includes a number of large research Universities and high-tech companies. The Research Triangle Park (RTP) is located in Durham County and was designed to nurture public/private partnerships between local Universities and Industry. As such, there is a fairly significant community need for employees trained in high-tech support areas, such as Programming, Electronics Engineering, and Information Systems / Networking. Durham Tech has offered these programs for several years, but feedback from members of our Networking Programs Advisory Committee revealed a need for Information Systems Security to be added to the curriculum. Two major thrusts were developed: Adding Information Assurance topics to all of the courses in the existing program, and creating an entirely new degree with a specialization in Information Systems Security. The State Board of Community Colleges was at the time reviewing all of its IT courses in an Information Technology Curriculum Improvement Project (CIP), and one of its outcomes was to add security information to individual IT courses as well. The Board also produces a list of courses that make up an Information Systems Security program (called a Curriculum Standard), with brief course descriptions for each course in the program. As a local college, we are using that Program as a base for our locally-created Plan of Study, which will add courses that may be helpful to our specific service area.

One of the first courses in Durham Tech's Information Systems Security Plan of Study is called SEC 160 - Secure Admin I (Security Administration). This is a standard 3 credit hour class, with 2 hours of lecture and 2 hours of laboratory work per week for a semester-long course. It has a prerequisite of our basic security course (SEC 110 – Security Concepts) and our

basic Networking course (NET 125 – Networking Basics), but is designed to be a gateway course into later security topics like Intrusion Detection, Defense-In-Depth, and Attack Methodology. The course description is:

> "This course provides an overview of security administration and fundamentals of designing security architectures. Topics include networking technologies, TCP/IP concepts, protocols, network traffic analysis, monitoring, and security best practices. Upon completion, students should be able to identify normal network traffic using network analysis tools and design basic security defenses." (ncccs.cc.nc.us)

Because our existing Networking Technology program covers many issues in network design, the decision was made to focus on configuring a security architecture from the standpoint of the operating system. The course is designed to cover the necessary components of a networked operating system, and how that architecture may be designed and implemented with secure access and usage in mind. The course will also include system administration tasks, especially noting which actions (or lack of action) by the sysadmin may compromise or enhance overall security. Security defenses for the operating systems discussed in the course will also be presented, including host-based firewalls and encrypted network traffic. Finally, students will work with network analysis tools to identify normal and abnormal network traffic, identify their source and destination, and react appropriately.

Course Objectives

In designing the objectives for the course, several factors are considered. The first factor is the Course Description. Since this course description comes from the State Board of Community Colleges, the topics, ideas, and requirements listed in the description must be

accounted for to presume that the course will fulfill the requirements posted by the description. The second factor is prerequisites for the course. Since students will be expected to have covered basic networking and basic security, the course will be able to move along assuming prior knowledge of basic networking and security concepts like encryption, broadcasts, passwords, and policies. The third factor to consider is the courses following in the Program. Intrusion Detection and Defense-In-Depth are two courses which use this course as a prerequisite, and students leaving this course should be prepared to work with the topics covered in those courses. One final factor that is pertinent to our college is the college's push for tracking Learning Outcomes instead of Course Objectives. A Learning Outcome seeks to demonstrate a change in knowledge or skill, and includes the means to provide the outcome and the methods to assess the outcome. All of these factors were considered in creating the following objectives.

Objective 1: Participate effectively in class. This objective ensures that a student gets as much as possible from the class. Independent work, Group assignments, active question and answer participation, and performing assigned reading are all part of this objective.

Objective 2: Demonstrate methods for keeping networks and their computers secure. This objective is central to showing understanding of the facilities for securing networks, including protection against viruses, worms, malware, and OS and application exploits. Network topology, host- and network-based firewalls, patches, authentication and encryption, and policy enforcement are all methods that can use used to demonstrate comprehension. Benefits of virtual machines will also be integrated.

Objective 3: Implement secure access through authentication and encryption. This objective shows special interest in remote access to resources, and the difference between public

network and private ones. Cryptographic protocols will be discussed, as well as PKI certificate usage, password / token policy enforcement, VPN and VLANs, and IPsec.

Objective 4: Implement security policies. Policies will have been developed in a previous course, but they can sometimes be theoretical. This objective is designed with a focus on implementation – what policies can be enforced with respect to access methods, available resources, time and date restrictions, data management, and data classification or level of access.

Objective 5: Provide common application security. Specific issues with common applications (especially network-based applications) are dealt with in this objective. DNS, SMTP, HTTP, Instant Messaging, and other user- and system-accessible protocols with their implementation or protocol design weaknesses will be addressed and evaluated.

Objective 6: Ensure continuing security. This objective makes the assumption that any created network system, even if well designed, can never be assumed to be secure. Detection of active attacks through network traffic analysis, system and application log monitoring, and auditing procedures are the approaches used for this objective. Also implied in this objective is that waiting for something to break is a bad idea – tracking the warning signs of a problem or attack is just as important.

Objective 7: Ensure continuity. Disaster planning and recovery issues will be dealt with via this objective. Redundancy (for systems and data), fault tolerance, power and environmental conditioning, backups, recovery strategies, and clustering will be available options for demonstrating facility with this topic.

Objective 8: Troubleshoot security vulnerabilities. This objective is a comprehensive look at synthesizing information and methods from the entire course. Specifically, this course

lays the groundwork for penetration testing, by having students look for and identify weaknesses in configurations for operating systems, applications, firewalls, network devices, policies, and procedures in a given environment.

