

SEC401 | SECURITY ESSENTIALS BOOTCAMP STYLE

401.6

Unix/Linux Security



SANS

THE MOST TRUSTED SOURCE FOR INFORMATION SECURITY TRAINING, CERTIFICATION, AND RESEARCH | sans.org

Copyright © 2016, The SANS Institute. All rights reserved. The entire contents of this publication are the property of the SANS Institute.

PLEASE READ THE TERMS AND CONDITIONS OF THIS COURSEWARE LICENSE AGREEMENT ("CLA") CAREFULLY BEFORE USING ANY OF THE COURSEWARE ASSOCIATED WITH THE SANS COURSE. THIS IS A LEGAL AND ENFORCEABLE CONTRACT BETWEEN YOU (THE "USER") AND THE SANS INSTITUTE FOR THE COURSEWARE. YOU AGREE THAT THIS AGREEMENT IS ENFORCEABLE LIKE ANY WRITTEN NEGOTIATED AGREEMENT SIGNED BY YOU.

With the CLA, the SANS Institute hereby grants User a personal, non-exclusive license to use the Courseware subject to the terms of this agreement. Courseware includes all printed materials, including course books and lab workbooks, as well as any digital or other media, virtual machines, and/or data sets distributed by the SANS Institute to the User for use in the SANS class associated with the Courseware. User agrees that the CLA is the complete and exclusive statement of agreement between The SANS Institute and you and that this CLA supersedes any oral or written proposal, agreement or other communication relating to the subject matter of this CLA.

BY ACCEPTING THIS COURSEWARE YOU AGREE TO BE BOUND BY THE TERMS OF THIS CLA. BY ACCEPTING THIS SOFTWARE, YOU AGREE THAT ANY BREACH OF THE TERMS OF THIS CLA MAY CAUSE IRREPARABLE HARM AND SIGNIFICANT INJURY TO THE SANS INSTITUTE, AND THAT THE SANS INSTITUTE MAY ENFORCE THESE PROVISIONS BY INJUNCTION (WITHOUT THE NECESSITY OF POSTING BOND), SPECIFIC PERFORMANCE, OR OTHER EQUITABLE RELIEF.

If you do not agree, you may return the Courseware to the SANS Institute for a full refund, if applicable.

User may not copy, reproduce, re-publish, distribute, display, modify or create derivative works based upon all or any portion of the Courseware, in any medium whether printed, electronic or otherwise, for any purpose, without the express prior written consent of the SANS Institute. Additionally, User may not sell, rent, lease, trade, or otherwise transfer the Courseware in any way, shape, or form without the express written consent of the SANS Institute.

If any provision of this CLA is declared unenforceable in any jurisdiction, then such provision shall be deemed to be severable from this CLA and shall not affect the remainder thereof. An amendment or addendum to this CLA may accompany this courseware.

SANS acknowledges that any and all software and/or tools, graphics, images, tables, charts or graphs presented in this courseware are the sole property of their respective trademark/registered/copyright owners, including:

AirDrop, AirPort, AirPort Time Capsule, Apple, Apple Remote Desktop, Apple TV, App Nap, Back to My Mac, Boot Camp, Cocoa, FaceTime, FileVault, Finder, FireWire, FireWire logo, iCal, iChat, iLife, iMac, iMessage, iPad, iPad Air, iPad Mini, iPhone, iPhoto, iPod, iPod classic, iPod shuffle, iPod nano, iPod touch, iTunes, iTunes logo, iWork, Keychain, Keynote, Mac, Mac Logo, MacBook, MacBook Air, MacBook Pro, Macintosh, Mac OS, Mac Pro, Numbers, OS X, Pages, Passbook, Retina, Safari, Siri, Spaces, Spotlight, There's an app for that, Time Capsule, Time Machine, Touch ID, Xcode, Xserve, App Store, and iCloud are registered trademarks of Apple Inc.

Governing Law: This Agreement shall be governed by the laws of the State of Maryland, USA.

SANS SEC401 Day 6

Linux/Unix Security Essentials

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

SANS SEC401 Day 6

In order to provide the most basic forms of incident detection, prevention, mitigation and response to Linux/Unix operating systems, one must understand the basic file system structure, general commands and how they interact with the environment, file and directory permissions, common Unix services, logging, and a variety of security enhancements and tools available to them. This day is designed to bring all this general knowledge together and provide a detailed overview and hands-on experience in Securing Linux/Unix systems.

The following sites and references were used throughout this module:

- 1) www.zzee.com/solutions/
- 2) en.wikipedia.org
- 3) publib.boulder.ibm.com/infocenter/systems/
- 4) www.ibm.com/developerworks/linux/



CYBER DEFENSE CORE SECURITY ROADMAPS



System Administrator/Security Administrator

The core courses in the Career Roadmap focus on teaching system and security administrators how to blend fundamental information security defense into their jobs based on their unique knowledge of the systems they maintain. As the system and security administrators advance in their careers, a deeper knowledge of all security functions, including technical security policy foundations, is critical both for individual growth and to maintain defense against evolving security threats to any organization. These essential core foundational security courses will show these professionals how to successfully apply and integrate critical security concepts.

Applicable Job Titles/Roles

- System Administrators
- Network Administrators
- Database Administrators
- Network Operations



Security Analyst

The core courses in the Career Roadmap focus on teaching security professionals how to analyze security solutions and develop cost-effective solutions. Security analysts need to be able to assess risk across a range of complex environments. An understanding of creative countermeasures is required to design various security solutions that can be deployed across an organization. This critical role requires understanding the importance of cybersecurity and a risk-based approach to help protect the organization. An analyst must be able to perform continuous monitoring and implement automated solutions, which in turn will enable the analyst to audit and validate overall security across all aspects of an organization.

Applicable Job Titles/Roles

- Security Engineers
- Security Analysts
- Data Center Operators
- Help Desk/Technicians



Security Engineer

The core courses in the Career Roadmap focus on teaching the critical technical skills required to implement and maintain a range of risk-based security solutions. Many personnel focus solely on implementing effective defensive solutions across the enterprise. These professionals need more than a core foundation of expertise; they must have deeper technical knowledge to be able to solve a variety of complex problems involving cybersecurity. Defense specialists require a working knowledge of the critical technology and strategy not only to defend against a variety of attacks but also to perform timely detection. Both preventive and detective components are required to implement and integrate a cybersecurity strategy.

Applicable Job Titles/Roles

- Security Analysts
- Security Architects
- Security Auditors
- Security Engineers



Operations Management

The core courses in the Career Roadmap focus on teaching the skills required to understand and run security operations within an organization. Security is a critical part of organizational operations. Operational managers must understand the language of security, how it can impact an enterprise, and strategies that can be used to properly secure an organization. As threats continue to become more sophisticated, it is critical that anyone overseeing technology or involved in day-to-day operations understands the various approaches that can be used to reduce the risk to an organization. Operations managers must know what questions to ask to make sure staff is focused on the highest priority areas.

Applicable Job Titles/Roles

- Audit Compliance Management
- Consultants/Directors
- IT Management
- Data Center Managers



Cybersecurity Manager/Officer

The core courses in the Career Roadmap focus on teaching executives the language and importance of cybersecurity. Cybersecurity has entered the boardroom. Leaders in every organization need to have a high level of understanding of security to ensure that decisions are aligned with the organization's risk posture. Managers, directors, vice-presidents, and executives need to be able to ask the right questions to address issues that could affect the reputation and success of the organization. This career track will equip managers and executives to be fluent in the language of security and what it means to make proper risk decisions.

Applicable Job Titles/Roles

- Chief Information Officers
- Chief Information Security Officers
- Director/Security Consultants
- Security Managers
- Business Unit Managers

This page intentionally left blank.

SANS CYBER DEFENSE SPECIALIZED ROADMAPS

SEC440 → **SEC480** → **SEC565**

Security Architect

The core courses in the Career Roadmap focus on planning, designing, and implementing an effective security solution. In order for security to be effective it must be customized to the unique business, mission, and risks an organization faces. The security strategist must be able to identify core metrics and use them to design and oversee the implementation of a security system and network architecture. Having a secure robust network architecture is critical for an organization to have effective security.

Applicable Job Titles/Roles

- Security Managers
- System Architects
- Data Center Analysts
- Design Engineers

SEC501 → **SEC511** → **SEC503** → **FOR572**

Security Operations Center (SOC) Analyst

The core courses in the Career Roadmap focus on building, running, and deploying a SOC. Security has become critical for the success of an organization constantly under attack. Defense against the vast array of attacks requires continuous monitoring of the network and assessment of the security infrastructure. Properly trained people who can detect and respond to attacks are critical to the overall success and security of an organization.

Applicable Job Titles/Roles

- Security Consultants
- Security Operations Supervisors
- SOC Managers
- Security Operations Directors

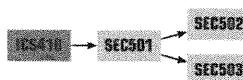
MIGT419 → **SEC566** → **AUD507** → **SEC511**

Security Risk Officer

The core courses in the Career Roadmap focus on assessing and analyzing risk and using that information to guide the priorities for security. In order for organizations to be successful in security, they must take a risk-based approach. Risk allows an organization to identify the vulnerabilities that have the biggest impact, based on the threats that have the highest likelihood of success, and which are most linked to the organization's critical assets. Proper metrics that map back to risk are used to assess and verify that an organization's security program is focused on the correct areas.

Applicable Job Titles/Roles

- Risk Engineers
- System Managers
- Risk Officers
- Auditors

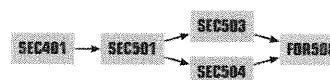


Industrial Control Systems (ICS) Analyst

The core courses in the Career Roadmap focus on teaching how to assess, implement, and secure ICS. Anyone who works in critical infrastructure needs to understand the associated threats and methods for security and the proper ways to protect systems that support a variety of ICS environments. ICS represent unique challenges not only in terms of threats, but also in terms of the unique methods that must be used to reduce risk to these systems. The focus is on providing an appropriate level of security based on the security challenges that these organizations face.

Applicable Job Titles/Roles

- Control System Engineers
- Control System Managers
- Operational Analysts
- System Administrators



Intrusion Analyst

The core courses in the Career Roadmap focus on teaching the foundations of security, as well as on the prevention and detection of threats. The most masterful prevention measures may be circumvented by skilled attackers. Successful attacks must be quickly identified to minimize the damage. The focus is on implementing appropriate prevention methods, rapid detection and assessment of malicious activity, and containment of harm in the aftermath of a successful attack.

Applicable Job Titles/Roles

- System Administrators
- IDS Specialists
- Security Analysts/Specialists
- SOC Engineers
- Intrusion Detection Analysts

This page intentionally left blank.

Module 29: Securing Linux/Unix

UNIX OVERVIEW – STRUCTURE AND COMMANDS

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

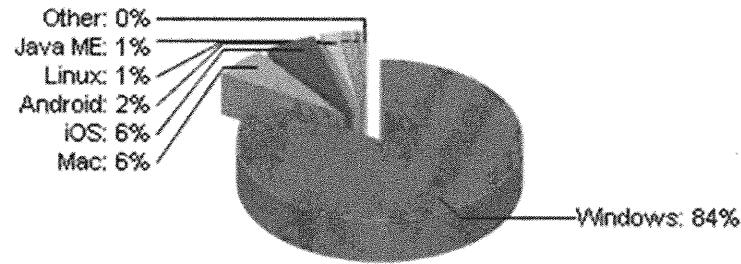
Module 29: Securing Linux/Unix

UNIX OVERVIEW - STRUCTURE AND COMMANDS

This section intentionally left blank.

Operating System Market Share

Total Market Share



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Operating System Market Share

One recent study found that some version of Windows was loaded on over 80% of all computers.[1] That is an amazing market share, and it means that when you talk to people about computers, they are most likely to be familiar with some version of Windows to some extent—more so than Macintosh computers and much more so than Unix or some variation of it.

References

- [1] <http://marketshare.hitslink.com/report.aspx?qprid=8>

Linux and Windows

Desktops and Servers

- Over 80% of all computers run Windows.
 - Most people have used Windows, many exclusively.
 - **Windows** started as a desktop OS, but is now a respected server platform:
 - Old: Single-user platform
 - New: Multiple processes for multiple users at the same time
 - **Unix** took the opposite route:
 - Old: Installed on servers with many simultaneous users
 - New: Used as a desktop OS

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Linux and Windows: Desktops and Servers

Although it's vital to be comfortable with the most common operating system (OS) in use today, there are substantial (and some say vital) reasons to know about the others as well. There's always room for people to argue over, which OS is "better" than another, but the bottom line is they all have advantages, limitations, and liabilities. Because they all get together on big, fast networks, they represent threats to each other that security-minded people need to know about and protect against.

Which OS you choose to use depends on what you want to do with a computer, and that's another reason to be familiar with everything that is available. Although Windows has morphed into an OS one might just as likely find on a server as on a person's desktop computer, it wasn't always that way.

Windows started its commercial life as a desktop OS only, something used by one person to do (basically) one thing at a time until the job was done. Think of a person in an office writing a report or a letter for a project.

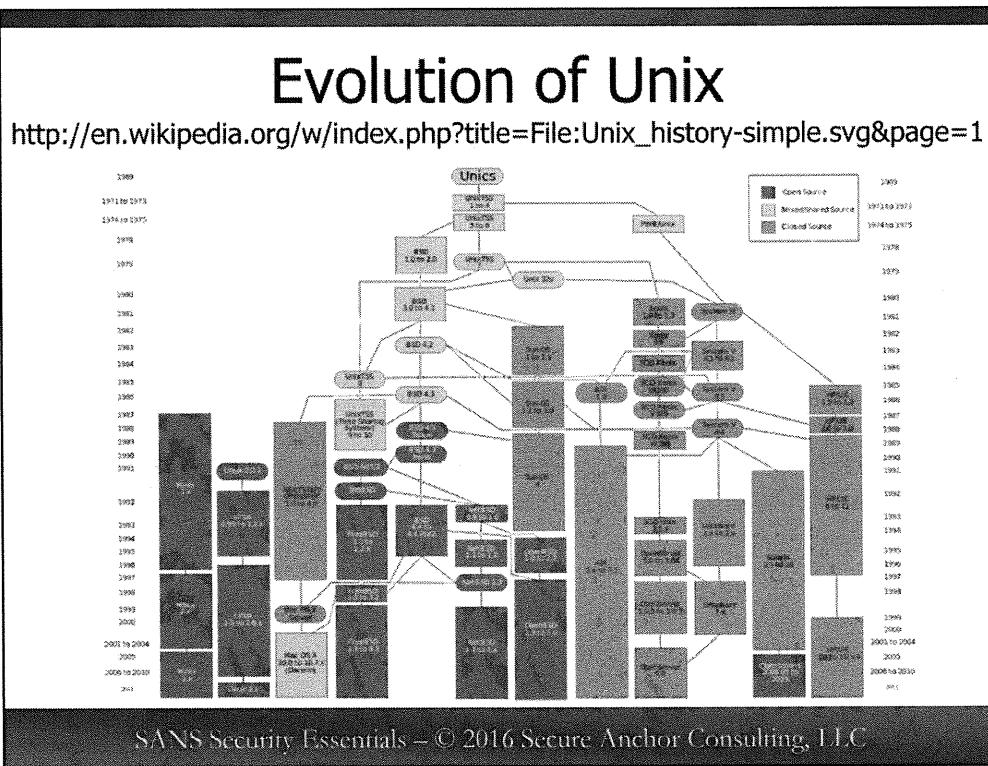
As things progressed, there came a need for several people to work on a project together, and that required them to all work on (or at least access remotely) the same computer running the same OS. For many reasons, including the need for people to collaborate and to keep all their files together, Microsoft was prompted to develop a server version of Windows that would perform multiple tasks for multiple users simultaneously. So, all of a sudden you could run a desktop version of Windows on your desktop computer and a server version of Windows on a server, which juggled and synchronized everyone's requests.

Interestingly, Unix developed along the same lines, but running in the opposite direction. Unix was conceived in 1969 as an OS that was going to run on a server with many users all doing different things at the same time.[1] The OS needed to manage not only these users and their processes, but also all the background programs that

made these processes available to the users in the first place. It became stable and dependable, and people started businesses that did nothing but write and maintain a version of Unix as their main source of income. One guy even thought of writing his own version and then shared it with the rest of the world expecting no money in return.[2]

References

- [1] <http://en.wikipedia.org/wiki/Unix>
- [2] <http://www.faqs.org/docs/artu/ch02s01.html>

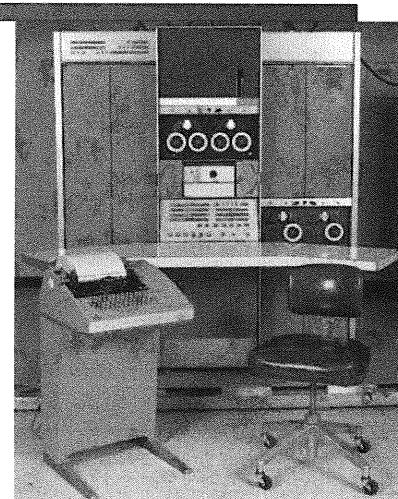


Evolution of Unix

Many things combined to give us the Unix landscape we see today. Legal considerations, advances in technology, mass collaboration among thousands of programmers, and the changing needs of users and administrators all shape what is put in software and how it is all put together. If you look at the chart, you can see the interesting evolution of Unix over the past 40+ years.

Different Variants of Unix

- Cygwin for Windows
- Mac OS X (BSD)
- Two major Linux distributions:
 - Ubuntu (Debian)
 - Fedora (Red Hat)



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Different Variants of Linux

Keep in mind that as Windows evolved from a desktop-only OS into desktop and server versions, and that Unix took the same route in reverse, *both* are available today for both uses. Even Mac OS X has a server edition to go with its desktop version. In general, the server edition of a particular OS is usually streamlined for multitasking, and it lacks the type of core software applications one would find on in a desktop OS (that is, no word processing, e-mail client, spreadsheet software, and so on).[1]

Unix Overview

Because Unix contains so many different kinds of software that perform similar functions (think about how many different calculator programs people may have come up with) and because this software can be packaged together for a specific goal or job, there are lot of different ways to put together a "complete" Unix suite. In other words, there is no collection of all programs known to run in Unix. You get someone's (or a group of people) take on which programs are useful and which ones are not useful.[2]

Summary

Unix is available to the student and user alike and it comes in many different forms. Some forms don't require you to install an entire operating system. Cygwin allows some Unix software to run directly in Windows. OS X for Macintosh computers is a Unix/BSD variant that is popular but completely proprietary. Finally, there are many full-blown Linux packages called distributions for many different purposes.