Course Outline

Course Outlines should serve multiple purposes. Firstly, they should provide a roadmap for the development of a syllabus by the Instructor, listing all the topics to be covered in the course. In addition, the outline also defines the specific issues that will be explored in each topic, allowing for prospective students and potential employers to review the content of a course to verify that it satisfies their needs. The outline is designed for flexibility in the actual presentation and assessment of the topics in the course, and is not a fixed and complete list. The outline presents the minimum of instructional requirement for the course, but also allows for additional topics to be introduced by the instructor as they deem appropriate. The order, depth, and breadth of each topic are also defined by the syllabus that the instructor develops for the specific instance of the course. For this Secure Admin course, the outline is as follows:

I. Securing Network and Computers
   - A. Types of Attacks – Malware, DoS, Spoofing, OS and Application exploits
   - B. Securing the Network
   - C. Securing the Operating System

II. Authentication and Encryption
   - A. Encryption Methods
   - B. Authentication Methods
   - C. VLANs and VPNs
   - D. Certificates and PKI

III. Implementing Security Policies
   - A. User and Group Account access policies
   - B. Securing Passwords

C. File System Rights
D. Network Access Control

IV. Network application security
A. SSL – Email, Web
B. Authenticated connections for file transfer
C. DNS, DHCP, and LDAP

V. Monitoring, Auditing, and Network Analysis
A. Intrusion Detection
B. Audit Trails and Log Files
C. Network Traffic Capture and Analysis

VI. Disaster Planning and Recovery
A. UPS and AC – Power and environmental conditioning
B. RAID
C. Hardware Redundancy and Clusters
D. Data Availability and Backup

VII. Vulnerability Assessment
A. Policy and the Human Component
B. Firewall Rule configuration
C. Patch Level detection
D. Port Scanning
E. Privileged Accounts

Specific Lab Exercises and Configurations

In creating lab exercises for this course, the main goal is to support instruction by providing a hands-on activity for student to complete in the specific topic or area of interest. Presented below are activities that demonstrate concepts in two areas of focus for the course: Auditing and Log Monitoring, and Policy Implementation and Access Restriction for User Account Password and Login Restrictions.

As a design philosophy, these labs are being created with Virtual Machines in mind. The benefits of Virtual Machines are numerous. The resources of lab computer systems can be

utilized more effectively, multiple environments can be configured quickly and easily, and access to external resources can be provided without permitting attacks to those resources. A special benefit for this class may be the ability to package the labs as a downloadable unit, allowing for the presentation of the class in a distance environment.

As such, the following labs are designed to be used in an environment configured as follows: VMWare Server 1.0 installed on designated host machine, sufficient RAM installed into host machine, sufficient disk space for all VM images, full configuration of VMnet capability on host machine, and Internet access.

Logging, Auditing and Log Monitoring

Logging is the tracking of machine activity. Log files are stored record of tracked activity, and can be a simple as a plain text file or as complex as a database. On UNIX and Linux Operating Systems, the logging facility is provided by the syslog service, often storing data in the local file /var/log/messages. On Windows Operating Systems, logging is provided by the Event Log service, by default storing data in local System, Security, and Application logs. Linux logging information is available through standard text file viewing and manipulation programs, but the Windows log results are only viewable through the Event Viewer. In addition, Windows logs are difficult to automatically export to a processable format, often necessitating the purchase of a specialized application for managing Windows logging information. This exercise will demonstrate logging events on both Linux and Windows systems. In addition, those events will be tracked as they occur. Finally, we will use the syslog facility to combine events from both the Windows and the Linux machines into a single, easy to access and process logging facility.

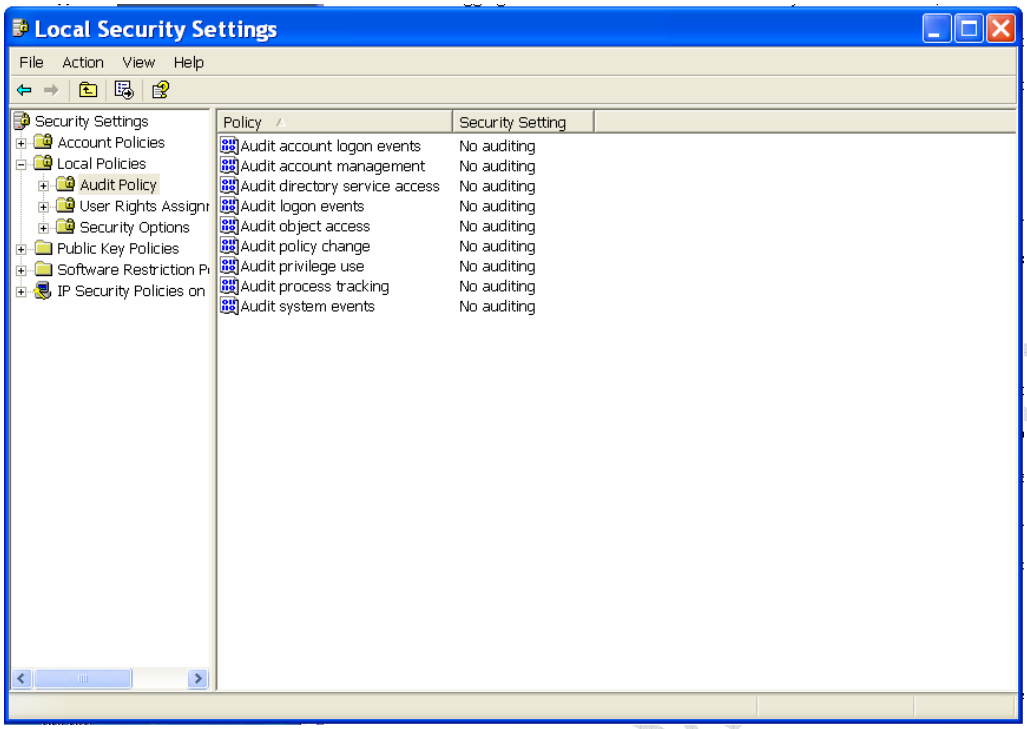
Setup

Start with a Windows XP workstation and a Linux workstation that can communicate over the network with each other. If using Virtual Machines, the simplest way to do this is to ensure that the two machines are on the same virtual network and share the same network address. For this example, assume that the Linux workstation is 192.168.221.125 and the Windows workstation is 192.168.221.129. You will need to verify that they can communicate via UDP port 514, so any firewalls or filtering of that traffic on the two machines or the network between them should be disabled. (Verification of that traffic could be an additional exercise. One method is to use the `netcat` utility on both machines, The Linux computer with the command `nc -l -u -p 514`, then the Windows computer with the command `nc -u 192.168.221.125 514`. Any text entered at the resulting prompt followed by a return will echo to the other computer if traffic is being passed correctly.)