References

[1] <http://www.december.com/unix/tutor/overview.html>

[2] <http://tldp.org/LDP/lame/LAME/linux-admin-made-easy/linux-overview.html>

Cygwin for Windows (1)

- Libraries for compiling Linux source:
 - *Some* Linux software can run in Windows
 - You must have access to the source code
- Cygwin isn't Linux
 - *Some* Linux commands and utilities
 - NOT a "Linux emulator" for Windows
- You could compile server software in Windows
- Not a replacement for running Linux in a virtual machine

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Cygwin for Windows (1)

Introduction

The Cygwin project is clear about what it will and will not do for users. It is a neat way to make some useful Linux tools available in Windows and can greatly expand the power Windows already has.[1] It can also make the switch to Windows easier if you started in computers with Linux or MacOS X. It doesn't make Windows capable of running everything and anything Linux has to offer.

Cygwin

In short, Cygwin installs into Windows (in all versions except CE)[2] like any other program; however, it's not just a program. It's more like an interface to a whole lot of software. Cygwin can be a small installation of a group of Linux commands or it can be an entire environment of development tools that have a lot of power. Some of these tools are used to develop software for the GNU Project (GNU stands for GNU's Not Unix)[3] and when "moved" over to Windows, they allow programmers to do some neat advanced programming in Windows. This in itself is useful, but even if you're not a programmer, simply installing Cygwin gives you access to some Linux utilities that don't have an equivalent on the Windows command line.

Installing Cygwin

Cygwin itself is small; it is one setup.exe file. Once you run this program, it will proceed to fill up your hard drive with goodies if you want it to!

You'll be greeted with a menu that allows you to choose software that's already been compiled in Windows, and there are a lot of choices. You can also ask Cygwin to download the source code for a specific program if you like. Everything from games to advanced security tools are available. You can even download compilers and, if you're adventurous, you can start to compile programs that aren't already on the Cygwin list.[4]

A "base" install will give you a nice (but small) selection of Linux shells and commands, and you can add more programs later (by running the original setup.exe file again) if you like and as you get more familiar with Linux. However, if you are already familiar with Linux, it's easy to find the edges of the Cygwin install.

When you're in a real Linux installation, there are literally thousands of commands at your fingertips. If a command isn't specifically included in the Cygwin base, you either have to hope it is already compiled for Cygwin or you're going to have to locate the source code for that command and compile it for Cygwin yourself.

Linux Versus Cygwin

However, just because you find something useful in Linux doesn't necessarily mean you can make it run in Cygwin and in Windows. It might and it might not. It depends on a lot of different variables, many of which are beyond your control. If you can find the source code for that application, you might be able to, but you have to know how to compile programs and that's not always fun when dealing with libraries and dependencies.^[5]

If you were working in an exclusively Windows environment and you wanted to use Linux server software, such as the Apache web server, on one computer you could actually choose this at the (version 1.5) Cygwin menu. Once installed, little would stop you from standing it up on your network as a production server.

The bottom line is that Cygwin is perfect for trying things out or as way to study how things work, but not for something you and other people are depending on to work consistently. The developers of a server program reasonably expect to have access to critical command-line tools always available in Linux, and when they are not there, unexpected behavior is likely. Don't use Cygwin for such purposes. Load Linux on the computer in question to get the reliability Linux server software is famous for.

If you absolutely have to run Linux server software in Windows, you'll want to look into a virtual machine.

Summary

While it's possible to get server software to run in Cygwin, it's a bad choice to do this because Cygwin won't necessarily provide everything the server software needs to be dependable. It's okay to add some Linux functionality to Windows with Cygwin, but for production servers, either a full install of Linux or a virtual machine (covered later in this course) would be a much better route. Cygwin is ideal for desktops and labs.

References

- [1] <http://cygwin.com/>
- [2] <http://x.cygwin.com/>
- [3] <http://en.wikipedia.org/wiki/Cygwin>
- [4] <http://www.physionet.org/physiotools/cygwin/>
- [5] <http://www.redhat.com/software/cygwin/>

Cygwin for Windows (2)

Where Does Cygwin Fit In?

- Allows detailed scripting in Windows:
 - Less critical with PowerShell
- Get Windows to interact with Linux-specific services
- GNU tools in Windows:
 - Bring powerful command-line tools to Windows

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Cygwin for Windows (2): Where Does Cygwin Fit In?

Introduction

Cygwin isn't Linux. Cygwin doesn't turn your Windows computer into a Linux emulator. However, there are definitely situations where making a Windows computer a little "Linux smart" (or at least "Linux aware") is a good thing. Maybe company policy forbids you from wiping a hard drive to install a full Linux distribution. Maybe there are Windows commands and functions you simply can't live without so you can't convert over to Linux wholesale. Cygwin still has some good uses and deserves a place in your toolbox precisely because it is a "half way" solution. [1]

Power of Cygwin

One thing Cygwin is particularly good for is adding powerful scripting to Windows that otherwise would not be possible.[2] There are times when an administrator wants certain things to happen automatically, and without any user intervention. Although the built-in Windows command line features provide some excellent functions, there are just some things it either can't do easily or can't do at all. Here's an example. Say you need to have certain log files automatically e-mailed to an administrator once a day, but the file can't be over 5,000 lines long. If it is, the file needs to be truncated down to no more than 5,000 lines. Can you even do that with the Windows scripting language? If you could, it would likely be a pretty lengthy file. With the correct Cygwin installed, it boils down to three lines:

```
cat logfile | sed -e '1,5000!D' > temp  
mv temp logfile  
mutt -s "Log report - 5K" bob@admin.com < logfile
```

That's hard to beat as far as efficiency goes. Detailed, involved, and complicated scripting has been going on in Linux for years and years because it hails from a server background. Windows is still relatively new to the server world (comparatively speaking) and it's just not that far developed.[3]

Functionality of Cygwin

Keep in mind that if you add Cygwin to a Windows install, you not only have added commands available, you can also now access and make use of literally thousands of scripting hints and examples on the Internet available to you via Google. In this way, Cygwin not only adds functionality to Windows, but it also brings with it an almost never-ending archive of real-world examples you can capitalize on.[4]

Cygwin can also add some connectivity options to Windows. Perhaps you have a Windows machine that needs to interact with a service on a Linux server. You can either make the Linux server speak a Windows protocol (complicated) or effectively adapt Windows by installing whatever native Linux libraries and programs it needs in Cygwin (easy!). A large array of disks on a Linux server providing network storage is a good example of this type of situation.

Cygwin Commands

There are some Linux commands that have no equivalent in Windows and are handy to have around. Cygwin is a great way to add these to your Windows box. The domain name specific command dig is an excellent example; it's going to give you information that is much more complete and useful to the administrator than nslookup.

Cygwin makes it possible to use GNU development tools when writing Windows or Linux code. These tools are tried and true and won't work in Windows without Cygwin. If you're a programmer, this can be a real godsend.[5]

You may want to use your Windows computer as a display server for a remote X Windows Server across the network. When it's set up, you'll be able to "drive" the remote X Windows box as if you were right on the console, with only your graphical movements and commands going across the network.

Summary

Cygwin is a sort of "half-way" step toward Linux. This can be viewed as useful because you can add a lot of power to your scripting (batch files) by installing Cygwin and using some of the command-line tools that are in Linux. You might also want to make a Windows box interact with a Linux service, and Cygwin offers a quick and reasonably easy way to add this functionality. Some Linux commands have no Windows equivalent (such as dig) and Cygwin allows you to add those commands without having to install Linux on the computer.

References

- [1] <http://www.redhat.com/software/cygwin/>
- [2] <http://lifehacker.com/software/command-line/geek-to-live-introduction-to-cygwin-part-iii-scripts-packages-and-more-181282.php>
- [3] <http://www.physionet.org/physiotools/cygwin/>
- [4] http://www2.warwick.ac.uk/fac/sci/moac/currentstudents/peter_cock/cygwin/
- [5] <http://cygwin.com/>

Mac OS X (1)

- Market share / popularity:
 - Second most popular OS in the world
 - Approximately a 6% install base
- History: MacOS previous to OS X not considered high powered:
 - Seemingly geared for creating digital graphics
 - Not particularly robust as a workhorse desktop
 - Certainly not as a server
- OS X changed the attitudes of many

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Mac OS X (1)

Introduction

A recent study on market share found over 80% of all computers run some version of Windows. It also found that around 6% of computers run Apple Inc.'s Mac OS X.[1] Taking this research into account, one can say with reasonable certainty that OS X is the second most common OS in the world.

MacOS

MacOS releases previous to OS X were often not considered high powered by serious computer users and not quite ready for serious use. There have always been people wholly dedicated to older MacOS versions because they were viewed as being particularly well suited for graphic work and desktop publishing in general. Some considered the whole MacOS approach more intuitive for beginning users than Windows or Linux, making it, in their view, a decent first desktop OS. At the same time, these folks didn't consider older MacOS versions to be robust enough to put up with the hard use a software developer or mathematics major might subject it to.[2]

Pre-MacOS X

Pre-OS X releases were known for having particularly "noisy" network protocols. One Appletalk printer looking for a network address could effectively saturate very fast fiber-optic networks as it continually broadcasts many times a second for an Apple application to give it the information it needs.

The old Apple Finder program (similar to Windows' Network Neighborhood) located network resources by constantly pounding the network with inquiries (as opposed to passively listening for things to make noise). On a small network, such as one a user might build at home, this is no big deal. But on a large network with lots of different OSes running on a lot of different servers, it created real bandwidth issues. So for years, Mac users on networks were asked to use Finder sparingly, if at all.[2]

When Mac OS X came out and people had a chance to work with it at length, a lot of attitudes changed permanently. MacOS had come of age with version X.

Summary

Mac OS X is the second most common OS. Before Apple released OS X, many considered MacOS as a reasonably light-duty desktop—not something you would depend on in a server role. Older MacOS versions had a reputation as being "chatty" on networks and could cause problems on heterogeneous networks. OS X represented a complete departure from old MacOS models and changed the attitudes of many.

References

- [1] <http://marketshare.hitslink.com/report.aspx?qprid=8>
- [2] http://en.wikipedia.org/wiki/Mac_OS

Mac OS X (2)

Security-Specific Features Built In

- Password assistant: Grades the strength of passwords
- FileVault: Home folder encryption with 128-bit AES
- OpenSSH: Encrypted
- Ipfw-based firewall
- Most network services disabled by default

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Mac OS X (2)

Security-Specific Features Built In

Anyone reasonably serious about computer and network security can appreciate the lineage Mac OS X comes from simply by reviewing some of the powerful security software that comes with it by default:

- Password assistant (10.3 and up): A tool that grades the strength of passwords as you create them. You select what qualities you want your password to include (case sensitive, punctuation, numbers, and so on), and it will color code your password choice based on its strength. Red is weak, yellow is just okay, and green is a strong password. It will even allow you to set the strength you're looking for and make suggestions. It is sometimes difficult to explain to users to choose a strong password, and this tool is a great guide for someone who has never done it before.
- FileVault (10.3 and up): Strong encryption for your home folder. Recent and repeated laptop thefts have brought to light the importance of hard disk encryption. FileVault is designed to encrypt the home folder of the user only, but it uses 128-bit Advanced Encryption Standard (AES), which is highly respected for its strength.[1]
- OpenSSH (10.1 and up): The open source version of Secure Shell (SSH), which is an encrypted remote access suite that replaces telnet and FTP.[2] There are a bunch of SSH clients available for a variety of operating systems, but the fact that OS X includes it by default gives you an idea what their overall security posture is.
- IPFW (10.2 and up): A personal firewall program installed and turned on by default.[3] Later versions have GUI frontends and are stateful, but there are handfuls of Linux desktop releases that either don't come with a firewall package or don't turn it on by default.

Finally, and again a good indicator of how serious Apple has taken security, a default OS X install has most network services disabled by default. Instead of installing all sorts of bells and whistles and allowing anyone to connect to them remotely, Apple decided to force users to enable services as they need them. This is a much safer approach than many operating systems.

Summary

OS X installs out of the box with some powerful security tools built in and turned on by default.

References

- [1] <http://www.macdevcenter.com/pub/a/mac/2003/12/19/filevault.html>
- [2] <http://www.securemac.com/macosxopenssh.php>
- [3] <http://www.macdevcenter.com/pub/a/mac/2005/03/15/firewall.html>

Linux Kernel

- Linus Torvalds originally wrote the Linux kernel in the early 1990s
- It and the source code were released to the public; it was "given" to the world via the GNU Project
- Other programmers from around the world joined in
- Even though he originally wrote it, today, Torvalds' code represents about only 2% of the kernel

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Linus Torvalds, Creator of the Linux Kernel

In the early 1990s, a Finnish software engineer named Linus Torvalds started a new hobby: rewriting the "Minix" OS (a Unix teaching tool) totally from scratch for the Intel 386 platform.[1] What he created was one of the first Unix-like operating systems to run on a personal computer. Just as interesting, Torvalds didn't want any money for it. He was willing to effectively give the source code for his idea to the world to let other people use, improve, and pass around to others that might be able to use it. This set the stage for something liberating in the computer world.

The GNU Project was already up and running as a repository and, just as important, a distributing point for such ideas. It was in this fertile soil that Torvalds' idea—called Linux—took root and was made available to anyone who was interested.

One of the great symbiotic relationships of the world came together. GNU gave Torvalds a vehicle for broadcasting his work to the world, and Torvalds gave GNU something for all their software to run on—a robust and completely open source kernel. The kernel acts as an intermediary between the hardware and the applications in an OS. The GNU Project adopted Torvalds' work and it became the last missing piece required to make GNU a complete and portable OS. This is what is commonly called Linux, although the collection is more properly referred to as GNU/Linux.[3]

Summary

Needless to say, the idea of Linux and the marriage of it to the already running GNU Project took off. Programmers from around the globe began to take notice and make contributions to Linux kernel. To give you an idea how many cooks worked on this soup, even though Linus Torvalds originally wrote the kernel by himself, today his code represents only about 2% of the kernel![4]

References

- [1] <http://www.linux.org/info/linus.html>
- [2] <http://www.gnu.org/gnu/linux-and-gnu.html>
- [3] <http://www.gnu.org/gnu/why-gnu-linux.html>
- [4] http://en.wikipedia.org/wiki/Linus_Torvalds

Ubuntu

- Based on Debian Linux
- Comes in Desktop and Server editions
- Language support: Installs in 40 different languages
- Support architectures: amd64, i386, UltraSPARC, and PowerPC
- APT-based package management
- No graphical firewall included

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Ubuntu

Ubuntu is one of the key Linux distributions.[1] The name comes from an African word, which roughly translated, means humanity to others. The project is sponsored by a South African company owned by Mark Shuttleworth who made billions with an Internet company and is now the driving/unifying force for Ubuntu Linux.[2]

You can download a desktop or server edition of Ubuntu and in "derivatives" that focus on education, open licensing, multimedia work, or with specific desktop environments (or window managers).[3] There are also localized editions with full language support in eight non-English languages. Speaking of languages, that's one of the areas where Ubuntu is a recognized leader in Linux distributions as they have 40 total languages to choose from when you run the installation.

Ubuntu aims to be a free replacement to Windows, so the packaging for the i386 platform is streamlined to make the installation as carefree as possible. There are alternate CDs available for the amd64, Sun UltraSPARC (server only), and older Apple PowerPCs.

Ubuntu is based on the venerable Debian distribution, which has been around since 1993 and is well known for its package management system, the Advanced Packaging Tool (APT). Debian has always strived for stability with willingness to avoid cutting edge software in lieu of dependability. Ubuntu inherits this stable environment and APT makes it straightforward to install software:

```
apt-get install <package name>
```

APT will go out and research the package in question, see what libraries and other programs that package might need (these are called dependencies), goes out to the Internet to download them, installs it all, presents any configuration choices to the user, and then checks its work; all of this is done automatically. Ubuntu comes out with a new release every six months, which is supported by security updates for 18 months and maintains a Long Term Support (LTS) release that will have security updates for 3 to 5 years.[1]

Some critics of Ubuntu do not believe its stance on security is very robust. Indeed, Ubuntu does not, by default, come with any firewall enabled and there isn't a graphical control for the firewall.[4] This is something to keep in mind if you try Ubuntu. Another critique of Ubuntu is that, for better or worse, it follows Debian's conservative attitude towards stability over cutting edge software. So if you're looking for the latest 3-D game or hot application, you might want to look at another distribution, such as Fedora Linux, which prides itself on having the latest software.

Summary

Ubuntu comes in server and desktop editions and a few derivatives for things like educational or multimedia work. Ubuntu leads the Linux desktop community in supporting non-English languages: 40 altogether. Ubuntu supports the major desktop architectures and uses the highly respected APT package management system, which installs software and any required dependencies automatically. Ubuntu is not particularly well known for security and its firewall is not configured by default.

References

- [1] <http://distrowatch.com/dwres.php?resource=major>
- [2] [http://en.wikipedia.org/wiki/Ubuntu_\(Linux_distribution\)](http://en.wikipedia.org/wiki/Ubuntu_(Linux_distribution))
- [3] <http://www.ubuntu.com/products/whatisubuntu/derivatives>
- [4] <https://wiki.ubuntu.com/UbuntuFirewall>

Fedora

- Based on Red Hat Linux
- Installer makes user choose: desktop, server, or other
- Support: i386, x86_64, PowerPC, alpha, and sparc
- 25 languages, a separate project for more translations
- RPM-based package management using yum
- Firewall included and enabled by default

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Fedora

Currently, Fedora is the third most popular distribution. The Fedora project is based on Red Hat Linux, which is one of the most profitable and largest Linux companies in the world.[1] Red Hat maintains a series of commercial products, most notably Red Hat Enterprise Linux. The fact that there is very much a for-profit side of Fedora is a sore point for some critics who feel that Fedora is simply the test bed for Red Hat's Enterprise release.

Unlike Ubuntu, which makes various editions available to the user to download in a pre-packaged form, Fedora users all download the same installation program, which then interviews the user on exactly what kind of install they want: desktop, server, or dedicated machine somewhere in between.

Fedora supports most of the architectures that Ubuntu does, including i386, PowerPC, sparc, and the additions of x86_64 and alphas (no amd64). The company also offers special releases dedicated to specific Windows managers (we discuss these later).

Fedora doesn't support the same number of non-English languages as Ubuntu, but they are up to 25 languages and have a separate translation project that will add more.

Red Hat developed the Red Hat Package Manager (RPM), which resolves dependencies on installation much like APT does for Debian and Ubuntu.

Fedora also makes use of a newer package manager called yum that works from the command line and has become popular in the last few years. Fedora comes out with a new release every six months and supports that release with security updates for 18 months and will possibly extend that support depending on the response of the user community.[3]

Fedora comes with a firewall that is enabled by default.[4] There is also a graphical configuration component that shows the user how the rules are created in an easy-to-understand format.

Summary

Fedora is based on Red Hat Linux and managed by Red Hat Inc., a for-profit Linux company. Rather than choose a version to download, Fedora users all download the same installer program, which then asks the user what kind of installation they want. Fedora supports popular hardware architectures and 25 non-English languages with more on the way. Fedora uses Red Hat's RPM package management, which resolves dependencies for the user automatically. There is a firewall included and enabled by default with Fedora.

References

- [1] <http://distrowatch.com/dwres.php?resource=major>
- [2] http://www.news.com/Red-Hat-releases-new-hobbyist-Linux/2100-7344_3-5103510.html
- [3] <http://fedoraproject.org/wiki/Legacy/FAQ>
- [4] <http://tinyurl.com/35mxm9>

Linux Live CDs

- Typically runs off of a CD or DVD
 - ISO can also be run within a virtual machine
- Allows you to "test drive" a distribution
- Pre-installed tools
- No hard drive required!
- Excellent for forensic work
- Helpful for troubleshooting

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Linux Live CDs

Introduction

It used to be that trying out Linux meant you had to find a spare hard drive and system you weren't using, find the right documentation, and then set aside a few hours. That's to try one distribution. If you wanted to try out another distribution, you needed to find another free afternoon. You also couldn't compare and contrast different distributions, unless you had several spare PCs around to dedicate to the cause. This effectively left a lot of the potentially-curious-about-Linux crowd out.[1]

Live CD

Live CDs changed all that. Now in a single afternoon with a stack of blank CDs you can check out a dozen Linux distributions and not make a single modification to your hard drive. Heck, you don't even need a hard drive! Do a Google search for "Linux Live CD" and you'll find literally hundreds of CD and DVD image files that you can download (usually in .iso format) and burn to a CD. Once you pick one and burn it, you end up with a bootable CD that is smart enough to figure out the hardware on your PC, and in most cases, you can boot directly into the same GUI you'd get if you installed everything permanently on your hard drive, except it isn't permanent! Reboot the machine and it will come back up in the native OS on the hard drive as if nothing ever happened. Some of these Live CDs allow you to evaluate a full Linux distribution, some are for rescue and recovery, some have nothing but hardware diagnostics, some are set up for no other reason than to showcase a particular application.[2]

Live Distro

If you do another Google search for LiveDistro, you'll quickly discover that the same concept applies to any bootable media: USB Flash drives, floppies, external hard drives, even iPods! The flexibility this presents is great for some folks. Imagine you're travelling and need access to some of your Linux applications. You can borrow a friend or co-worker's computer, boot a Live CD into Linux, do your work, and when you return, the computer will act as if you never used it.[3]

Summary

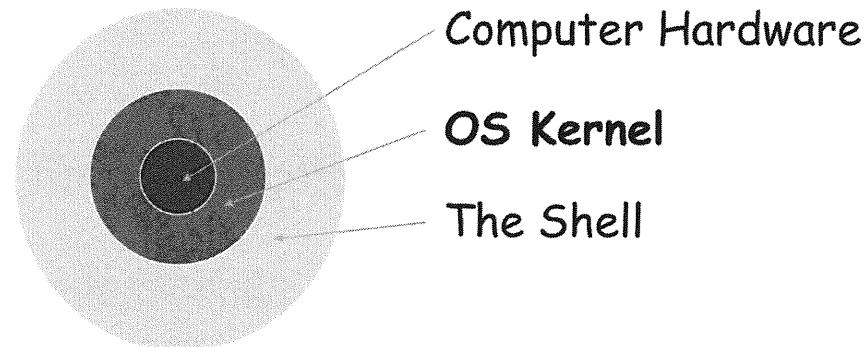
Live CDs are also an excellent tool to use when investigating a computer's installation, without taking the risk of modifying anything. As mentioned earlier, if you need to make a forensic copy of a hard disk, a Live CD is an excellent tool to use. You can also troubleshoot possible hardware problems with a Live Distro. You may run into a Windows computer with a problem connecting to the Internet. If you can get online with a Live CD on that machine, you know it's a software problem—if not, then maybe it's a hardware problem.

References

- [1] <http://www.frozentech.com/content/livecd.php>
- [2] <http://en.wikipedia.org/wiki/LiveDistro>
- [3] <http://distrowatch.com/dwres.php?resource=cd>

Operating System Overview

- The memory resident part of an operating system that directly interfaces with the hardware is called the kernel



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Operating System Overview

Any operating system overview begins with a 50,000-foot view of the key elements. The three key elements of Unix are:

The kernel: A memory resident part of the operating system

The shell: The portion of the operating system with which users and process interact directly

The hardware: A collection of components that house data and provide the means to communicate

This sounds minuscule, but understanding how these components interact to each other allows us to understand where the levels of protection efforts can be directed. For example, say I want to prevent an elevation of privileges action, I can focus my protection efforts towards the interaction between the shell and the kernel, keeping a process or user within its respective realm of operation.

Kernel

- The kernel manages the hardware and the executing processes
- Kernel services include:
 - The filesystem
 - Low-level network protocol support (for example, IP)
 - Memory and process management

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Kernel

The kernel is the most important part of the Unix operating system or any operating system for that matter. Without the kernel, operating systems do not function. The kernel is responsible for order, instructions to all (both hardware and software), and for complete interaction of all system elements. Destroy the kernel and the system becomes useless. Destroy a user or application and only limited functions are affected, yet the rest of the operating system functions.

The kernel of any operating system is generally loaded into memory at boot up. This allows faster interaction between all components, and yet, must have a dedicated space in memory to reside without shifting. What I mean by this is that it is when loaded, it is normally the first thing loaded and may be predictable. Hackers and software developers alike know. Here is where a common “line in the sand” is drawn and where focus on software development security is targeted.

Logical File System

/ (root file system; top of directory hierarchy)
/dev, /devices (directory containing files used to talk to system devices)
/usr (primary OS directory; read-only)
/var (contains log files, queues, and so on)
/bin, /usr/bin, /usr/local, /opt (executable programs; some SUID/SGID)
/home, /export/home (user home directories)

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

There used to be a lot of variance in where different files were located in different Unix "flavors," but modern Unix variants have settled down to using the same general file system layout. The top of the file system tree is the root directory, /. Below this directory are various important subdirectory trees.

The /dev directory contains the special device files that programs running on the system use to communicate with the physical hardware devices controlled by the operating system kernel. Unix systems derived from the AT&T "System V" (SYSV) Unix standard, such as Solaris and HP-UX, often will put these device files into the /devices directory but also usually maintain a /dev directory for compatibility with other systems.

/usr is where most of the critical components of the operating system live, including system binaries, programming libraries and tools, and online documentation. This directory structure can be thought of as being read-only after the operating system is loaded; not many changes under /usr should occur unless the operating system is upgraded or patches are installed. /var is where the system keeps frequently changing data, such as log files and temporary queues for system services such as e-mail and printing.

Although programs provided with the operating system end up in directories like /usr/bin and /usr/sbin, other programs can be found scattered throughout the system. A standard convention is to put third-party software obtained from the Internet into the /usr/local directory. SYSV-derived systems like Solaris and HP-UX often will put third-party software (particularly commercial software) into /opt. Different sites may choose to use a different directory naming scheme for third-party software, however, such as /pkg or /sw.

User home directories often are found under /home. For large Unix networks, though, /home often is an NFS mounted directory, and /export/home is the directory in which the files physically reside on the file server. Again, many other local conventions exist. Some sites prefer directory names like /users or /u1, /u2, ..., etc.

... and the Physical File System

- The logical Unix file system is made up of multiple physical disk partitions
- Disk partitions are mounted at various points in the file system
- Different security options can be set on each mount point

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Although the Unix file system appears to be a single logical entity to the users on the system, it actually is made up of several pieces (called partitions) that correspond to physical sections of the machine's disk drives. Partitions generally are assigned to critical pieces of the logical directory structure. For example, the system might have one partition for the root file system, another for /usr, another for /var, and so on. During the boot process, all of these partitions are mounted into their proper place to make the file system appear contiguous.

Disk partitions, their mount points, and the options applied to each partition generally are described in a file called /etc/fstab. Solaris uses the name /etc/vfstab apparently just to be perverse. These file system options in the /etc/fstab file allow the administrator to specify different security settings for various partitions.

From a Unix perspective, disk drives have 16 different partition "slices," either numbered 0-15 or lettered a-p. For example, Linux would use the device named hda7 to specify slice 7 (which technically is the 8th slice since the numbering starts at 0 of the first (disk a) hard disk (hd) on the system. Note that older Unix systems such as Solaris allow only eight physical partitions per drive.

One slice generally is reserved as the "overlap" partition, which represents the entire disk geometry. By convention, this usually is the third slice of the disk because early Unix systems required the first slice to be the root file system and the second slice to be the swap space for the virtual memory system. This "overlap" slice should not be used for file systems. Note that not all possible disk partitions need be used. In fact, some administrators just create a single partition that spans an entire disk and ignores the other slices. It turns out that this is a bad idea for several reasons.

First, partitions make it more difficult for problems with one subsystem, such as processes logging data in /var, to cause a denial-of-service to other processes writing data in a different partition, such as programs writing scratch files in /tmp. Partitions are created at system installation time and with a fixed amount of space. If syslogd were to fill up /var with spurious log messages, then no more logging could be done until space was freed up by deleting old log files or removing other data from the /var partition. Not being able to log new messages is bad, but not as bad as the entire system becoming unusable because the logs consumed all available disk space.

A second reason for partitioning is to make backups easier. Some partitions like /usr rarely need to get backed up because they change so infrequently. On the other hand, user home directories might need to get backed up every night.

Third, and most importantly from a security perspective, splitting the Unix file system up into different partitions allows the administrator to set different security options on different parts of the file system. Although the exact partitioning scheme for a machine will vary from system to system and site to site, it is important that the root file system /usr and /var be separate partitions in order to apply appropriate security measures to various OS directories. Non-OS data (user data, home directories, and application data) and third-party applications not supplied by the OS vendor generally should be put into their own partitions so that they do not "pollute" the OS directories.

The df Command

	Partition Total Size		Available Space		Mount Point
\$ df					
Filesystem	1k-blocks	Used	Available	Use%	Mounted on
/dev/hda5	3099260	2383084	558744	82%	/
/dev/hda1	38859	3485	33368	10%	/boot
/dev/hda7	5486908	20	5208164	1%	/home

Disk Device Space Used Percent Full

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Administrators can display the currently mounted partitions with the df command. df stands for disk free, because the command actually was created to help administrators find partitions that were running out of disk space.

# df					
Filesystem	1k-blocks	Used	Available	Use%	Mounted on
/dev/hda1	256667	108443	134972	45%	/
/dev/hda6	3201980	593908	2445416	20%	
	/home				
/dev/hda	33099292	2665948	275908	91%	/usr
/dev/hda5	2063504	104048	185463	66%	/var

The first output column shows the disk device associated with the partition, and the last column shows the piece of the logical file system to which the partition corresponds.

The middle columns in the df output show the total size of the partition, the amount of disk space currently used, and the amount of free disk space, all reported in 1 kilobyte chunks. The attentive reader will note that the bytes used figure plus the bytes available amount is not equal to the total size number. For performance reasons, standard Unix file systems typically will reserve 5-10% of the total file system for free space. When a partition gets to 90% full, no user except root is allowed to write data into the file system. Note that the percent of capacity figure in the second-to-last column of the df output takes into account this reserved space. Thus, df shows 100% full when the partition actually is only 90-95% full.

Note that df shows only currently mounted file systems. Typically, the machine also will have a partition devoted to "swap space" for the virtual memory system. This is a "raw disk" partition that is not mounted into the logical Unix file system. However, this swap partition will have an entry in the /etc/fstab file. Information about swap partitions currently in use can also be obtained using the swapon -s command.

File System Security Goals

- Protect OS binaries in /usr.
- Prevent introduction of SUID programs and unauthorized devices
- Allow other software to be installed
- Discourage denial-of-service attacks

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

When thinking about file system security, it is useful to keep a few important goals in mind:

An attacker who compromises a system likely wants to install a root kit, a set of binaries that gives the attacker a back-door into the system and helps her escape detection by the system administrator. Typically, the binaries that the attacker replaces are OS programs in the /usr/bin and /usr/sbin directories, so protecting the /usr file system is important.

The administrator also should attempt to stop people from creating or bringing unauthorized set-UID and set-GID programs on the machine. Unauthorized device files can be equally dangerous because they may allow normal users to get what normally is privileged access to disk drives, the system memory, etc., bypassing normal system access controls.

Conversely, system administrators need to be able to apply patches to the operating system and update software that has been installed on the machine. If they are unable to do this, then the system gradually becomes less secure as new exploits that cannot be patched are discovered.

Also, to avoid denial-of-service attacks, administrators should partition the system carefully. For one thing, this means splitting the file system appropriately; take the earlier example of making /var a separate partition so that overwhelming the system logs will not take out the entire machine. However, the administrator also should take care to provide sufficient free space in heavily used partitions to accommodate unexpected growth.

File System Security Options

ro: File system is mounted read-only (files and directories cannot be modified)

nosuid: SUID/SGID bits are ignored on all programs in the file system

nodev: Unix device files don't work

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

A combination of sensible partitioning and appropriate use of file system mount options in the /etc/fstab file can help achieve a secure installation. While there are a lot of different mount options administrators can set on a given file system, there are just a few specific options required to protect the security of file systems on the machine:

The ro (read-only) option causes the Unix operating system kernel to prevent writes or updates to the given file system. When a file system is in read-only mode, nobody can update files or directories in that partition, add new files, or delete files.

The nosuid option means that the operating system simply ignores the set-UID and set-GID bits on executables in the file system. So, if a user attempts to execute a set-UID program out of a nosuid file system, the program runs as the user and not as the owner of the program.

Similarly, nodev means that special "device files" are ignored in the file system. These are the kinds of files usually found in /dev and /devices and are used to communicate with the system hardware devices. Device files appearing in other directories usually are a problem, the only exception being special file systems used by anonymous FTP servers and the like.

Goal of Security

All file systems should either be mounted read-only or nosuid.

- /usr and /usr/local contain SUID/SGID programs, but can be read-only
- Most other file systems must be writable, but have no SUID/SGID programs
- / file system contains /dev, but all other file systems can be mounted nodev

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

The basic rule to follow when applying file system options is simple: file systems should either be mounted nosuid or read-only. Additionally, the nodev option should be used wherever possible. Set-UID/set-GID executables and system device files should really be restricted to operating system directories that are tightly controlled by the system administrator.

On most Unix systems, the critical set-UID and set-GID programs are all located in directories under the /usr file system, with /usr/bin and /usr/sbin being the most common locations. Some additional third-party set-UID and set-GID programs also may show up in /usr/local or /opt. However, all of these directories usually can be mounted in read-only mode because data usually is not written into these directories; Unix programs tend to write data under the /var directory or in scratch directories like /tmp. Mounting these file systems read-only helps stop attackers from planting root kits into the file system and stops the introduction of rogue set-UID or set-GID files and system devices.

Other file systems typically need to allow write access but should not have any set-UID and set-GID programs showing up in them. Thus, these file systems can be mounted nosuid to again prevent the introduction of rogue set-UID and set-GID executables.

As far as system device files go, the system devices live in /dev and /devices, which always are parts of the root file system. It turns out that if the primary system devices are not found in the root file system, the machine usually cannot boot. It is unusual to have device files elsewhere in the file system, so pretty much every other partition except the root file system can be mounted nodev. This means that even if attackers were able to create device files in these file systems, they would not be able to use them. Of course an attacker could still create a malicious device file in the root file system or change the permissions on a device file already in /dev or /devices, so this is not a perfect solution. However, it is a huge improvement over the default situation, which is no security options being set at all.

Shell

- The shell is the command-line interpreter that a person uses to run programs on the computer
- Provides the user with an interface to the system:
 - The shell listens to the terminal
 - It translates requests into action by the kernel and programs

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Shell

The shell is the portion of the operating system with which the user has direct interaction. It can also be the vehicle through which applications obtain permissions and interact with the kernel. As you can see, the shell is like a car. To get from point A to point B, one must get into the car, and if all goes well and functions properly, one gets to the desired outcome or destination. However, without the desired skills, capabilities, or permissions, we cannot achieve our goal.

Examples of Shells

- For Unix:

- Bourne Shell (sh)
- C Shell (csh)
- Bourne-Again Shell (bash)
- Korn Shell (ksh)
- exTended C Shell (tcsh)

- For DOS:

- COMMAND.COM

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Examples of Shells

This slide depicts some of the most common shells used in Unix environments, and it identifies or demonstrates the equivalent component in Windows.