Lab Scenario: Audit Logging in Windows

Login to Windows with Administrator privileges. Select Start, then Run, then type `secpol.msc`. This will bring up the Local Security Settings Management control window.

Expand Local Policies, then highlight Audit Policy, as seen below.



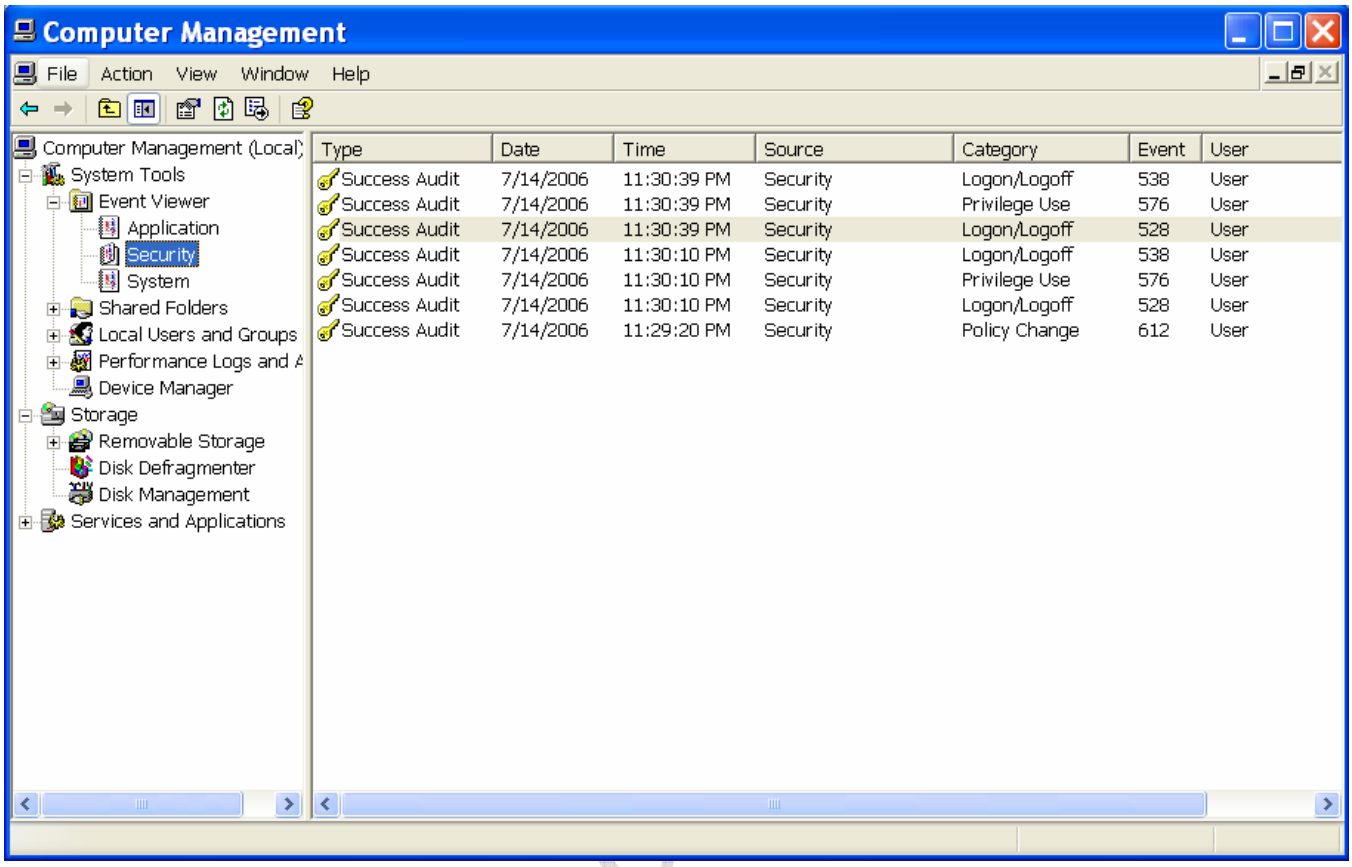
Before enabling audit tracking, view the Security logfile to see if there are any entries currently being tracked. Select Start, Run, then type `eventvwr`. The Event Viewer will often show no entries in the Security logfile, because auditing of security events is not enabled by default.

To track logins to user accounts, select the properties of the Audit logon events item by double-clicking. In the resulting dialog box, check the Success and Failure settings. Note that many more of these options may be desirable to audit, including account management and privilege use. Enabling tracking of too many events will lead to two problems. First, log file sizes will become enormous, and the Windows XP default logfile size is only 512KB. Excessive logging will result in older logs being deleted or new events not being logged. Secondly, logging too many events, while useful in an after-the-fact tracking expedition, will often result in information overload for an administrator. Like an email system with every message marked

"Urgent", over-zealous tracking will often mean that truly important messages are overlooked in a flood of minutia. Once the Success and Failure properties have been activated, select OK to apply the settings.

At this point, user account logon and logoff events will be tracked. To create entries in the log, Click Start, Log Off, then Switch User. If Switch User is not available, then select Log Off. Both actions will generate auditing information, but Switch User does not close any open applications and is faster to execute than Log Off and Log On. At any rate, once you have returned to the logon screen, reenter your login credentials and return to the Windows Desktop. By doing so, you will generate both logon and logoff events.