Important

Windows Versus Unix

<u>DOS</u>	<u>Unix</u>	
• Dir/w	• ls	Lists contents of directory
• Dir	• ls -l	Long listing; shows attributes
• Dir /ah	• ls -al	Lists hidden and regular files
• cd	• cd	Changes the working directory
• rename	• mv	Renames files
• attrib	• chmod	Changes file attributes
• md	• mkdir	Makes a new directory
• rd	• rmdir	Removes a directory
• del	• rm	Deletes files
• copy	• cp	Copies files

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Windows Versus Unix

This slide provides a correlation between DOS and Unix commands for those who may already have familiarity with DOS command-line functions.

Commands You Need to Know

- cat (concatenate) To view files
- pwd Print working directories
- more To view a page at a time
- man To read the help manual
- find To find specific files
- grep To perform string searches
- su Switch user accounts
- Piping using the | character
- Creating output files using >
- (See Appendix for the usage of each command)

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Commands You Need to Know

This slide contains some of the key Unix commands. Some of the more predominant commands in analyzing incidents and locating items are the grep and find commands. These are demonstrated later.

Summary

- You can choose many variants of Unix.
- Carefully designing the installation of an operating system with proper partitions can increase overall security
- Key commands can be helpful in working with a system:
 - ps for running process
 - su for switching between user accounts

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

In the first module, we looked at the various types of Unix systems available and how they can be used in an organization. We also looked at the components of a system and how through careful design, you can increase the overall security by mounting partitions with different security options. The previous section focused on the key commands that you need to understand to work in Unix.

Module 30: Securing Linux/Unix

PERMISSIONS AND USER ACCOUNTS

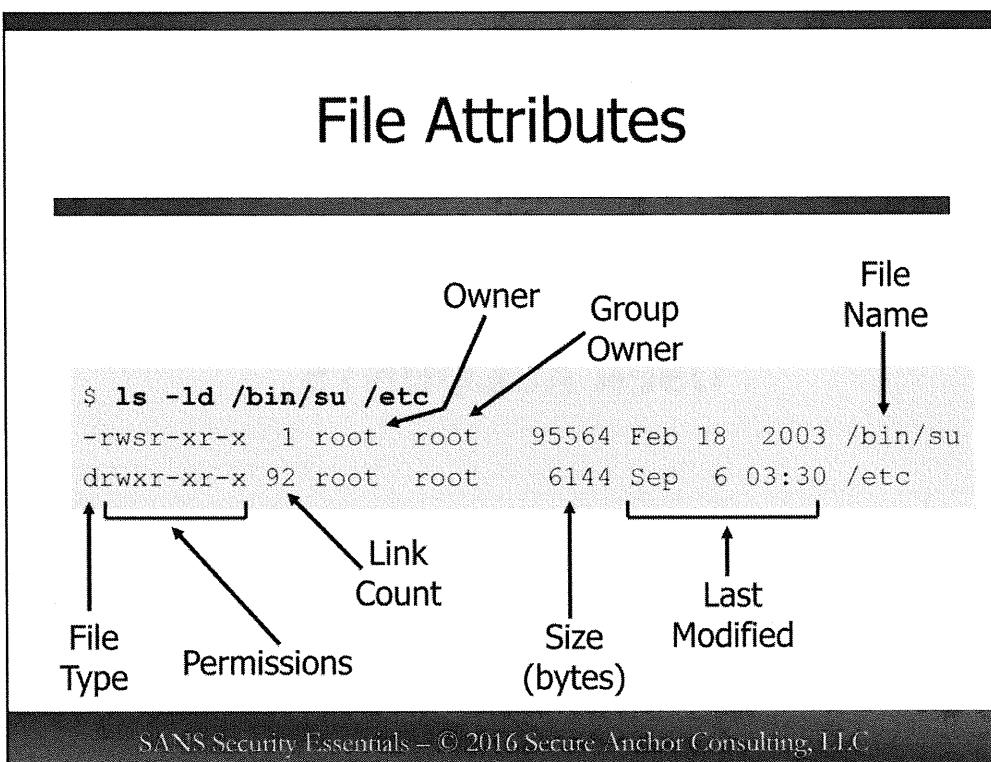
SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Module 30: Securing Linux/Unix

PERMISSIONS AND USER ACCOUNTS

This section intentionally left blank.

File Attributes



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

The Unix operating system stores a number of different attributes for files and directories, including the file's owner and group owner and the access control settings on the file.

The easiest way to view the attributes for a file is with the `ls -l` command, which gives a detailed file listing:

```
$ ls -ld /bin/su /etc
-rwsr-xr-x 1 root root 95564 Feb 18 2003 /bin/su
drwxr-xr-x 92 root root 6144 Sep 6 03:30 /etc
```

Note that the `-d` option forces the `ls` command to display the file attributes for the actual directories listed—in this case, `/etc`.—rather than showing the attributes for all of the files in that directory, which is the default.

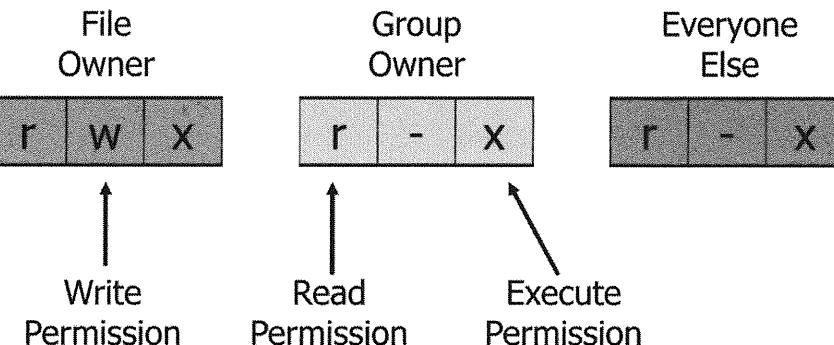
The first column of the listing is a block of letter codes showing the type of file and the access permissions set on the file. The letter in the first column of this block shows which type of object this is. The symbol `-` means a regular file, and `d` indicates a directory. There are several other letter codes defined for other special types of files, but they are not important for the current discussion. The other nine columns in this initial block of letter codes describe the access permissions on the file, which are our concern for most of the rest of this section.

The next column in the file listing is the link count field. Again, understanding the meaning of the link count field is not critical for this discussion.

The next two columns show the owner of the file and the group owner of the file. These combine with the file access permissions to tell the system who has access to the file.

The fifth column is the file size in bytes. Next comes the last modified time on the file. Note that dates within the last year display the actual time of modification, whereas older dates simply display month, day, and year. This is a human readable formatting decision; actual Unix dates are stored in a completely different internal format. The last column displays the file name.

Unix File Permissions

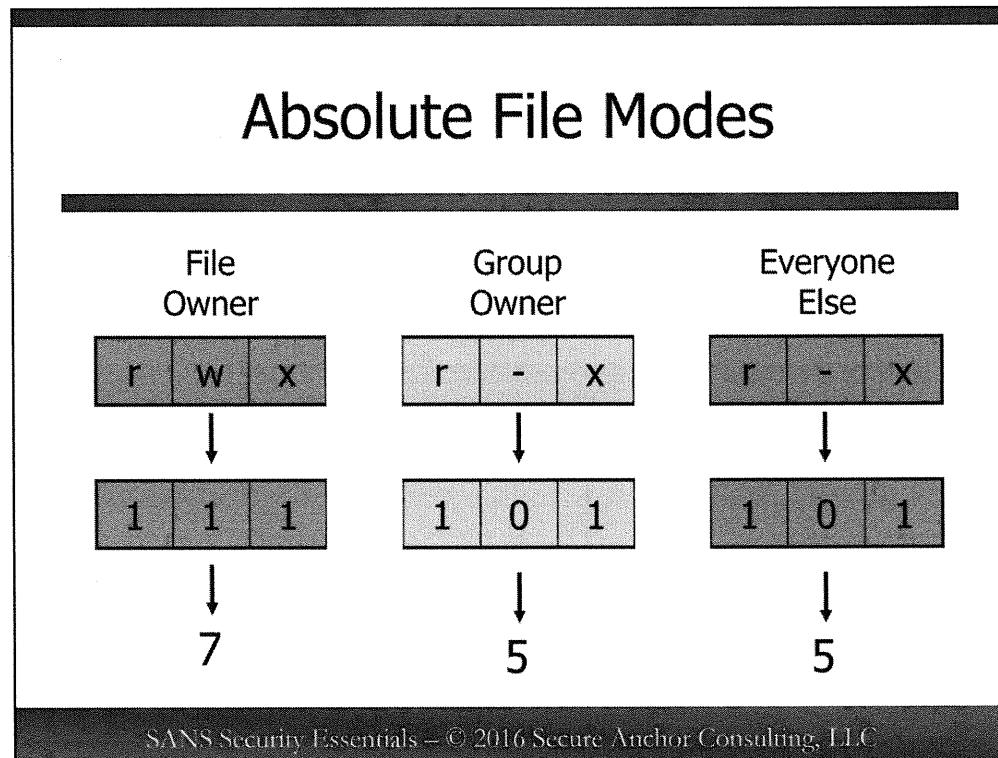


SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Let's focus our attention on the permission settings in the first column of the ls -l output. Unix file permissions use a security model with fairly coarse granularity. Only three basic permissions—read, write, and execute—can be set. Read permissions (r) give the ability to look at or view the contents of a given file and also the ability to make a copy of that file elsewhere on the system. Write permission (w) is the ability to modify the contents of the file. Execute (x) is the ability to run that file as a program. Note that a program file may have execute permissions set without allowing read permissions. Users will be able to execute the program code stored in the file but not read the contents of the file to make a copy of that program elsewhere on disk.

Read, write, and execute permissions can be set for three different categories of people: the file's owner, people belonging to the Unix group listed as the group owner of the file, and everybody else (or other). The first three permission flags shown by ls -l (after the initial letter that indicates the type of file) are the permissions that are set for the owner of the file. The next group of settings in the middle three columns applies to the group owner of the file. The final three permission flags apply to the everybody else category. Note that if a particular permission flag is not set, then a dash (-) is displayed.

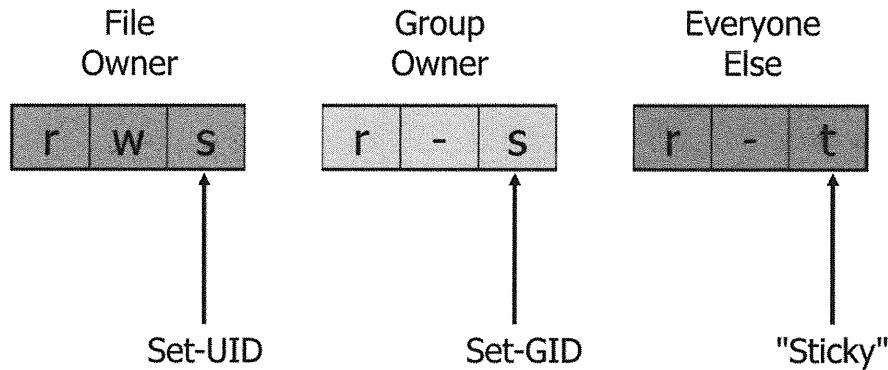
R=4 Suid
W=2 SGid
X=1 Sticky



Although the ls command displays permission settings using the letters r, w, x, and so on, the operating system actually stores these settings in a completely different internal format. Unix access permission flags often are referred to as bits because their representation in the Unix operating system as binary digits. A one means the given flag is turned on, and a zero means the flag is not set. Because there are three bits to be set for each ownership group—read, write, and execute—these permissions usually are represented in octal notation.

Consider the example in the illustration. The owner of the file has read, write, and execute permission. All bits are turned on, so the binary representation is 111, which is octal 7 (4+2+1). The group owner and other category only have read and execute set but not write permissions. The binary representation is 101, which is 5 (4+1) in octal notation.

Other Permission Bits



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

There are three other permission flags that optionally may be set on a file: set-UID, set-GID, and sticky. Because there is no room to display these extra three bits in the ls -l output, the ls command shows these settings by replacing the Xes in the normal output. If set-UID is set, then an s is displayed instead of the x for the file's owner. Similarly, set-GID is represented by showing an s instead of an x for the group owner category. The sticky flag is represented as a t in the very last column of the permissions flags, hiding the x for the everybody else category.

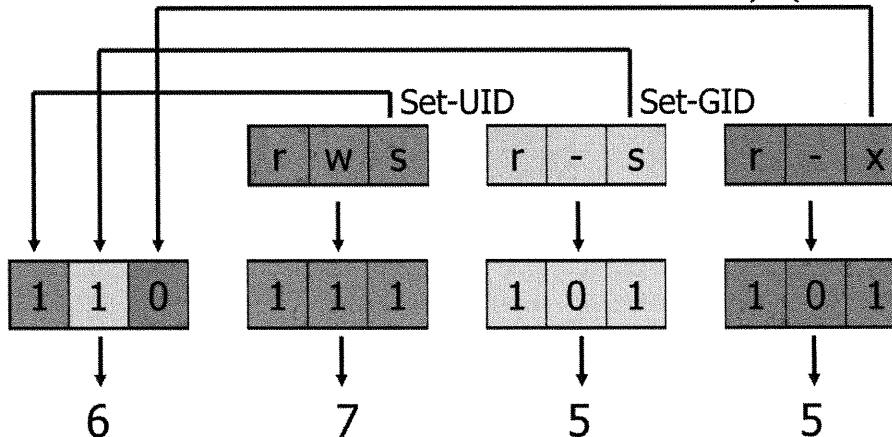
Set-UID and set-GID are interesting innovations that were created by the original Unix developers. Certain programs need to run with special access privileges not available to normal users. For example, the passwd command that allows users to change their passwords needs root privileges so it can update the /etc/shadow file. The set user ID flag (usually shortened to set-UID or just suid) causes a program to run as the owner of the executable, rather than as the user that executed the program. Thus, the passwd program is set-UID and owned by root so that it can update the shadow file when executed by a normal user.

Set-GID (set group ID) functions the same way, but gives only the user who executes the program additional group access rights. The Unix commands that are used to print files often are set-GID to a special line printer (lp) group that is allowed to copy files into the system's printer queue directories.

The so-called sticky flag originally was developed in the early days of Unix on slower machines. The idea was that any program which had the sticky bit set was supposed to "stick around" in the memory of the operating system after the program had finished executing. This was a win on programs that needed to be executed frequently because they did not need to be read back into memory constantly. Modern Unix systems, which use shared libraries, have better caching algorithms, and run on faster hardware, generally ignore the sticky bit on executables. However, the sticky bit now has a different special meaning when applied to directories in which the owner is the only person who can delete files in that directory.

Other Bits in Absolute Mode

"Sticky" (not set)



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

How are the set-UID, set-GID, and sticky bits represented? A fourth octal digit representing these values can be put in front of the three octal digits used to represent the read/write/execute permissions for the various ownership categories. In this leading octal digit, set-UID is the leading most significant binary digit (4 octal); set-GID is the next bit (2 octal); and sticky is the final least significant bit (1 octal).

In the illustration above, both the set-UID and set-GID bits are set, but the sticky bit is not. The octal representation of the leading digit is therefore 6 (4+2). In addition, read/write/execute is set for the file owner and read/execute is set for the group owner and everybody else categories, just as in the previous example. Thus, the complete numeric representation for this file's mode is 6755.

Because this octal notation completely specifies all of the permission bits on a file, it generally is referred to as the absolute mode of the file. The r/w/x notation displayed by ls is usually referred to as the symbolic mode of the file. Unix commands that deal with file permissions are usually able to handle either absolute or symbolic mode.

Files Versus Directories

	File	Directory
Read	can read file contents	can get directory listing
Write	can modify file contents	can create/remove files
Execute	may execute file	may access files in directory
Set-UID	file executes with privileges of file's owner	N/A
Set-GID	as above but for group owner	group ownership of new files is inherited
"Sticky"	N/A	only owner may remove files

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Because the Unix file permissions model is so limited, the meaning of each of the permission flags we have seen so far is "overloaded." In other words, the permission flags have different behaviors depending on whether or not they are being applied to files or directories.

For example, being able to "read" a directory means that the user can run the ls command (or other similar commands) to get a listing of the files in the directory. The execute bit on a directory gives the user the ability to change directories into the given directory and to access files out of that directory, but a directory listing cannot be obtained unless the read permissions are also set. Generally speaking, directories need the execute flag turned on to be useful, but sometimes it is appropriate to give execute rights on a directory but not read privileges, such as for the everybody else access class. This would allow the owner of the directory to create files in the directory for various users on the system. A user would be able to access her file if the owner of the directory told her the explicit name of the file but would not be able to get a directory listing to see the other user's files.

Write permissions on a directory give the ability to change elements of the directory file. In Unix, directories store file names and pointers to the file contents. So, being able to modify a directory means being able to rename files as well as to add files to the directory and delete files from the directory altogether. This often is confusing to users of other operating systems where create, delete, and rename privileges are implemented as separate attributes on individual files.

Set-GID has a surprising meaning on directories, as well. If the set-GID bit is set on a directory, and a user creates a new file in that directory, then the group owner of the new file will be the group owner of the directory rather than the primary group to which the user belongs (which would be the normal default behavior). This is useful when using Unix groups as a mechanism for sharing files in a large project because it ensures that new files created in the project tree end up being owned by the special group created for the project.

When the sticky bit is set on a directory, then only the owner of a given file may remove that file from the directory. This bit often is set on /tmp and other world-writable directories to prevent users from stomping on each other's files (either accidentally or purposely). These so-called world-writable directories are examples of potential security issues of which administrators should be aware on their systems.

Look Out: World-Writable Directories

- World-writable directories such as /tmp are used by programs to hold intermediate results
- What if an attacker clobbers your program's temporary file and substitutes his own?
- Golden rules:
 - Avoid world-writable directories if possible
 - Always set the sticky bit for world-writable directories

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Look Out: World-Writable Dirs

Potentially "Dangerous" Permissions

Unix systems typically have a number of directories that are world-writable, meaning that write permissions are enabled for the everybody else category. In fact, world-writable directories generally are configured so that all permissions (read, write, and execute) typically are turned on for all three ownership categories. This means that anybody on the system can create, rename, and delete files at will in these directories. An example of such a directory would be the standard Unix /tmp and /var/tmp directories, where programs are supposed to write "scratch" data and intermediate results.

Historically, though, world-writable directories have caused enormous security problems. A simple example is the Unix C compiler used to compile programs from source. This compiler usually makes several "passes," in which the source code is transformed from one form to another before being turned into an actual executable. The results from each compiler "pass" are put into temporary files in /tmp. Now, imagine an attacker quickly substituting a file containing malicious code for one of the files created during a compiler pass—suddenly there is a virus or Trojan horse in the program!

Therefore, it is extremely important to ensure that the sticky bit is set on all world-writable directories, as is the default for system directories, such as /tmp and /var/tmp. With the sticky bit set, our attacker would not be able to delete or rename a user's intermediate compiler files to substitute the attacker's Trojan horse code.

Many programs these days simply avoid directories like /tmp altogether and instead are starting to use their own private temporary directories to which normal users on the system do not have write permissions. For example, the Sendmail e-mail server writes temporary data files containing messages that are currently being processed into a private directory called /var/spool/mqueue that is accessible only by the root user.

Look Out: SUID/SGID Programs

- Double-edged sword:
 - You can't run Unix without SUID/SGID programs
 - However, rogue SUID/SGID programs can easily compromise a machine
- Keep track of the SUID/SGID programs provided with your operating system
- Raise an alarm if new or unexpected SUID/SGID programs appear

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Set-UID and set-GID programs historically have been one of the most common weaknesses that attackers have exploited to get privileged access on Unix systems. For example, an attacker may be able to bring a set-UID root copy of the Unix command shell onto the system on a CD-ROM or USB drive. When the user executes this shell, she instantly gets root privileges without providing a password. Set-UID programs with buffer overflows or other coding errors may allow attackers to "escape" from the running program and get an interactive shell.

It may be a good idea to make a list of all of the set-UID and set-GID programs on the system when the machine is installed for the first time. Then, periodically scan the system for set-UID and set-GID programs and compare the results against the original list. If "new" set-UID or set-GID programs appear, be suspicious.

All of this, of course, begs the question, "How can administrators scan their file systems for set-UID and set-GID files in the first place?" It also would be nice to be able to locate world-writable directories to confirm that their sticky bit is set.

find Your Way Around

Find recently modified files:

```
find /etc -mtime -1 -print
```

Find all subdirectories:

```
find /dev -type d -print
```

Run a command on all files:

```
find /dev -type d -exec ls -ld {} \;
```

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Locating any object in the Unix file system typically is done with the find command. The find command has a syntax that is rather different from most standard Unix commands, so let's first cover some of the basics.

The arguments to the find command generally are divided into three groups: `find <directory> <qualifiers> <action>`. Here `<directory>` is the path where find should begin working—use a period (.) to start the search from the current directory. `<action>` usually is either `-print`, which simply prints the names of matching files, or `-exec` followed by some other Unix command.

There are many, many `<qualifiers>` that can be used to match different sorts of files.

Here are some simple examples:

```
find /etc -mtime -1 -print  
find /dev -type d -print
```

The first example would print the names of all files under the `/etc` directory tree and whose last modified time (`-mtime`) is less than one day old (`-1`); in other words, all files that have been changed in the last 24 hours. `-mtime +7` would show all files whose last modified time is greater than seven days ago; that is, files which have not been changed in the last week. The `-type` qualifier matches files of a particular type: `d` for directories, `f` for regular files, and so on. Attackers love to hide their tools in directories under `/dev` because the huge number of regular files in this directory tends to "hide" new files and directories from the notice of system administrators.

The `-exec` action has a rather strange looking syntax associated with it:

```
find /dev -type d -exec ls -ld {} \;
```

The `-exec` is followed by a normal Unix command line, but it has a couple of notable exceptions. First, use `{}` instead of the normal filename argument to the command that find should execute. Every time find discovers a file or directory that matches the search criteria, it executes the command specified with `-exec`, except that it substitutes the name of the matching file where the `{}` characters are. The find command also requires that the command line after `-exec` be terminated with `\;`. This allows the find command to know when the arguments to `-exec` are done, in case after the `-exec` there are other find options that need to get parsed.

File Modes + find

Show world-writable directories:

```
find / -type d -perm -0002 -ls
```

Find SUID/SGID files:

```
find / -type f \
  \(-perm -04000 -o -perm -02000 \) \
  -print
```

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Now, as far as the current task is concerned, the find command has a -perm option for locating files that have certain file permission flags set. -perm understands both absolute and symbolic file modes.

Using symbolic mode, here is a find command for locating world-writable objects throughout the entire file system, starting from the top or root of the file system, the directory /:

```
find / -perm o=w -print
```

Generally, world-writable means that write permissions are enabled for the everybody else ownership category. o=w is how this is represented typically in symbolic mode; the o stands for other. u= is used to specify permissions for the owner (user) of the file, and g= is used to specify permissions for the group owner.

The dash before the o=w means that find should display any objects which have *at least* the write bit set for the other category but which also may have other permission bits set. Without the dash, the find command would only report objects that *exactly* match the permissions specified. It is very unlikely that anything in the file system would have only the write bit set for other and no other flags turned on.

Now this same command could be written using absolute file modes:

```
find / -perm -0002 -print
```

Again, the dash before the octal permission value means match at least these flags, as opposed to match exactly.

The two find commands shown above will display any world-writable file, directory, or other object encountered in the file system. This probably is a good idea because even world-writable files should be discouraged.

Users do not want anybody on the system to be able to modify their files, and system configuration and log files should never be world-writable.

However, suppose the administrator wished to find only world-writable directories, perhaps in order to find ones that did not have the sticky bit set. This is accomplished easily by combining the -type d qualifier with either form of the -perm option:

```
find / -type d -perm -o=w -print  
find / -type d -perm -0002 print
```

The find command does the expected thing here and shows only objects that match both criteria (a logical and).

More complicated logical expressions can be built up using and (-a), or (-o), not (!), and parentheses. For example, it is possible to write a find command that shows only world-writable directories that do not have the sticky bit set:

```
find / -type d \(-perm -o=w -a ! -perm -o=t\) -print  
find / -type d \(-perm -0002 -a ! -perm -1000\) -print
```

Parentheses and the ! character are interpreted as special characters by the Unix command shell. Putting a backslash (\) in front of these characters prevents the Unix shell from evaluating these special characters and instead causes them to be passed into the find command for parsing.

A similar sort of logical grouping can be used to locate files that have either the set-UID or the set-GID bit set:

```
find / -type f \(-perm -u=s -o -perm -g=s\) -ls  
find / -type f \(-perm -4000 -o -perm -2000\) -ls
```

-ls is a special action that's similar to using -exec ls -l {} \;

This certainly is one way the administrator could generate a list of set-UID and set-GID files currently installed on the system. Comparing lists generated with the -ls action will tell the administrator not only if new set-UID or set-GID files have been added to the system, but also will alert the admin if file attributes on previously installed set-UID and set-GID programs have changed (such as the file size or last modified time).

Set Perms with chmod

- Can use symbolic file modes:

```
chmod u-s myfile          # turn off SUID bit  
chmod g+w myfile          # give group write  
chmod o-r myfile          # no read for "others"  
chmod a-x myfile          # turn off execute
```

- Can also use absolute file modes:

```
chmod 766 myfile          # file is "world write"  
chmod 600 /etc/lilo.conf   # only owner perms set  
chmod 1777 /tmp           # set "sticky bit"
```

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Of course, being able to simply locate problematic files and directories is not enough. Administrators also need to be able to change the permissions on files and directories, such as adding the sticky bit to a world-writable directory or removing the set-UID bit from an executable. The Unix chmod (change mode) command is used to set permission flags on files and directories. Like the -perm option for the find command, chmod supports both absolute and symbolic mode.

Symbolic mode is most useful for setting or unsetting specific individual permission flags on a given file:

```
chmod u-s myfile          # turn off SUID bit  
chmod g+w myfile          # give group write  
chmod o-r myfile          # no read for "others"  
chmod a-x myfile          # no execute for all
```

chmod uses the same letters for identifying the ownership categories that find does: u (user) for the file owner permissions, g for the group owner, and o for the other or everybody else category. chmod also recognizes a (all), which means apply the permission setting to all categories. Actually, a is the default, so chmod a-x myfile and chmod -x myfile do the same thing on most Unix systems. The ownership category is followed by a plus or a minus, indicating whether the given flag should be enabled or disabled.

In fact, multiple ownership categories and multiple permission flags can be specified simultaneously. Even commas can be used to specify a list of flags to be set and unset. For example, the command chmod u+s,go-r myfile would turn on the set-UID bit on the executable myfile and simultaneously remove the read bit for both the group owner and other categories. It is not uncommon to disable read privileges on set-UID binaries because on some Unix systems, this may be the only way to prevent users from executing set-UID binaries in a symbolic debugger and potentially obtaining sensitive data.

However, this last chmod command comes close to specifying the entire permission list for myfile. At some point, using absolute mode to specify the exact permissions for a file is appropriate:

```
chmod 766 myfile      # file is "world write"  
chmod 600 /etc/lilo.conf # only owner perms set  
chmod 1777 /tmp        # set "sticky bit" on /tmp  
  
chmod 4711 myfile      # set-UID executable, no read
```

Of course, making a file completely world-writable by setting its mode to 666 is dangerous, particularly if the file contains logging information or system configuration data. The second example above is where we set the mode on /etc/lilo.conf to 600 to protect the clear text password string stored there. The third example sets the normal permissions on the /tmp directory; everybody on the system has read, write, and execute permissions on the directory, but the sticky bit is set so that users cannot clobber each other's files. The last example sets permissions in a similar fashion to the chmod u+s,go-r myfile symbolic mode command, though the absolute mode specifies exactly the complete vector of permission flags.

User and Group Ownership

- chown changes file ownership:

```
chown eric /home/eric
```

- chgrp changes group ownership:

```
chgrp other /home/eric
```

- Usually, you can do both at once:

```
chown eric.other /home/eric      # old BSD  
chown root:root /etc/lilo.conf   # modern
```

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

User and Group Ownership

Sometimes simply changing the permissions on a file is not enough. Administrators also need the ability to change the owner or group owner of a file or directory. The chown (change owner) and chgrp (change group) commands are used to modify ownership rights to files and directories:

```
chown eric /home/eric  
chgrp staff /home/eric
```

In fact, on most Unix systems the chown command can set the file owner and group owner simultaneously. The usual syntax looks like chown eric:staff /home/eric, with the format being <owner>:<group>. Older Unix systems may use a period instead of a colon—chown eric.staff /home/eric—but now this is very rare.

Most Unix systems allow only the superuser to use the chown command. Allowing normal users to chown files, even files that they own, can be dangerous. For example, a user could create an executable and make it set-UID. If that user is then allowed to change the ownership of that file to root, he would have a set-UID root executable that he could use to compromise the system. For this reason, if the chown command is allowed to be run by a normal user on a Unix system, the command generally will strip off the set-UID and set-GID bits automatically when the owner of the file is changed.

Allowing users to chown their own files can also have an impact on the system if file system quotas are enabled, restricting users to a fixed amount of disk consumption. If normal users can run chown, then they can make their files owned by other users on the system, effectively giving themselves "more disk space" by stealing it from other users. Ultimately, this can become a denial-of-service attack, at least as far as the other users on the system are concerned.

It should also be noted that both chown and chgrp allow the user to specify either usernames and group names or instead to use numeric UIDs and GIDs, or even a combination of the two. Most users and administrators find it more convenient to use human-readable names, however.

The Unix Password System

SANS Security Essentials – © 2014 Eric Cole

The Unix Password System

A basic overview of how Unix accounts are structured and how the operating system stores account information is necessary before we can look at how to manage users, restrict accounts, and apply other sorts of access controls. The basic Unix account structure actually is very simple. In fact, one of the criticisms of the basic Unix security model is the lack of "granularity" of access control provided to the administrator. As you will also see, many of the administrative functions provided by other operating systems (such as forcing password changes at next login, and keeping a "history" of previously used user passwords) are not widely implemented on Unix systems.

Usernames and Passwords

- Every user has an assigned username and password
- Usernames and passwords are case-sensitive
- Usernames are generally limited to eight characters for backwards compatibility

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Usernames and Passwords

Every user on the system has an identity assigned to her in the form of a unique username. Associated with each username is a password that the user must enter in order to be able to log into the system. Once they've logged in, users may change their passwords using the passwd command (note, that's passwd, not password).

Usernames can contain any alphanumeric characters and are case-sensitive. Thus, Alice and alice are different usernames. Most Unix installations stick to all lowercase usernames by convention, just to avoid confusion. Older Unix systems assumed that usernames would be eight characters or less—even modern Unix operating systems may contain certain programs that will abort when encountering a user name longer than eight characters. Many sites have adopted an eight-characters-or-less username policy just to be on the safe side.

Eight or less lowercase-only characters is a rather restrictive username space. This often becomes an issue in mixed Unix and Windows environments when the administrators want users to have a common username for both their Windows and Unix logins. The more restrictive Unix standard often becomes the lowest common denominator requirement for selecting usernames.

Unix passwords also are case-sensitive. You are probably familiar with the common mistake of attempting to enter a password with the caps-lock key on. Unix systems generally allow passwords to be made up of any printable character: alphanumeric characters, punctuation marks, and even spaces. Some Unix systems allow non-printable control sequences in passwords, as well.

Usernames Versus User IDs

- Usernames are purely for the convenience of human beings
- Unix systems store ownership information in terms of User IDs (UIDs)
- Commands such as chown will accept either usernames or UIDs

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Usernames Versus User IDs

Each username on the system is also associated with a numeric user ID (UID) value. All usernames, UIDs, and passwords are stored (along with other information about the user's account) in the system account database. On most Unix systems, this database actually is two files: /etc/passwd and /etc/shadow. We examine the format of these files in some detail later on in this module.

What is important to understand for right now is that as far as the Unix operating system is concerned, the UID entry for the user in the system account database is what is important, not the username. Usernames are only for the convenience of the human beings who use the system. All data about file and directory ownerships is stored internally in the operating system by UID.

The primary purpose of the /etc/passwd file, in fact, is to allow the system to relate UIDs to usernames for the convenience of users and administrators on the system. If a user's entry is deleted from the passwd file, Unix commands still will display the numeric UID on files that were owned by that user but will be unable to associate that UID number with a more human-readable username. Unix commands that deal with file ownership generally accept either usernames or UIDs as arguments.

Originally, the maximum possible UID value on a Unix system was 65,535 (UIDs were "signed" two-byte quantities). And, of course, some UIDs are already reserved by the operating system. Fortunately or unfortunately, we have reached a point now where large Unix installations are dealing with total user populations of over 65,000 users. In these cases, not all users can have unique user IDs.

It turns out that this can cause serious problems, especially when sharing files between systems using protocols like NFS. Remember that Unix access controls are based on UIDs and not usernames—if two users share the same UID and can crossmount each other's home directories, then they can access each other's files!

The Superuser

- Unix systems have an all-or-nothing security model
- Superuser access provides the capability to control all files, processes, and devices
- By convention, the superuser account is named root

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

The Superuser

The Unix security model recognizes two classes of users: normal users who can only manipulate their own data and a single all-powerful superuser who can do anything to any object on the system. The superuser can read any file, change the permissions and ownership on any file, delete any file, change user passwords at will, start and stop processes, and add and remove devices. The goal of an attacker trying to break into a Unix system is to get superuser access because then the attacker "owns" the system.

Over the years, Unix has received a certain amount of justifiable criticism for this all-or-nothing security model. Recently, Unix operating system vendors have started implementing more granular sorts of access controls at the kernel level, such as the Role-Based Access Controls (RBAC) provided by SELinux and grsecurity. It is ironic that the original, more trivial Unix security model was developed originally in reaction *against* earlier operating systems (notably Multics) that had exactly the same sorts of complicated security models now being reintroduced into Unix.

From the early days of Unix, the primary superuser account has been the root user. People familiar with the Windows environment may find it simplest to think of the Unix root account as being equivalent to administrator access under Windows. Unix users tend to use root and superuser interchangeably.

UIDs and the Superuser

- Any account with a UID of 0 has superuser privileges
- Other UID 0 accounts may exist besides the root account; these are usually locked
- Attackers often try to create new UID 0 accounts to get root access

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

UIDs and the Superuser

However, the superuser account could just as easily be called "eric." As far as the Unix operating system is concerned, any account with UID 0 has superuser privileges. Some sites even go so far as to grant certain users administrative rights to the system by setting the UID on their accounts to 0. This is a terrible idea, however, because it ruins any possibility of auditability on the system. Nobody will be able to tell which of these users crashed the system or removed some critical file. Plus, any one of those accounts might have an easily guessable password that will allow some outside attacker unlimited access to your system. Furthermore, even trivial mistakes by these—such as accidentally deleting the wrong directory—can cripple the machine.

Some Unix operating systems include a few UID 0 accounts besides root for special purposes. Generally, these accounts will be set up so that normal interactive logins are impossible—either the account has an invalid password or logging into the account runs some special program that performs a certain task rather than giving interactive command access. A favorite tactic of attackers is to change the configuration on UID 0 accounts that are normally locked in order to have a "back door" to access the system as the superuser. Audit extra UID 0 accounts on a regular basis, or remove them from the system entirely if they are not needed.

Another ploy on systems with large account databases is to add accounts with UID 0 buried someplace in the middle of the database. Administrators may not notice the new account, and again the attacker has a "back door" to superuser access on the system.

Becoming Superuser

- "Anonymous" root logins are bad:
 - No audit trail, no accountability
 - May allow exhaustive guessing attacks
- Under normal circumstances, log in as a normal user and use su or sudo
- You should probably still allow for root logins on console in case of emergencies

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Becoming Superuser

Unix systems generally provide three legitimate ways of getting superuser access. In the Unix lingo, this is typically referred to as "going root."