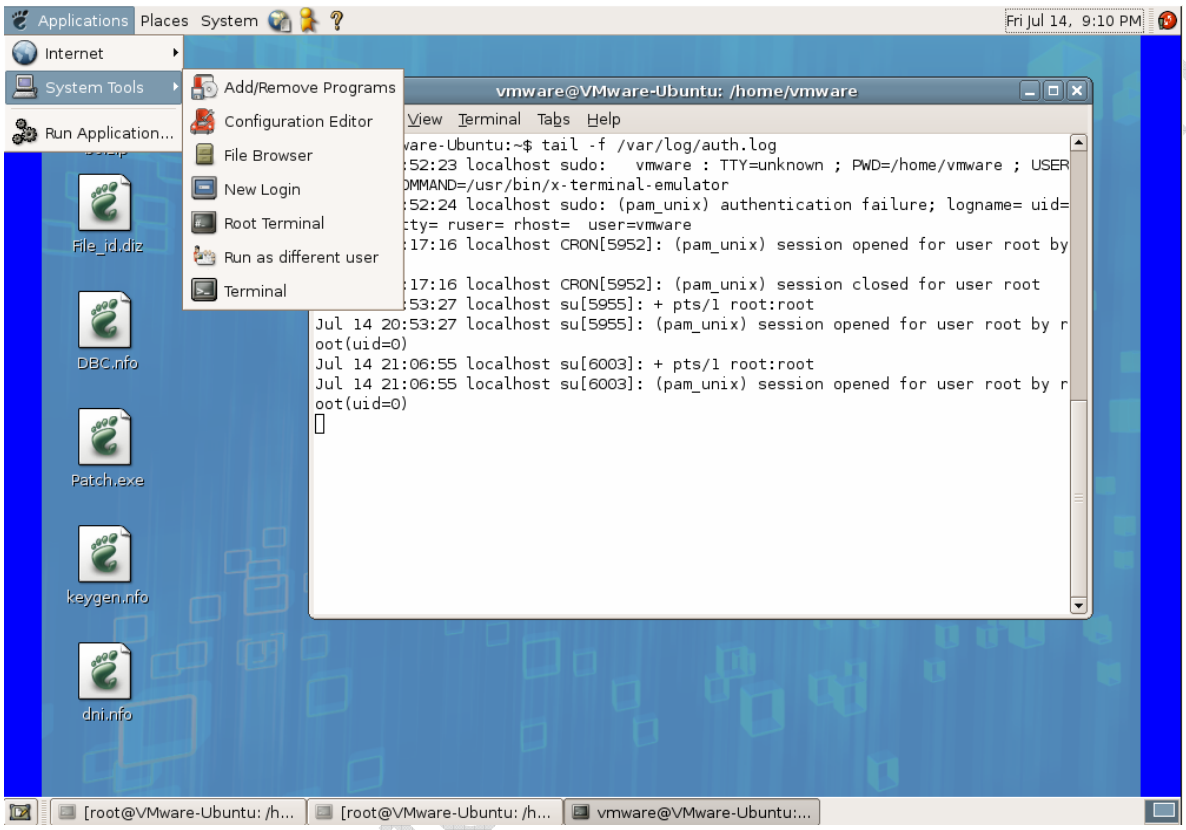
To view these events in Windows, select Start, then Run, then type `compmgmt.msc`. This will bring up the Computer Management Console. Under System Tools, expand Event Viewer, then Security to view the Security log. Note that Windows maintains three separate logfiles for Applications, Security, and System.



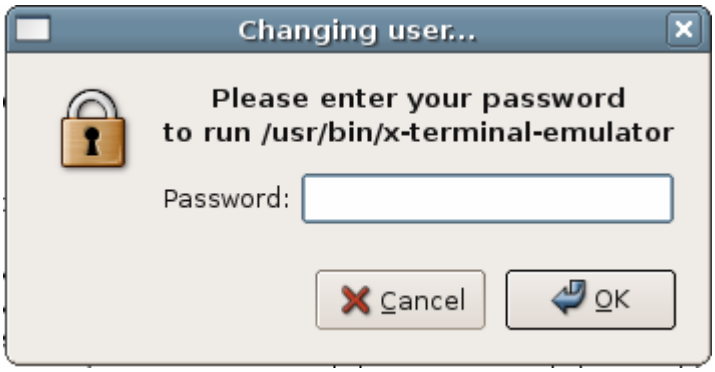
Lab Scenario: Audit Logging in Linux

Logging in Linux machines can vary greatly between different distributions. With minor modification, this part of the exercise can be applied to any Linux platform using the syslog facility. There are extensive additional options for logging besides the standard syslogd, including the “multilog” daemon from Daniel Bernstein. For this exercise, all work may be performed on the VMWare Virtual Browser Appliance, which is based on Ubuntu Linux. Mosts Debian based distributions should work in the same way with no modification, but other distributions such as Red Hat Enterprise Linux or SUSE Linux Enterprise Server may use additional log files or syslogd.conf parameters.

Assuming you have the VMWare Browser Appliance, ensure that it is started, then bring up a command prompt by selecting Applications, then system tools, then terminal. At the resulting command prompt, type the command `tail -f /var/log/auth.log`, as shown.



This command prints the last 10 lines of the auth.log log file, which is where authentication entries are stored by the syslog facility. It will also (via the –f flag) continue to show new entries as they are added to the logfile. To generate an entry in the log, select Applications, then System tools, then Root Terminal. This will result in a dialog box prompting for a password. The password is for the root user, and on the Browser Appliance is “vmware”.

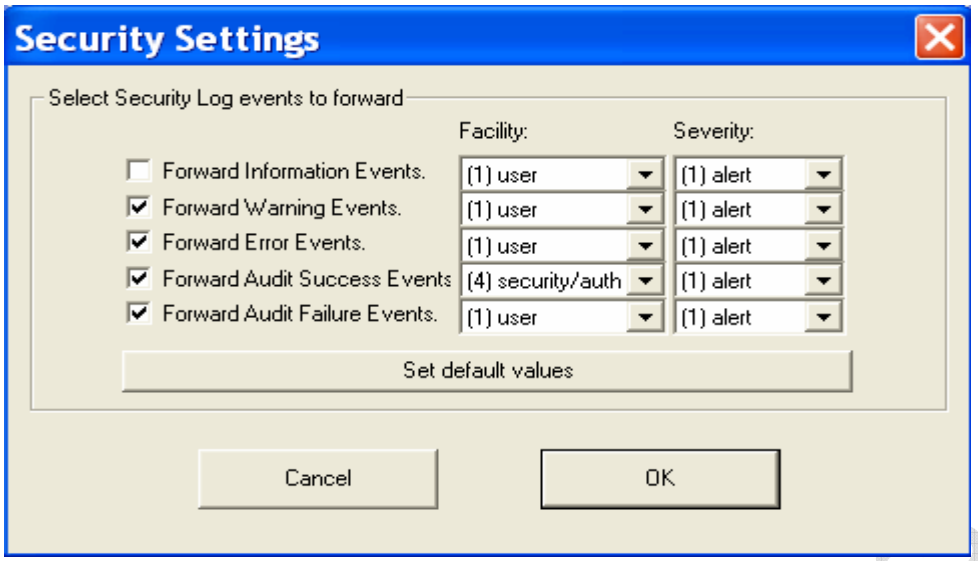


As the resulting command prompt window appears, you will notice that the logfile has new entries, indicating that there was a successful session opened for the user account. Once the Root Terminal has opened, type the command `su vmware` at the prompt. This will give you the permissions of the vmware user, but also generate logging information in the auth.log file.