The first mechanism is to log into the system as root with the root password. This is usually referred to as an "anonymous" root login because there's no way to tell exactly which user is logging in as root. Anonymous root logins are a bad idea because there is no accountability and no possibility of an audit trail. If the system crashes right after somebody logs in as root, whom should the administrator track down to find out what happened? Also a remote attacker may be able to try an exhaustive guessing attack to break into the root account. Better that you not allow remote root logins at all.

It occasionally is necessary to log in as root on the system console in an emergency when there is no other way to access the system, but the system console should be the only place where root logins are allowed. In particular, *never* allow root logins over the network—particularly when using insecure channels such as telnet or rlogin from which an attacker with a password sniffer can grab the root password.

The second standard way of getting root access is with the command /bin/su or /usr/bin/su, depending on the version of Unix. This command actually allows a normal user to become any other user on the system, as long as she knows the other user's password. However, su most commonly is used to become root. In fact, running su without specifying an alternate username defaults to su root. Every time a user runs the su command, information about the attempt is logged in the system logs, whether the attempt succeeded or failed. This way, administrators can find out who did what in the case of system problems and detect when users are trying to gain unauthorized access.

It is considered good practice to always type the full pathname of the su program; that is, type **/bin/su** and not just **su**. This helps to avoid getting trapped by Trojan horse su programs designed to capture the root password. If an attacker is able to get his malicious su program into a directory that appears before /bin or /usr/bin in the administrator's executable search path, problems can result.

A third mechanism for granting root access on Unix systems is the sudo tool. The sudo command allows a user to run a single command with root privilege as long as he has a proper listing in the /etc/sudoers file created by the system administrator.

The sudo tool has a number of advantages over the normal Unix su command:

- It prompts users for their own passwords rather than the root password. At many sites that use sudo, most of the users and administrators may not even know the root password for the systems on which they work! This makes life easier when administrators and users leave the company because the root password no longer has to be changed on all systems.
- It allows for fine-grained access controls. The administrator may specify a list of specific commands that a given user may execute with superuser privileges and even in some cases specify which command-line options may be used.
- It produces a much higher level of logging than the normal su command. Every command executed via sudo is logged in great detail.

Most of the newer open source Unix operating systems include sudo by default. For operating systems that do not, the source code is widely available and compiles easily on many different platforms. It is easy to install and configure and is well worth using.

"System" Accounts

- Unix operating systems tend to come with a lot of dummy accounts for various apps
- Typically, these are accounts with low UID numbers (UID < 100?, 500?, 1000?)
- Attackers will sometimes activate these accounts as "back doors" into the system
- If you're not using a particular service or app, then remove (or block) the account

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

"System" Accounts

Unix systems typically will ship with a number of accounts that are associated with a particular application on the system rather than an individual user. For example, NFS uses the special "nobody" user; Oracle requires an "oracle" database administrator account; and the system might have an "apache" or "www" user set up for running the Web server. These accounts are typically given low UID numbers to distinguish them from other accounts in the password file. Red Hat appears to reserve all UIDs less than 500 for "system" type accounts, whereas Debian Linux uses UIDs less than 1000. Other Unix systems like Solaris reserve UIDs 0-99.

Usually the default /etc/shadow file from the vendor ships with these accounts disabled and with an invalid password entry, but attackers commonly will try to insert valid password strings as a back door in these system account entries. Setting invalid shells on these accounts in /etc/passwd where possible is a good idea, as is regularly auditing the system passwd and shadow files for changes to these accounts.

Disabling the account may be preferable to deleting the account altogether. In most cases, each of these system accounts "owns" one or more files and directories on the system. If the account were completely deleted from the passwd and shadow files, then the original UID for that account might end up getting re-assigned to a different user. Remember that Unix systems store file ownership information by UID. By reassigning the UID to a new user on the system, the administrators have created a situation where the new user now "owns" the operating system files that were created for the previous system account. If these files are critical configuration files, directories, or system devices, then the user might be able to get increased privileges on the system.

A similar argument can be made for blocking rather than deleting user accounts when a user leaves the organization or no longer requires access to the system. Typically that user will leave behind some number of files and directories she owns. Administrators want to avoid UID overlaps, but it also can be useful to know who originally created a file long after that individual has moved on.

Of course, having large numbers of blocked accounts in the passwd and shadow files makes it harder to audit these files for malicious changes, so some sites opt for a strict policy of purging old user accounts and data from their systems soon after a user leaves. Similarly, if the system administrator is certain that a given non-user account no longer owns any files on the system, that account may be deleted from the passwd and shadow files.

Passwd File (1)

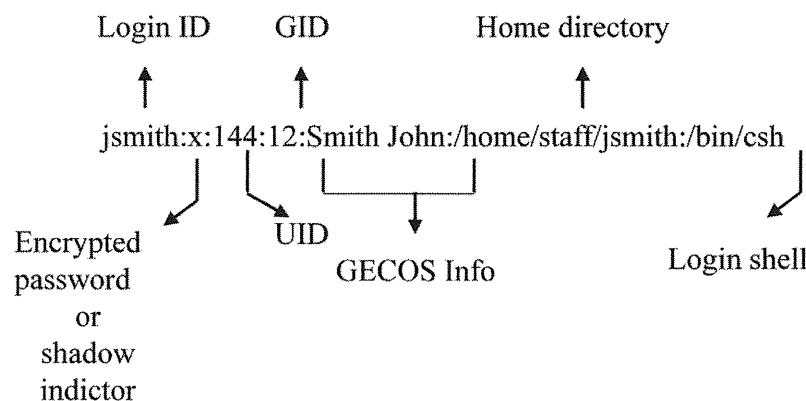
- In early Unix, the encrypted password was stored in the /etc/passwd along with user information. Everybody could read the encrypted passwords, but the hardware was too slow to crack a well-chosen password.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Passwd File (1)

Originally, Unix systems kept a user's one-way encrypted password along with user account information in the /etc/passwd file. This file is used by many tools (such as ls) to display file ownerships, and so on. The file needs to be world-readable because processes must match user ID #'s with the users' names. This lends itself to potential security risks, and therefore, Unix incorporated what is now called the shadow file to hold encrypted passwords.

Passwd File (2)



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Passwd File (2)

Each field in the password file is separated by a colon (:).

- 1st: User ID.
- 2nd: Location for the encrypted password (normally an x to indicate the system is shadowed).
- 3rd: User identification number.
- 4th: User's primary group.
- 5th: GECOS field (remarks).
- 6th: User's home directory.
- 7th: What happens upon login (normally this will be a process).

Passwd/Shadow

- Today's Unix environment uses a two-file system: the /etc/passwd file and a second file normally called shadow
 - However, the original format of /etc/passwd did not change
- Most Unix platforms encrypted passwords are stored in /etc/shadow:
 - AIX "/etc/security/passwd"
 - FreeBSD "/etc/master.passwd"
- The shadow file is accessible only by root

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Passwd/Shadow

On current Unix systems, the encrypted password is no longer contained in the world-readable /etc/passwd file. It is now contained in the more secure file called /etc/shadow. In place of the encrypted password, an x is used as a place holder.

AIX

/etc/passwd
/etc/security/passwd

FreeBSD

/etc/passwd
/etc/master.passwd

HP-UX

/etc/passwd
/tcb/files/auth/r/root

LINUX(Red Hat) & Solaris

/etc/passwd
/etc/shadow

Shadow File

username:passwd:last:may:must:warn:expire:disabled:reserved

username:Npge08pfz4wuk:9479:0:10000: : :

Days since 1 Jan 70
password last changed

Days after which
password must be changed

Days before password
may be changed

Days before password
is to expire

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Shadow File

username:passwd:last:may:must:warn:expire:disabled:reserved

- Username: The name matching the entry in the /etc/passwd file.
- Password: The encrypted password along with the salt and type of encryption.
- Last: The number of days since the last password change based on 1 Jan 1970.
- May: The number of days before the password must be changed again.
- Must: The number of days before the password must be changed.
- Warn: The number of days before the password must be changed, so the user gets a warning message.
- Expire: The number of days before the password expires.
- Disable: The number of days before the account gets disabled.
- Reserved: Not used at this time.

useradd

useradd <username>

```
File Edit View Terminal Tabs Help
[root@localhost ~]# su tigger
[tigger@localhost root]$ id
uid=502(tigger) gid=502(tigger) groups=10(wheel),502(tigger) context=user_u:syst
em_r:unconfined_t
[tigger@localhost root]$ useradd winnie
useradd: unable to lock password file
[tigger@localhost root]$ sudo useradd winnie
Password:
[tigger@localhost root]$ tail -n 1 /etc/passwd
winnie:x:504:504::/home/winnie:/bin/bash
[tigger@localhost root]$ id
uid=502(tigger) gid=502(tigger) groups=10(wheel),502(tigger) context=user_u:syst
em_r:unconfined_t
[tigger@localhost root]$
```

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

useradd

The useradd command allows administrators the ability to create new user accounts. It adds new accounts to the /etc/passwd, /etc/shadow, and /etc/group, and it creates the user's home directory.

Enabling Password Aging (1)

- Controlled by two configuration files:
 - /etc/login.defs
 - PASS_MAX_DAYS: Maximum number of days a password is valid. The default is 99,999 days.
 - PASS_MIN_DAYS: Minimum number of days before a user can change the password since the last change. The default is 0 days.
 - PASS_MIN_LEN: Minimum length of password (in Linux, this value is controlled by the PAM). The default is 0.
 - PASS_WARN_AGE: Number of days when the password change reminder starts. The default is 7 days.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Enabling Password Aging (1)

On most Unix OSes, the password-expiring and password-rotation features are disabled. To enable these features, one must simply apply values to configuration files in /etc/login.def.

Account Password Information

chage -l <user>

```
File Edit View Terminal Tabs Help
[root@localhost default]# chage -l piglet
Last password change : Dec 29, 2008
Password expires      : Mar 29, 2009
Password inactive     : Apr 05, 2009
Account expires       : never
Minimum number of days between password change : 3
Maximum number of days between password change : 90
Number of days of warning before password expires : 7
[root@localhost default]#
```

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Account Password Information

The chage command provides an administrator a quick look at the current value for password aging.

Enabling Password Aging (2)

– /etc/default/useradd

- INACTIVE: Number of days after password expiration that account is disabled. Value of -1 feature disabled (default).
- EXPIRE: Account expiration date in the format YYYY-MM-DD. Default is None.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Enabling Password Aging (2)

In addition to configuration file entries, the administrator must add a value in the /etc/default/useradd file.

An Introduction to PAM

- PAM stands for Pluggable Authentication Modules
- System libraries handle Linux authentication
- Four management groups:
 - Authentication
 - Passwords
 - Sessions
 - Accounts
- Configuration files are in /etc/pam.d

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

An Introduction to PAM

The Pluggable Authentication Modules (PAM) are system libraries that handle Linux authentication and the many tasks that go along with it. Originally invented by Sun Microsystems, PAM has evolved into a robust technology that simplifies and strengthens security on Linux systems by providing individual modules for a large number of diverse applications requiring authentication services.

PAM employs four management groups that handle specific types of authentication requests. The four groups perform the following functions:

- Authentication (auth): Used to establish a user identity and possibly membership in a group.
- Passwords (password): Used when a user provides authentication credentials such as a password.
- Sessions (session): Aids in implementing services and tasks associated with user authentication.
- Accounts (account): Used to perform actions not based on authentication.

All of the PAM configuration files are located in /etc/pam.d, and they are named for the service that they control. The common format for these files is as follows:

Type Control Module-path Module-arguments

The type field is the management group the module is a part of. The control field specifies the action to take if the PAM authentication fails. The Module-path and Module-arguments fields specify the name and path for the module in use (within the /lib/security directory) and what arguments should be passed to it.

An example of a PAM file is shown here (for the file /etc/pam.d/su):

```
#%PAM-1.0
auth      sufficient  pam_rootok.so
# Uncomment the following line to implicitly trust users in the "wheel" group.
#auth      sufficient  pam_wheel.so trust use_uid
# Uncomment the following line to require a user to be in the "wheel" group.
#auth      required   pam_wheel.so use_uid
auth      include    system-auth
account  sufficient pam_succeed_if.so uid = 0 use_uid quiet
account  include    system-auth
password include    system-auth
session  include    system-auth
session  optional   pam_xauth.so
```

Enforce Stronger Passwords

- To start enforcement, modify the file /etc/pam.d/system-auth by adding the following:

```
"password requisite /lib/security/$ISA/pam_cracklib.so retry=3  
minlen=8 lcredit=-1 ucredit=-1 dcredit=-1 ocredit=-1"
```

- The pam_cracklib module checks the password against dictionary words and other constraints

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Enforce Stronger Passwords

Password enforcement is managed on Linux systems by the PAM (Pluggable Authentication Modules). The main configuration files are maintained in the /etc/pam.d directory.

System-wide password enforcement is controlled mainly by the system-auth configuration file. To set minimal password requirements, the following variables must be set:

- minlen=\$: Minimum length of password must be \$.
- lcredit=-\$: Minimum number of lower case letters must be \$.
- ucredit=-\$: Minimum number of upper case letters must be \$.
- dcredit=-\$: Minimum number of digits must be \$.
- ocredit=-\$: Minimum number of other characters must be \$.

Restricting Use of Previous Passwords

- Edit the /etc/pam.d/system-auth file and add/change the following:
 - pam_cracklib arguments:

```
"password requisite /lib/security/$ISA/pam_cracklib.so  
    retry=3 minlen=8 lcredit=-1 ucredit=-1 dcredit=-1  
    ocredit=-1 difok=3"
```

- pam_unix arguments:

```
"password sufficient /lib/security/$ISA/pam_unix.so nullok  
    use_authtok md5 shadow remember=6"
```

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Restricting Use of Previous Passwords

The following variables set reuse policy:

- difok = \$: This is the number of characters that must be different from the old password \$.
- pam_unix: Modules remember old password uses.
- remember=\$: This value is the number of old passwords remembered.

If /etc/security/opasswd does not exist, you will need to create the file.

Locking User Accounts after too Many Login Failures

- Edit the /etc/pam.d/system-auth file and add/change the following:
 - pam_tally module is used to count fail login:
“auth required /lib/security/\$ISA/pam_tally.so
onerr=fail no_magic_root”
“account required /lib/security/\$ISA/pam_tally.so
per_user deny=5 no_magic_root reset”

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Locking User Accounts after too Many Login Failures

By default, most Unix systems do not lock an account out after a set number of login attempts. By setting this value, you prevent brute force password-guessing attacks. Do not apply to the root account because this can cause a DoS (Denial of Service) for the administrator.

The pam_tally module will lock individual user accounts after too many failed su or login attempts.

onerr=fail tells the system what to do when reaching a set number of fail=lock accounts.

No_magic_root tells the system not to lock the root account. This prevents a DoS against the root account.

per_user keeps account of each individual use.

Deny= \$ is the number of attempts made before account locks \$.

faillog -u <user> lists the current number of bad logins.

faillog -u <user> -r will unlock the account.

faillog -u <user> -m -1 will turn off locking on lock out of a particular user.

You can also unlock accounts with these commands:

- passwd -l <user>
- usermod -L <user>
- passwd -u <user>
- usermod -U <user>

Summary

- Understand Unix file permissions
- Understand basic user and group management
- Understand password security capabilities
- Utilize common utilities for detection and recovery

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Summary

In this section, we provided the knowledge individuals involved in managing or securing Unix systems must know. We discussed what Unix is, how the basic structure is designed, basic commands to manipulate processes and files, permissions associated with protecting and providing the basic access rights, user and group management, and an understanding of the password capabilities of the operating system. We concluded the section with a few common utilities to assist in the identification of running applications and network connections.

Module 31: Securing Linux/Unix

BOOT PROCESS, RUN LEVELS, AND SERVICES

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Module 31: Securing Linux/Unix

BOOT PROCESS, RUN LEVELS AND SERVICES

This section intentionally left blank.

Objectives

- Understand the Unix boot process
- Understand the common Unix services:
 - How to enable and disable
- Identify a method for providing defense in depth to our services

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Objectives

In this section, we identify how Unix systems boot from the kernel to the selection and function of individual services. Additionally, we will address a method of providing defense in depth to our key services at the host level.

How are Services Started?

- With Unix, services are generally started in one of four ways:
 - At boot time
 - Automatically by inetd/xinetd
 - Cron scheduler service (crontab)
 - Command line

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

How are Services Started

Let's recap the boot and service start-up procedures.

Services are normally started in one of four ways.

- At boot by a start-up script
- By init or rc scripts, which use configuration files such as inetd/xinetd
- By cron or other automated process management utilities
- Command line by an authorized user

Ways Processes are Started: init

- Init is short for initialization
 - Starts init processes at boot
 - Stops init processes at shut down
- It provides a layer between the kernel and the user
- Configuration file: /etc/inittab

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC.

Ways Processes are Started – init

Once the Linux kernel finishes loading, it immediately starts looking for what init processes it needs to start. Think of an init process as being started when the computer boots and continues to run until the system is shut down—a program that provides the most fundamental layer of goodness that goes between the kernel and the user. Init is short for initialization[1] and it comes in two styles: SysV, as in Debian/Ubuntu and Red Hat/Fedora, and BSD, which you'll find in FreeBSD and other BSD-type distributions. Let's take a look at the SysV approach, which is the most widely used in Linux.[2]

The /etc/inittab file lists each of the init processes the system should start at boot and stop and shut down. For example, the "getty" service, which allows users to begin logging in to the system[3] would have an entry in /etc/inittab like this:

```
2:23:respawn:/sbin/getty 38400 tty2
```

Let's move through this line left to right: The first number tells the system the id of the program (2), which is a unique number assigned to each process. The second entry shows what runlevels should use this process in (2 and 3) (the runlevel is the mode the kernel is in that can include single user, multi-user, and reboot, depending on what number is used—in this case, 2 and 3 are multi-user.[4]) The third entry (respawn) tells the system that it should automatically restart this process should it die for some reason. The last entry is the full path to the binary for getty with some parameters.[5]

Summary

Init (short for initialization) starts init processes in Linux at boot and stops them when the system shuts down. Init processes provide a fundamental layer between the kernel and the user. A list of these init processes is kept (in Debian) in the /etc/inittab file.

References

- [1] <http://en.wikipedia.org/w/index.php?title=Init&oldid=177210611>
- [2] <http://www.freeos.com/articles/3243/>
- [3] http://www.comptechdoc.org/os/linux/startupman/linux_suiglog.html
- [4] <http://en.wikipedia.org/wiki/Runlevel>
- [5] http://www.comptechdoc.org/os/linux/startupman/linux_suinit.html

Ways Processes are Started: inetd

- It oversees network services
- Start a service when there is a request (*not at boot*)
- Service must be listed in two files:
 - /etc/inetd.conf
 - /etc/services
- It requires TCP wrappers to provide security

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Ways Processes are Started – inetd

Inetd is the "super server" daemon that is responsible for starting network services. Note that unlike init, which starts processes at boot, inetd starts network services like the Apache web server and the SSH server when there is a request from the network for this service.[1]

The configuration file for inetd is /etc/inetd.conf and any network service it starts must have a corresponding line for this service in this file as well as the /etc/services file in order to function properly. For example an inetd.conf entry for the telnet service would look something like this:

```
telnet stream tcp nowait root /usr/sbin/tcpd in.telnetd
```

Among other things, this line tells inetd the path to the telnet file and what kind of service it is ("stream" in this case). The corresponding /etc/services entry for telnet would look like this:

```
telnet 23/tcp
```

Which tells inetd that this is a TPC-based service that listens for connections on port 23. In short, /etc/inetd.conf connects names of services to names of servers, and /etc/services connects port numbers to protocols.[2]

Summary

Inetd starts network services when requests for that service come in from the network. Any service needs to be listed in the inetd conf file /etc/inetd.conf and the /etc/services file to function correctly.

References

[1] <http://www.freeos.com/articles/4314/>

[2] <http://en.wikipedia.org/w/index.php?title=Inetd&oldid=170384094>

Ways Processes are Started: xinetd

- It has more security features than inetd:
 - Performs access control
 - Helps prevent DOS attacks
 - Logs all sorts of information
 - Binds IP address to a service
- Xinetd can start a server that isn't listed in /etc/services

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Ways Processes are Started – xinetd

Introduction

Xinetd is the eXtended InterNET services daemon and is considered a replacement for inetd. Xinetd has some expanded functions as well as some nice added security features. You can think of Xinetd as inetd with TCP Wrappers like support built in. This allows the system to perform security functions before key services are started. [1]

Example

For instance, an admin can limit which hosts can connect to a specific network service by their:

- IP address
- Hostname
- Domain name
- Time of access

This is useful for all sorts of services.[2] You can imagine the admin on an intranet server for a company limiting connections to only those addresses and hostnames that belong to the company. They might also want to limit access to only working hours if staff shouldn't be accessing the intranet in the middle of the night.

Xinetd

is also flexible enough to set reasonable limits on (among others) the number of connections any host can make, how many incoming connections will be answered at a time, and killing services if any of these limits are exceeded. This is an excellent way to fend off denial-of-service (DOS) attacks and portscans on a server.[3] Admins can set up detailed logging in xinetd that can help figure out if there is a DOS attack or some other nastiness at a later time.

Network Filtering

An administrator can bind (or force) a specific service to a specific IP address. If the system has more than one network interface, this capability can be extremely useful. For example, a company's web server may have one network adapter on the Internet and another on the local area network. It would be a reasonable configuration to bind secure shell (SSH), frequently used to remotely access a system, to the local area network interface. This way the system would not allow any SSH connection request from the Internet and which would foil any reconnaissance or attacks on this service. [4]

Alternative Services

Xinetd also has the ability to start a server that isn't specifically listed in /etc/services, which is somewhat desirable and scary at the same time. If you are working on a development server and testing a new piece of software you'd like to run as a service, xinetd will definitely help get the thing running when there's an appropriate network request for that service. The liability with this function is that you'd want to be very careful about how you configure this non-standard service to run, and you'd definitely want to use the access control part of xinetd to keep from making it available to the entire Internet.

Summary

Developers wrote xinetd as a replacement for inetd. Xinetd has more functions and added security, like being able to control access to services and help protect against port scans and DOS attacks. Xinetd can also make detailed logs, bind specific IP addresses to specific services, and start services that are not listed in the /etc/services file.

References

- [1] <http://www.linuxfocus.org/English/November2000/article175.shtml>
- [2] <http://www.xinetd.org/faq.html>
- [3] <http://en.wikipedia.org/wiki/Xinetd>
- [4] http://www.linuxcommand.org/man_pages/xinetd8.html

The Methods for Starting Processes: Cron

- Scheduling daemon
- Starts an action (in the background) at preset time
- Can also be employed by users
- Crontab file used to store the jobs that are going to run

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Ways Processes are Started: Cron

Init starts things at boot, inetd and xinetd when there is a network request. What about processes that need to run at times other than boot or when a request comes in? Cron to the rescue!

The cron daemon (crond) works in total sync with the system clock. Every minute crond makes its way through the /etc/crontab file to see if there is a task for it to perform at a scheduled time. An entry in this file consists of a specific or reoccurring time, the user the system should execute the task as, and the command itself or path to a script. It should be noted that any task crond executes is performed in the background. Here's an example of an entry in a crontab file:

```
25 6 * * * root /usr/bin/apt-get update
```

The first five fields in this line represent the minute, hour, day of month, month, and day of week the command should be executed. The format for this time is explained in the man (manual) page for crontab[1]:

field	allowed values
minute	0-59
hour	0-23
day of month	1-31
month	1-12 (or names, see below)
day of week	0-7 (0 or 7 is Sun, or use names)

A field may be an asterisk (*), which always stands for first-last.

In this example, the command should be executed at 25 minutes past 6 am every day of the month; the asterisk (*) is a wildcard. The system will start this command as the user root and is meant for a Debian-based system as its the command to update the APT repository contents for the local system.

If the administrator allows it, regular users can use a crontab file of their own, usually located in their home directory called crontab. This is a nice feature for users and allows them—with the usual restrictions placed on their account—to start programs or scripts at a preset time using the same syntax as the /etc/crontab file.

Summary

Crond is the daemon that can start programs or scripts at a preset time by making an entry in the /etc/crontab file. Regular users can also take advantage of cron by editing the crontab file in their own home directory.

References

- [1] [http://man.cx/crontab\(5\)](http://man.cx/crontab(5))

How Unix Systems Boot (1)

1. boot loader:

- First, a boot loader is executed in order to prepare the system to begin executing the Unix kernel code
- Typically two stages:
 - The first stage is the Master Boot Record (MBR)
 - The MBR, in turn, usually runs a more complex boot loader from elsewhere on the system disk
 - The second stage is the loader program that actually starts the Unix kernel

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

How Unix Systems Boot (1)

Unix boot happens in a two-stage process: the MBR, and then the kernel loader.

How Unix Systems Boot (2)

2. Kernel initialization and execution:

- The kernel is software responsible for initializing and managing the system's hardware resources
- The kernel handles communication between the applications running and the hardware devices

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

How Unix Systems Boot (2)

In the second stage, the system initializes the kernel and executes key software packages that are required to manage the system resources.

How Unix Systems Boot (3)

3. initial processes (init):

- Starts up some initial system processes that then allow other processes on the system to be run
- These processes manage systems:
 - Virtual memory system
 - Process scheduler

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

How Unix Systems Boot (3)

After the kernel process starts, it's time for the system to start the init process, which manages virtual memory and the process scheduler.

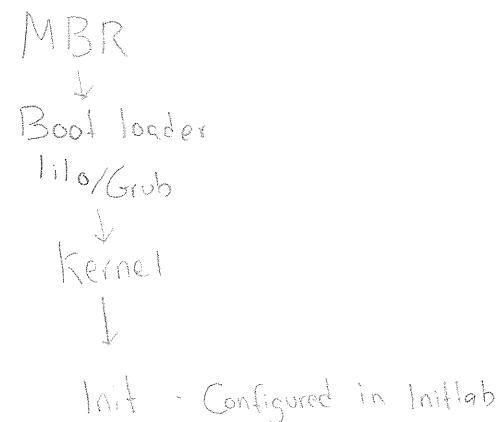
How Unix Systems Boot (4)

4. init run system start-up scripts:
 - Runs one or more start-up scripts, which actually start the programs and services that most users interact with

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

How Unix Systems Boot (4)

After init completes, it runs user-defined service start scripts. The determination of run levels is based on which scripts are configured to run at this level.



Boot Loader

- lilo
 - In lilo.conf, add two lines to the top of the file:
 - password=<password>
 - Restricted
- grub
 - In grub.conf, add the following line at the top of the file:
 - password --md5 <md5hash>
 - *Use the grub-md5-crypt to create hash.*

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Boot Loader

Securing the boot loader process would be the next logical step in securing an operating system. The requirement of a password to even boot the kernel has its pros and cons. Pro, one must have physical access to the system to enter the password, thus allowing for the physical restrictions of access to the systems. Con, again one must have physical access to start the system. A power outage or remote reboot, or any occurrence demanding the system reboot requires a physical access and password.

Two distinct methods of boot loaders for the Linux environment offer a password option: lilo and grub. All one has to do is add one or two lines to the boot loader configuration files and a restricted guideline and you are ready to force password loading.

Run Levels

- init selects which set of scripts to run based on the run level:
 - In a normal system boot, the run level is predefined in the file /etc/inittab
 - Each flavor of Unix has its own defined set of run levels
 - Normally defined rc0 through rc6 (rc=run condition)

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Run Levels

Solaris

S, s Single user mode. Doesn't require properly formatted /etc/inittab. File systems required for basic system operation are mounted.

0 Go into firmware (sparc)

1 System Administrator mode. All local file systems are mounted. Small set of essential system processes are running. Also a single user mode.

2 Put the system in multi-user mode. All multi-user environment terminal processes and daemons are spawned.

3 Extend multi-user mode by making local resources available over the network.

4 usually not used.

5 Shut the machine down so that it is safe to remove the power. Have the machine remove power, if possible.

6 Reboot

HP-UX

0 System is completely shut down. All processes are terminated and all file systems are unmounted.

1,s,S Single-user mode. All system services and daemons are terminated and all file systems are unmounted.

2 Multi-user mode, except NFS is not enabled.

3 Multi-user mode. This is the normal operational default state. NFS is enabled.

4 Multi-user mode with NFS

5 usually not used

6 Reboot

OpenBSD

- 1** Permanently insecure mode—always run system in level 0 mode.
- 0** Insecure mode—immutable and append-only flags may be changed. All devices may be read or written subject to their permissions.
- 1** Secure mode—system immutable and append-only flags may not be turned off; disks for mounted file systems, /dev/mem, and /dev/kmem are read-only.
- 2** Highly secure mode—same as secure mode, plus disks are always read-only whether mounted or not. The settimeofday(2) system call can advance only the time.

ULTRIX, Digital Unix / Tru64

- 0** System is completely shut down. All processes are terminated and all file systems are unmounted.
- 1** Single-user mode.—all system services and daemons are terminated and all file systems are unmounted.
- 2** Multi-user mode, except NFS is not enabled.
- 3** Multi-user mode—this is the normal operational default state. NFS is enabled.
- 4** Not Used
- 5** Not Used
- 6** Reboot

Irix

- S,**s** Enter single-user mode—when the system changes to this state as the result of a command, the terminal from which the command was executed becomes the system console.
- 0** Shut the machine down so it is safe to remove the power. Have the machine remove power if it can.
- 1** Put the system into system administrator mode. All file systems are mounted. Only a small set of essential kernel processes run. This mode is for administrative tasks such as installing optional utilities packages. All files are accessible and no users are logged in on the system.
- 2** Put the system into multi-user state. All multi-user environment terminal processes and daemons are spawned. Default.
- 3** Start the remote file sharing processes and daemons. Mount and advertise remote resources. Run level 3 extends multi-user mode and is known as the remote-file-sharing state.
- 4** Define a configuration for an alternative multi-user environment. This state is not necessary for normal system operations; it is usually not used.
- 5** Stop the IRIX system and enter firmware mode.
- 6** Stop the IRIX system and reboot to the state defined by the initdefault entry in inittab

Linux

- 0** Halt the system.
- 1** Single-user mode.
- 2-4** Multi-user modes—usually identical. Level 2 or 3 is default (dependent on distro).
- 5** Multi-user with graphical environment. This applies to most (but not all) distros.
- 6** Reboot the system and return to default run level.

inittab

- The init program looks at the /etc/inittab file for instructions on which program to start during system startup
- The first section passes the default run level (3 in the example that follows):
id:3:initdefault:
si::sysinit:/etc/rc.d/rc.sysinit
- The second section contains the location of the script to be run before all others:
- The third section contains the location of all run conditions directories on the system:
 - Additional instructions can also be included in this section:
 - What happens when CTRL-ALT-DEL is pressed?
 - What happens if UPS passes a power loss condition?
 - Also, provide how many tty are started

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

inittab

The inittab defines which run level will be default and where all the different run level scripts are located. It can also be used to assign predefined key values such as what to do when the CTL-ALT-DEL keys are pressed.

Run Condition Directory

- Typically, there is one directory for each run condition. These directories are usually named rcX.d (where X is the run level)
- In each directory are links to the scripts in init.d directory:
 - The scripts themselves are not found in the rc directory
 - Each link starts with either S or K (start or kill) followed by a number:
 - S08iptables
 - S12syslog
- The system will run the scripts only with an S during startup and a K during shutdown
- Number values tell the system the order in which to call the script (small to largest)
- Any file that does not start with an S or K will not be accessed
- An easy way to prevent a service that normally starts during a run condition is to rename the startup and shutdown links

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Run Condition Directory

All systems will have a rc (run command) directory. There will be one for each defined rc level.

This is where the system administrator can define which services start at each run level.

If the file starts with a capital S, it starts up on boot. If it starts with a capital K, it is killed during shutdown.

All others are ignored. The number values set the order, smallest to largest, in which they are run.

Inetd / xinetd

- The Internet services daemon or super server used to manage most of the TCP/IP daemons
- It runs in the background listening on common ports for connections and spawns the appropriate daemon
- inetd's replacement on some OS xinetds:
 - Configured in the file: /etc/inetd.conf
 - Configured in directory: /etc/xinetd.d/

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Inetd / xinetd

Often called a *super server*, inetd listens on designated ports used by Internet services such as FTP, SSH, and telnet. When a TCP packet or UDP packet comes in with a particular port number, inetd launches the appropriate server program to handle the connection. xinetd uses an updated version found in the latest Linux versions.

inetd

- Following are a few lines from an example /etc/inetd.conf file:
ftp stream tcp nowait root /usr/sbin/ftpd ftpd
ntalk dgram udp wait root /usr/sbin/talkd talkd
time stream tcp nowait root internal
time dgram udp wait root internal
- Any line that starts with an "#" is a remark. So an easy way to disable a service is to remark it out
- Fields are:
 - Service Name
 - Socket Type
 - Protocol Name
 - Wait/NoWait
 - Server Path
 - Server Args

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

inetd

ServiceName

Contains the name of an Internet service defined in the /etc/service file.

SocketType

Contains the name for the type of socket used for the service. Values for the *SocketType* parameter are:

stream: Specifies that a stream socket is used for the service.

dgram: Specifies that a datagram socket is used for the service.

sunrpc_tep: Specifies that a Sun remote procedure call (RPC) socket is used for the service, over a stream connection.

sunrpc_udp: Specifies that a Sun RPC socket is used for the service, over a datagram connection.

ProtocolName

Contains the name of an Internet protocol defined in the /etc/protocols file.

Wait/NoWait

Contains either the wait or the nowait instruction for datagram sockets and the nowait instruction for stream sockets.

The Wait/NoWait field determines whether the inetd daemon waits for a datagram server to release the socket before continuing to listen at the socket.

ServerPath

Specifies the full path name of the server that the inetd daemon should execute to provide the service. For services that the inetd daemon provides internally, this field should be internal.

ServerArgs

Specifies the command-line arguments that the inetd daemon should use to execute the server.

The maximum number of arguments is five.

The first argument specifies the name of the server used.

If the SocketType parameter is sunrpc_tcp or sunrpc_udp, the second argument specifies the program name and the third argument specifies the version of the program.

For services that the inetd daemon provides internally, this field should be empty.

xinetd

- It was originally developed as an open source replacement for inetd
- It can be installed on most Unix systems and is standard on Red Hat, Fedora, and many other Linux distributions
- Adds extra features over standard inetd, including:
 - More logging options
 - Built-in IP address-based access control
 - Redirection of services to services on other ports or other systems
 - Built-in support for warning banners
 - Resource thresholds

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

xinetd

xinetd features built-in access control mechanisms such as TCP Wrapper ACLs, extensive logging capabilities, and the capability to make services available based on time. It can also place limits on the number of servers that the system can start.

Xinetd Key Files / Directory (1)

- /etc/xinetd.conf: The global xinetd configuration file.
 - Read only once when the xinetd service is started

```
# Simple configuration file for xinetd
# Some defaults, and include /etc/xinetd.d/
defaults
{
    instances      = 60
    log_type       = SYSLOG authpriv
    log_on_success = HOST PID
    log_on_failure = HOST
    cps            = 25 30
}
includedir /etc/xinetd.d
```

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Xinetd Key files/Directory (1)

xinetd.conf

- **Instances:** Sets the maximum number of requests xinetd can handle at once.
- **log_type:** Configures xinetd to use the authpriv log facility, which writes log entries to the /var/log/secure file. Adding a directive such as FILE /var/log/xinetdlog here would create a custom log file called xinetdlog in the /var/log/ directory.
- **log_on_success:** Configures xinetd to log if the connection is successful. By default, the remote host's IP address and the process ID of server processing the request are recorded.
- **log_on_failure:** Configures xinetd to log if there is a connection failure or if the connection is not allowed.
- **Cps:** Configures xinetd to allow no more than 25 connections per second to any given service. If this limit is reached, the service is retired for 30 seconds.
- **includedir /etc/xinetd.d/:** Includes options declared in the service-specific configuration files located in the /etc/xinetd.d/ directory.