Lab Scenario: Combining Audit logs from Window and Linux using syslog

Since Windows XP does not support syslog natively, a third-party application is required to support the syslog facility. The NTsyslog project is a GPL licensed program that will run as a service under Windows, monitor Application, System, and Security events, format them into a single line and send them to a syslog host. This program is easily installed on a Windows XP host by downloading the ISToolInstaller from the NTsyslog SourceForge webpage, extracting the two zip files, and starting SetupNTSyslog-1.13.exe. Install the software using the defaults suggested by the installer, and once it's finished installing, start the NTsyslog Service Control Manager.

Two changes are necessary to to activate the service in this environment. First, click the Syslog Daemons button to bring up another window where you will enter the IP address of the Linux machine (192.168.221.125) into the Primary Syslog Daemon dialog box, then select OK. Next, use the drop-down menu to select the Security EventLog to forward to the syslog server. Highlight it and click the EventLog button.



In the resulting Security Settings dialog box, check the "Forward Audit Success Events" box, and change the Facility for that selection to "(4) security/auth". Click OK, then on the main NTsyslog Service Control Manager windows, click the "Start Service" button. At this point, indicated events are being forwarded to the syslog server. (To verify this, start a Root Terminal window on the Linux machine, and type the command `tcpdump port syslog`. Afterward, Switch Users on the Windows machine to generate Security log entries. Any syslog traffic sent to the machine will show up as packets on the network, and will be printed to the prompt.) Unfortunately, while the events are being sent from the Windows machine, they are not yet being accepted by the Linux syslog facility.

To fully enable syslog, the daemon running on the Linux machine must be configured to accept syslog traffic from the network. Most often, this is done with a command-line option upon service startup. To modify this setup perform the following steps. At a Root Terminal command prompt, edit the sysklogd service startup file by typing `pico /etc/init.d/sysklogd`. Other text editors than pico may be used, and the command may be stored in the file /etc/sysconfig/syslog in other Linux distributions. Once the sysklogd file is on the screen, move

the cursor about 15 lines down to the `SYSLOGD ="-u syslog"` entry. Add the flag –r to the entry, so it looks like `SYSLOGD ="-r -u syslog"`. Exit and save by pressing Ctrl-X, then enter to say yes save, then enter again to use the existing filename. Finally, restart the syslog service by entering the command `/etc/init.d/sysklogd restart.`

You should immediately see traffic appear if you are running the tcpdump command, and entries should also appear in the auth.log file that you are still monitoring. Success! You are now monitoring two machines from a single logfile.

Points of Discussion

Care should be taken that the syslog machine tracks the data logged. A special point of discussion should be focused on several security weaknesses that may be introduced by this setup. Is the syslog facility vulnerable to a Denial of Service attack by sending additional messages to the now-networked syslog facility? Is there any proof against spoofed entries being submitted into the syslog? Does the nature of UDP traffic provide any guarantee that the event log messages will be delivered to the syslog server in a reliable or timely fashion? What steps could be taken to ensure that these risks are minimized as much as possible?

Enforcing password policies

Windows and Linux both have the ability to configure and enforce password policies. By default, most of the available restrictions are turned off, but this is very dependent on the specific system. For instance, Windows XP allows enforcement of “complexity requirements” that consist of 3 parameters: it must be 6 characters long, must not include part of the username, and contain characters from 3 of 4 classes of characters (Uppercase, lowercase, numbers, and non-alphanumeric). Linux password complexity enforcement is often provided by the cracklib facility, which provides 3 checks: it must not contain generated words from the username and

GECOS (or Full Name) entries, it must not have simplistic patterns, and it must not be in the dictionary. Beyond complexity, password enforcement also consists of restricting the validity of a password, how many times an account may attempt logon before it becomes locked out or disabled, how soon before expiration a user is warned, and others dependent on the system.

In this exercise, several password policies are enforced for Windows XP and for Linux systems, and some of the differences in default capability between the two are illustrated. Password complexity is required, regular password changes are enforced, and accounts will be disabled or locked with too many attempts to login.

Setup

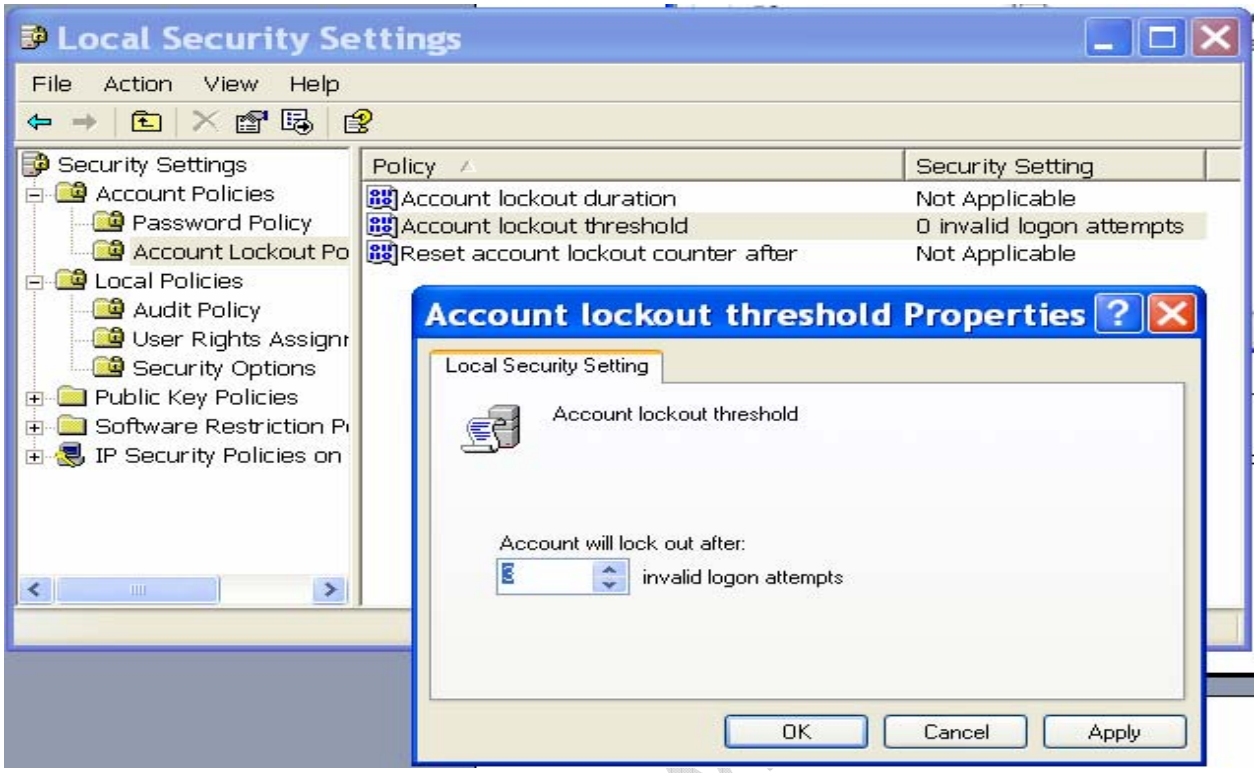
For the Windows lab scenario, start with a Windows XP Professional workstation that has not been attached to a domain or does not have Group Policy applied to it. For the Linux lab scenario, start with Ubuntu Linux with the cracklib libraries installed (verified by searching for libpam-cracklib in the synaptic package manager.) Either of these can be installed in a Virtual Machine, with Ubuntu easily available through the Browser Appliance.

Lab Scenario: Enforcing password complexity, aging, and lockout in Windows XP

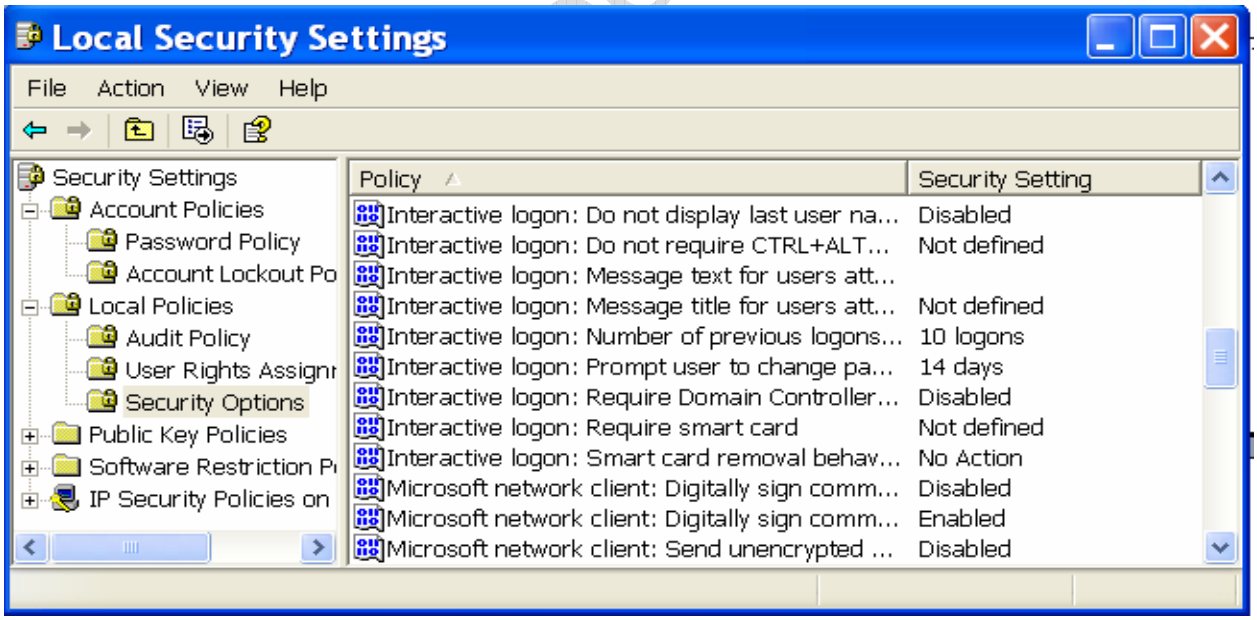
Login to Windows with Administrator privileges. Select Start, then Run, then type `secpol.msc`. This will bring up the Local Security Settings Management control window. Expand Account Policies, then highlight Password Policy, as seen below.

Select the properties of Maximum password age by double-clicking, and change the setting to 60 days. Set Password must meet complexity requirements to Enabled.

To enforce account lockout after bad attempts, highlight Account Lockout Policies, then change Account lockout threshold to 3. The default will be to change the Account lockout duration and Reset account lockout counter after setting to 30 minutes, so accept these defaults.



To enforce other related policies, expand Local Policies, then highlight Security Options.

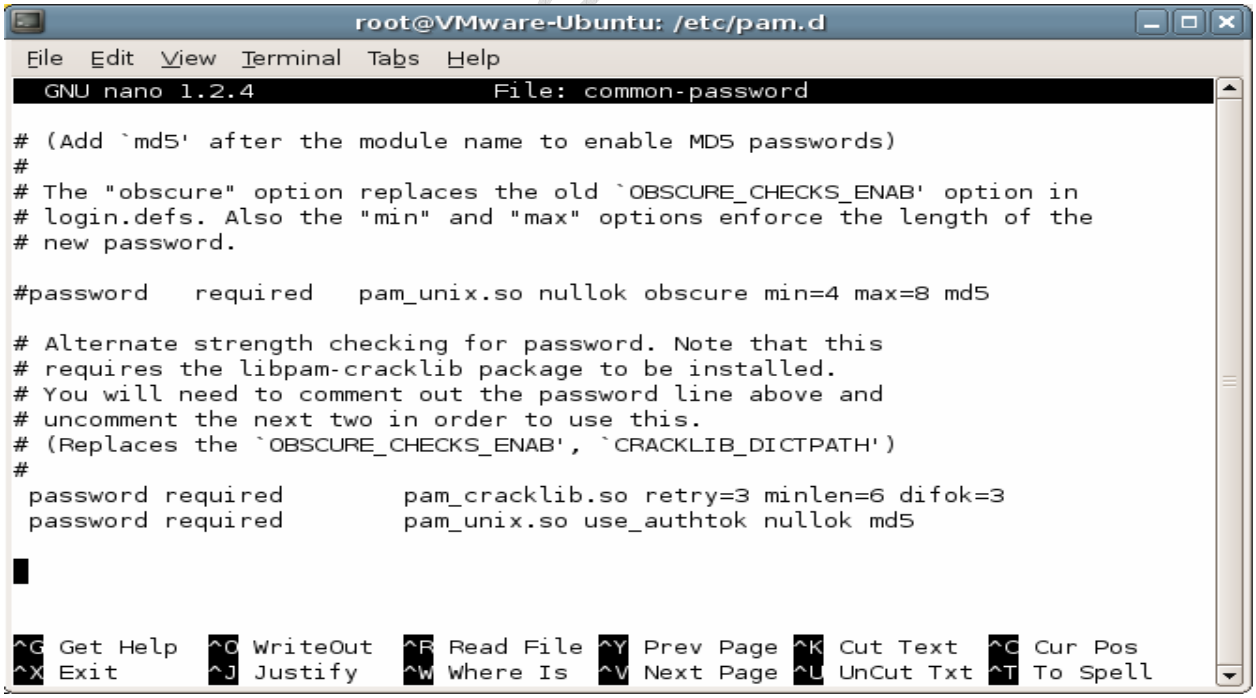


To verify that these policies were applied correctly, create a new user. During the creation process, attempt to set the password to be the same as the username. Once you have

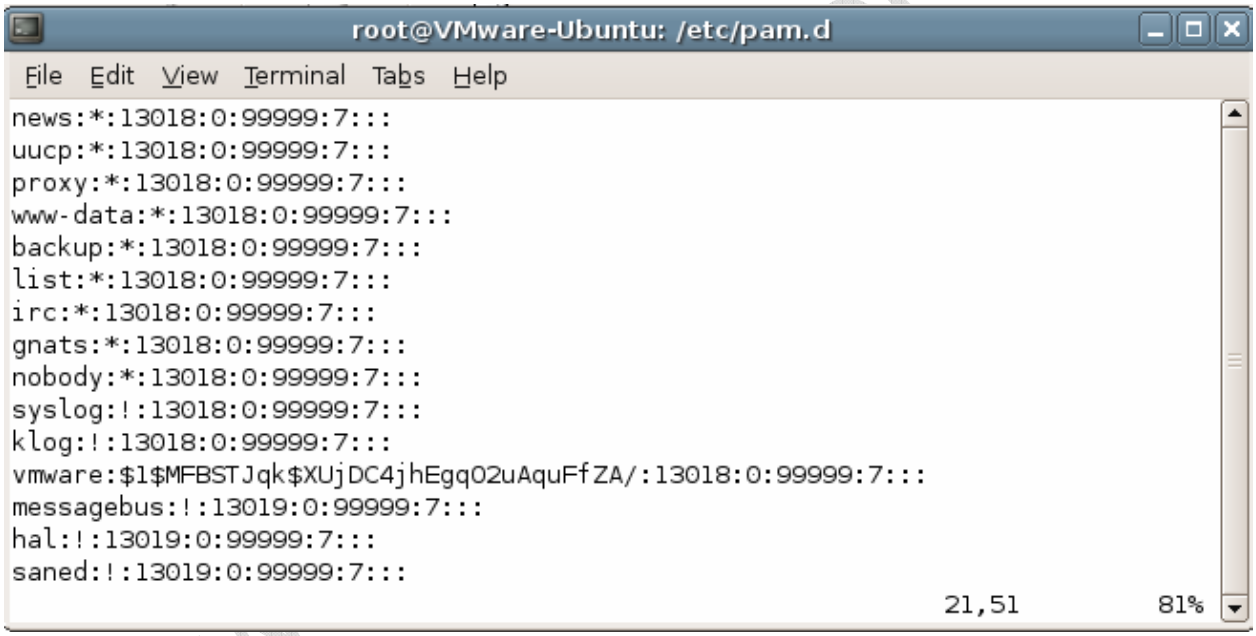
created the user with a valid password, attempt to login as that user using the incorrect password twice, then using the correct password. Do that again. Finally, logout and this time use the incorrect password 3 times before applying the correct password. Note that even though you have 4 unsuccessful login attempts in the previous 30 minutes, you were still allowed to login because the counter was reset upon a successful login. Also note that you now must wait 30 minutes to login, or have an administrative account reset your account status.

Lab Scenario: Enforcing password complexity, aging, and lockout in Linux

Create a Root Terminal window and type `pico /etc/pam.d/common-password`. Search about 15 lines down for a line the reads `password required pam_unix.so` (Which may have additional commands later on the line.) Insert a # at the beginning of the line to comment it out. Near the bottom of the file will be two lines that start #`password` and delete the # character from both lines. Save and exit the file by typing Ctrl-X, Y, enter.



To enforce a password change every 60 days, type the command `vipw -s`. (This will allow you to edit the /etc/shadow file in a safe manner, as it prevents a file with bad syntax from being saved to the active configuration, and it does require the vi or vim package to be installed. If these are not available, use the command `pico /etc/shadow`, but be extremely careful not to save a bad file to the system.) Search for the vmware user, and change the 99999 on the line to 60 by highlighting the first 9, typing x three times to delete three 9's, typing r, then 6 to change the next 9 to a 6, highlighting the next 9, typing r then 0. This will set the number of days before the password expires to 60 days. Save and exit by typing Esc, then :wq.



Finally, to automatically lock an account after too many login tries, you will need to edit two files in /etc/pam.d. Add the line `auth required pam_tally.so onerr=fail` to /etc/pam.d/common-auth, and add the line `account required pam_tally.so deny=3 unlock_time=1800 reset` to /etc/pam.d/common-account. The auth line will count the attempts to login, and the account line will disable the account if it attempts 3 unsuccessful logons for 30 minutes (1800 seconds) while resetting the lockout counter after a

successful login. Once these lines have been inserted, create the failure log by issuing the command `touch /var/log/faillog`.

To test these, at the Root Window, create a new user with `adduser patsy`. Login as patsy by starting a non-root terminal and entering the command `su – patsy` (no password should be required.) Once logged into the terminal as patsy, change the password with the passwd command. When prompted, try to change the password to the username, or to a short word. After choosing a valid password, logout with exit, then su – patsy again, this time providing the wrong password, as in the Windows lab above. Note that the account will be locked out for 30 minutes when this is done. To reset the account lockout counter, use the `faillog –r –u patsy` command.

Lab summary

Although Window and Linux have very different mechanisms for enforcing password policies, many of the same capabilities exist in both. By default, Windows has virtually no password checking or account lockout enforcement, and Linux has only username or length based checking, while allowing (optionally) that it be ignored. Linux, though, gives many more options for exactly how accounts are restricted, passwords are aged, and access is given. Options exist to change the ability to reset invalid login attempt counters, to specify greater password complexity and strength checking, and to control the lockout time on a per-user basis.

Conclusion

In the development of an Information Systems Security curriculum, individual Information Security course design needs to account for its place in the curriculum plan of study. Course objectives should be appropriate for the level of instruction, and both build on topics presented in previous courses and prepare students for more advanced topics in later ones. An

effective course will aid the Instructor in developing the course syllabus, the student in understanding the track of the course, and the potential employer or reviewer in understanding the breadth of topics covered.

References

* Alec Yasinsac, Jenny Frazier, and Marion Bogdonav, "Developing an Academic Security Laboratory", 6th National Colloquium for Information Systems Security Education 2002, June 3-7, 2002, Microsoft Headquarters, Redmon, Washington. http://www.cisse.info/history/CISSE%20J/2002/yasi.pdf

* Nieh, J. Leonard, O. C. EXAMINING VMWARE. Doctor Dobbs Journal 2000, VOL 25; PART 8, pages 70-79 http://www.ncl.cs.columbia.edu/publications/drdobbs2000.pdf

* Ji Hu, Dirk Cordel, Christoph Meinel. A Virtual Laboratory for IT Security Education. Proceedings of the Conference on Information Systems in E-Business and Egovernment (EMISA), Luxembourg, 6-8 Oct 2004, pp. 60-71 http://www.informatik.uni-trier.de/~meinel/papers/Trier-Emisa04-Hu.pdf

* Patricia Y. Logan. Crafting an Undergraduate Information Security Emphasis Within Information Technology. Journal of Information Systems Education, Vol 13(3) http://www.jise.appstate.edu/13/177.pdf

* Patricia Y. Logan, Allen Clarkson. Teaching students to hack: curriculum issues in information security. Technical Symposium on Computer Science Education, Proceedings of the 36th SIGCSE technical symposium on Computer science education (2005)

* Dr. XiangdongLi, Dr. Lin Leung. Development of a Security Education Program at a Minority Institution. Proceedings of the 10th Colloquium for Information Systems Security Education (2006) http://www.cisse.info/proceedings10/pdfs/papers/S05P04.pdf

Palmer, Michael. *Guide to Operating Systems Security*. Course Technology 2004

Wake Tech Community College CIS Course Syllabus Security Administration I NET 222 http://www.waketech.edu/curred/cis/syllabi/spring_syllabi/200601net222.html

Certified Internet Web Security Professional Series Course 2 – Operating Systems Security http://www.ciwcertified.com/publicreadaccess/catalog/outlines/CIW_Security_Professional/Oper_Sys_Sec_v3_1_Outline.pdf (visited July 14, 2006)

VMWare Player. http://www.vmware.com/products/player/ (visited July 14, 2006)

VMWare Server http://www.vmware.com/products/server/ (visited July 14, 2006)

VMTN Virtual Appliances http://www.vmware.com/vmtn/appliances/ (visited July 14, 2006)

Wireshark – packet capture software http://www.wireshark.org (visited July 14, 2006)

Course Technology –i.t.link –Thomson Course Technology's Magalog http://www.course.com/itlink/postsecondary/archives/fall02/onthehorizon.cfm (visited July 14, 2006)

North Carolina Community College System Common Course Library http://www.ncccs.cc.nc.us/Programs/docs/Common-Course-Library/CCL-Prefix-S/SEC-28Sep2005.pdf (Visited July 14, 2006)

Research Triangle Park http://www.rtp.org/ (visited July 14, 2006)

Top 100 Network Security Tools by Fyodor, developer of the nmap program http://sectools.org/index.html (visited July 14, 2006)

Netcat - the TCP/IP Swiss Army Knife http://www.vulnwatch.org/netcat/ (visited July 14, 2006)

D. J. Bernstein The multilog program http://cr.yp.to/daemontools/multilog.html (visited July 14, 2006)

Syslog and Windows – Windows Event Logs to Syslog Server http://troy.jdmz.net/syslogwin/ (visited July 14, 2006)

NTsyslog project http://sourceforge.net/projects/ntsyslog/ (visited July 14, 2006)

Windows XP Documentation - Password must meet complexity requirements http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/504.mspx?mfr=true (visited July 15, 2006)

GECOS Field http://en.wikipedia.org/wiki/Gecos_field (visited July 15, 2006)

Cracklib RPM http://rpmfind.net/linux/RPM/fedora/4/i386/cracklib-2.8.2-1.i386.html (visited July 15, 2006)

Linux password and shadow file formats http://www.tldp.org/LDP/lame/LAME/linux-admin-made-easy/shadow-file-formats.html (visited July 15, 2006)

The Linux-PAM System Administrator's Guide http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam-6.html (visited July 15, 2006)

Linux Security Cookbook / HOWTO / Guide http://www.puschitz.com/SecuringLinux.shtml#LockingUserAccountsAfterTooManyLogInFailures (visited July 15, 2006)