Xinetd Key Files / Directory (2)

- /etc/xinetd.d/ directory is the directory containing all service-specific files:
 - Contains the configuration files for each service managed by xinetd.
 - Names of the files correlate to the service.
 - File is read only when the xinetd service is started.
- service telnet
 - {
 - flags = REUSE
 - socket_type = stream
 - wait = no
 - user = root
 - server = /usr/sbin/in.telnetd
 - log_on_failure += USERID
 - disable = yes
 - }
- To Disable Service, ensure "disable=" is set to **yes**.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

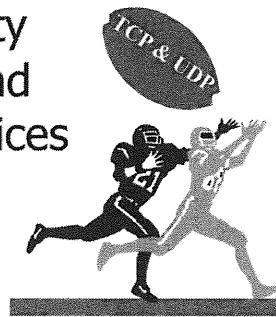
Xinetd Key files/Directory (2)

Service Configuration file

- **service:** Defines the service name, usually to match a service listed in the /etc/services file.
- **flags:** Sets any of a number of attributes for the connection. REUSE instructs xinetd to reuse the socket for a Telnet connection.
- **socket_type:** Sets the network socket type to stream.
- **wait:** Defines whether the service is single-threaded (yes) or multi-threaded (no).
- **User:** Defines what user ID the process will run under.
- **server:** Defines the binary executable to be launched.
- **log_on_failure:** Defines logging parameters for log_on_failure in addition to those already defined in xinetd.conf.
- **disable:** Defines whether or not the service is active.

TCP Wrappers (1)

- Written by Wietse Venema, co-author of SATAN and other security related tools
- TCP Wrappers is a simple utility that you can use for logging and intercepting TCP and UDP services started by *inetd* or *xinetd*

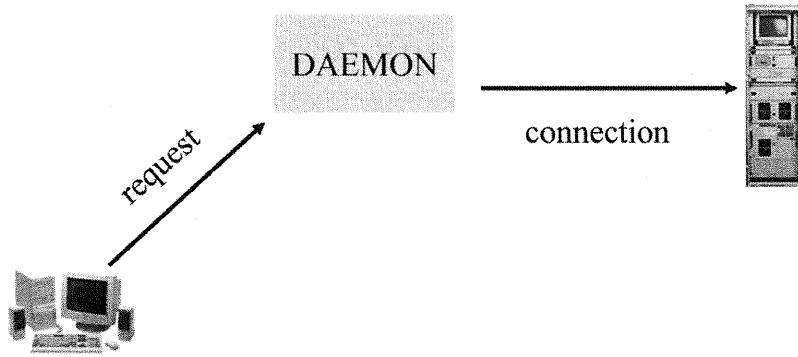


SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

TCP Wrappers (1)

TCP Wrappers is a host-based networking ACL system, used to filter network access to Internet Protocol servers on Unix operating systems. It allows host or sub-network, IP addresses, names and/or ident filtering and querying.

Before TCP Wrappers

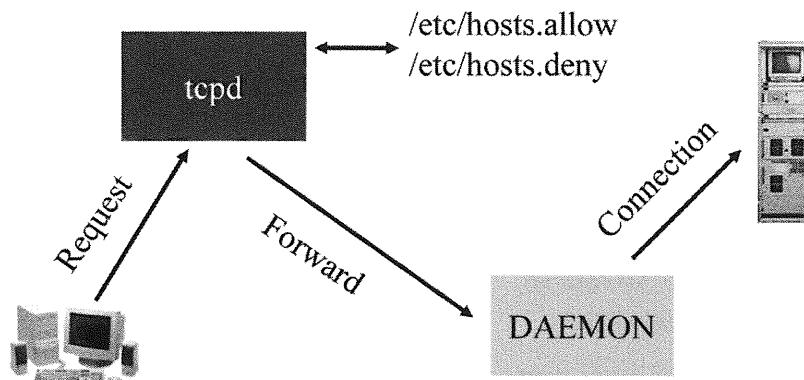


SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Before TCP Wrappers

This slide depicts a normal connection. Without a host-based firewall or ACL, access to services is a direct connection.

After TCP Wrappers



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

After TCP Wrappers

With TCP Wrapper installed, all connections must pass through a set of rules before being allowed to connect to a service.



TCP Wrappers (1)

- It optionally sends a “banner” to the connecting client
- Automatically performs a double-reverse lookup of the IP address. It drops a connection that does not match.
- It denies access to certain hostnames and services
- It logs all information to syslog

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

TCP Wrappers (1)

Beside being a manageable host-based firewall, it can provide additional features such as bannering a system, reverse name look up, and additional logging capabilities.

TCP Wrappers (2)

- Typically installed as tcpd
- Can cause “wrapped” services to fail if incorrectly installed
- Has two configuration files:
 - /etc/hosts.allow
 - /etc/hosts.deny
- Can be modified to alert the network administrator of suspicious activity

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

TCP Wrappers (2)

TCP Wrapper when installed, is normally in the /usr/sbin directory and called tcpd.

The main configuration files are hosts.allow and hosts.deny.

init.d

- The /etc/init.d directory, in most flavors of Unix, is the location of the control scripts (start, stop, resume, restart, and so on) for most services on the system
- Normally, there is one script for each service, and they identify what is to happen based on the request (stop, start, and so on)

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

init.d

The scripts are not actually stored in each of the rc directories but in the init.d directory.

The rc directory just has links that point to each respective script.

Each script contains the correct startup and shutdown steps for each process.

Service Command

```
service <service>  
<command>  
service --status-all
```

The screenshot shows a terminal window titled "root@localhost:~". The window contains the following text:

```
File Edit View Terminal Tabs Help  
[root@localhost ~]# service network stop  
Shutting down interface eth0: [ OK ]  
Shutting down loopback interface: [ OK ]  
[root@localhost ~]# service network start  
Bringing up loopback interface: [ OK ]  
Bringing up interface eth0:  
Determining IP information for eth0... done. [ OK ]  
[root@localhost ~]#
```

SANS Security Essentials - © 2016 Secure Anchor Consulting, LLC

Service Command

The service command runs a System V init script in as much of a predictable environment as possible, removing most environment variables and with a current working directory set to `/`.

Most System V init scripts are located in the `/etc/init.d/SCRIPT` directory.

The supported values of `COMMAND` depend on the invoked script; service passes `COMMAND` and `OPTIONS` to the init script unmodified.

All scripts should support at least the start and stop commands.

As a special case, if `COMMAND` is `--full-restart`, the script is run twice, first with the stop command, then with the start command.

`service --status-all` runs all init scripts, in alphabetical order, with the status command.

chkconfig

chkconfig --list

```
[root@localhost ~]# chkconfig --list
NetworkManager 0:off 1:off 2:off 3:off 4:off 5:off 6:off
NetworkManagerDispatcher 0:off 1:off 2:off 3:off 4:off 5:off 6:off
acpid 0:off 1:off 2:off 3:on 4:on 5:on 6:off
anacron 0:off 1:off 2:on 3:on 4:on 5:on 6:off
apmd 0:off 1:off 2:on 3:on 4:on 5:off 6:off
atd 0:off 1:off 2:off 3:on 4:on 5:on 6:off
auditd 0:off 1:off 2:on 3:on 4:on 5:on 6:off
autofs 0:off 1:off 2:off 3:on 4:on 5:off 6:off
cpuspeed 0:off 1:on 2:on 3:on 4:on 5:off 6:off
```

chkconfig -- level <#> <name> <on/off>

```
[root@localhost ~]# chkconfig --level 2 network off
```

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

chkconfig

The command chkconfig provides an overview of what services are started and what gets started at each run level.

It can also be used to toggle the on/off state for each service at each run level.

Service Management

- **The Golden Rule:**

- If you don't need it, turn it off
- If you're not sure, run a sniffer, validate the service, and with caution, turn it off!

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Service Management

The proper attitude to take when disabling services on a system is:

- If a service is not needed, turn it off.
- When in doubt, turn the service off and monitor the system!

Sites should change their thinking from "This service might be dangerous, it should probably be disabled" to "Is there one good reason why this service should be running?" The idea is to run only services that can be justified as being mission-critical.

Of course, whether or not a given service is needed or is mission-critical is a subjective judgment based on the business needs of the organization. In these cases, the right choice is rarely an absolute technical decision. There may be many technologies available to accomplish similar functionality, but some of the alternatives may be so difficult to manage or use that they are unacceptable, even though these solutions may provide higher levels of security.

For example, it is not absolutely necessary to run the NFS file-sharing service. There are other ways in Unix to share the same file across many machines, such as the rdist tool, which copies files from a central server to one or more machines. However, it is difficult to imagine a distributed workstation cluster using rdist to keep users' home directories in sync across multiple machines where users might work on files on many different systems.

The organization needs to make the call between absolute security and ease of use. Having made that call, the organization can look for additional ways to mitigate security problems caused by this decision. For example, the organization should use good NFS administration practices to protect their file servers and data, keep up-to-date on NFS-related patches, and deploy a strong firewall architecture around their enterprise network to protect corporate NFS servers and clients.

Also remember that this decision can and will be different for different parts of the network. NFS might be fine on an internal network protected by a strong firewall. On the other hand, it might be preferable to force administrators to distribute files using rdist via SSH on Internet-connected machines, such as Web and FTP servers. These systems are more easily accessible to external attackers, and they are also typically more critical to the business mission of the organization. A higher level of security is certainly desirable.

In addition to reducing the number of potential security vulnerabilities on the system, turning off services also improves system performance and reliability. Less memory and CPU time will be devoted to services that are not being used. The system will boot faster because there are fewer services to start at boot time. The system will also be more reliable because there are fewer processes running and less that could go wrong.

Patch a Disabled Service?

- Even if you disable a service, you should also keep them updated:
 - Prevent local exploits by unprivileged users
 - In case you re-enable services in the future

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Patch a Disabled Service?

Even if a service is not used, patch it! You may need it in the future, and it also helps prevent local exploits by your internal users.

Common Services

- File sharing:
 - NFS and Samba
- Naming:
 - NIS, LDAP, DNS
- RPC (Remote Procedure Call) services:
 - Portmapper
- Printing:
 - CUPS or lpd
- Internet:
 - Web, e-mail
- Network:
 - Routing, SNMP

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Common Services

Let's highlight a few common services used by networks to facilitate the sharing of resources among users. Although every Unix system offers a slightly different array of boot services, these services can generally be grouped into several broad categories. For example, Linux uses CUPS (the Common Universal Printing Service) to enable printing, but older proprietary Unix systems might use the standard Unix lpd (Line Printer Daemon) interfaces instead. Regardless of the specific implementation choices, however, there are certain security issues that are common to both printing systems, and it may be prudent to disable whichever printing system is configured on your particular flavor of Unix if you don't particularly have a need to print documents from your system. So as the next few slides discuss each of the categories shown on the slide, remember to consider all of the security issues we cover very carefully, even if your particular flavor of Unix doesn't run the exact same services as our Linux example system.

Network File System

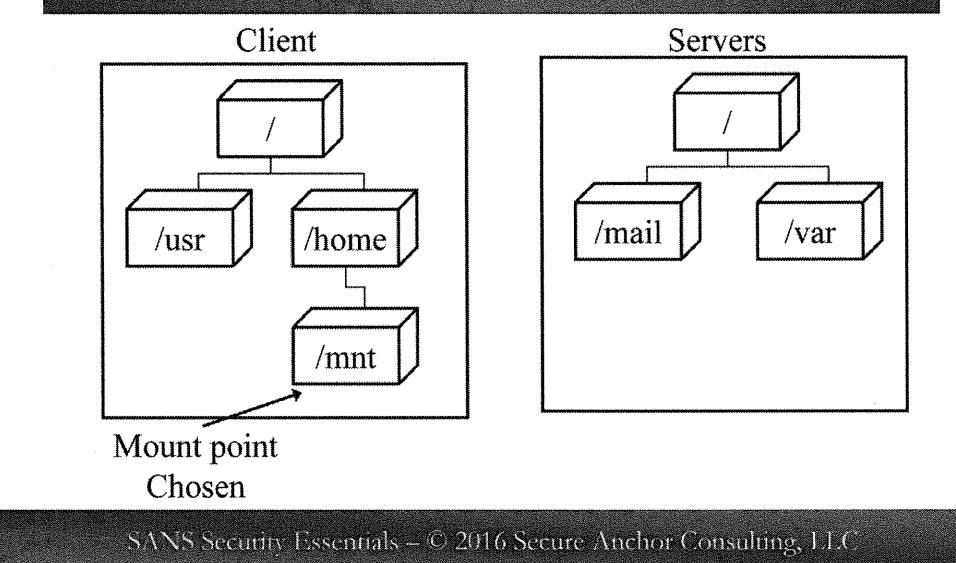
- The Network File System (NFS) provides transparent file access for clients with files and filesystems on “A” server
- It is important to make sure you have unique UIDs
- UDP port 2049

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Network File System

The Network File System (NFS) is a common service that is implemented as an RPC service. NFS allows certain directories and their contents to be made available to users of other computers.

NFS (1)

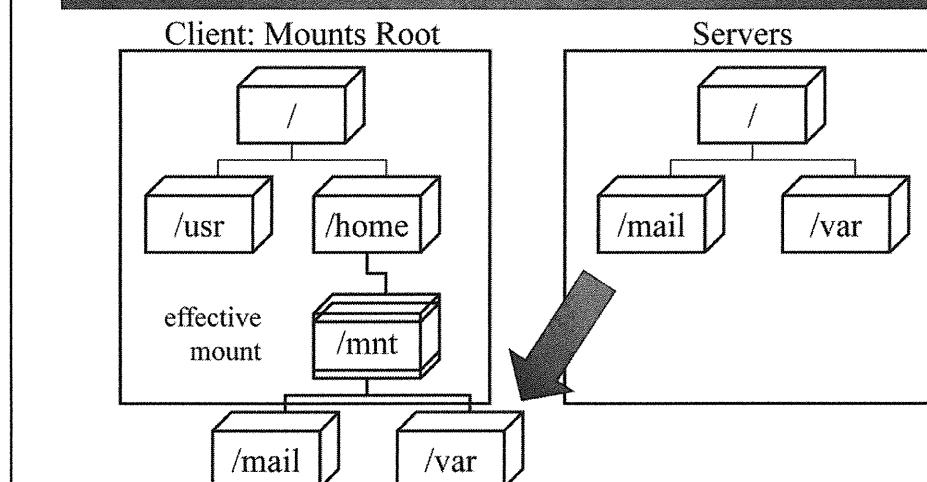


NFS (1)

In this example, we demonstrate how a server will establish an open share of its complete directory tree by establishing a share for the / directory. A client then creates a mount point (which is simply a directory in the file system) and points it toward the server's share point. Any attempt to look in that directory on the client will cause the client computer to make a remote procedure call to the server and display the contents of the server's hard drive instead of the clients local hard drive.

The user or client machine issues a command similar to the following: `mount <server IP>:<share name> /home/mnt`

NFS (2)



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

NFS (2)

This view shows the virtual file structure that exists by mounting the /home/mnt directory to the root of the remote server.

Samba

- Samba is software run on a Unix/Linux platform to allow a host to interact with a Microsoft Windows client or server as if it were a Windows file and print server

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Samba

Samba is a Unix utility that uses the Server Message Block (SMB) protocol to interact with Windows-based systems. This interaction is similar to the “connect” and “disconnect” of directory mapping and printer sharing on Microsoft systems.

NIS, LDAP, and DNS

- Significant security issues:
 - Sensitive information disclosure
 - Spoofing
 - Denial-of-service
 - Buffer overflows
- Protect name servers and deploy redundancy throughout network
- Utilize security configuration parameters for each service as appropriate

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

NIS, LDAP, and DNS

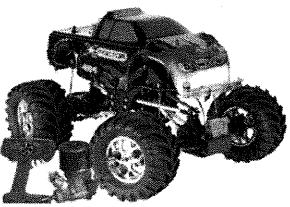
Unix systems generally employ one or more "naming services," including NIS (Networked Information Service), LDAP (Lightweight Directory Access Protocol), or DNS (the Domain Name Service). All of these services supply databases of information that can be accessed from all systems on the network. The closest analogues in the Windows universe are WINS and Active Directory.

NIS, formerly known as Yellow Pages (YP), was originally developed by Sun Microsystems but was widely adopted in the Unix community, and later by many non-Unix vendors. NIS was designed to allow administrators to centrally manage common system configuration files, such as the system password database and network configuration files, such as hosts and services. Clients could simply look this information up over the network, rather than the administrator having to make changes on every single system. This is an enormous win from the perspective of administrative convenience and had a lot to do with the early success of network computing.

LDAP is merely a specification on how to access generic data across a network from some central server. The information stored on that server and the way the data is organized are completely up to the local site, though certain standard schemas have been developed and are widely used. Many sites are now using LDAP as an alternative to NIS for storing information about users (including passwords), hosts, and other network devices. Unix LDAP implementations can even interoperate with Windows Active Directory servers, allowing for universal sign-on systems where users use the same usernames and passwords on both Unix and Windows.

DNS is the way IP address and host name information is shared on the Internet. Sites maintain DNS servers that hold information about local hosts at that site and consult the DNS servers at other organizations for information about their hosts. When a user types www.sans.org into a Web browser, the local DNS server consults the DNS servers for sans.org in order to find the correct IP address for the HTTP connection. Because sites have to maintain DNS servers in order to communicate with the outside world, most sites also use DNS as the standard way of sharing IP address and hostname information with internal hosts as well.

Remote Procedure Call



- Clients normally send commands to a server and the server sends replies back to the client
- Remote Procedure Call (RPC) is set up so that the client calls functions in the server program

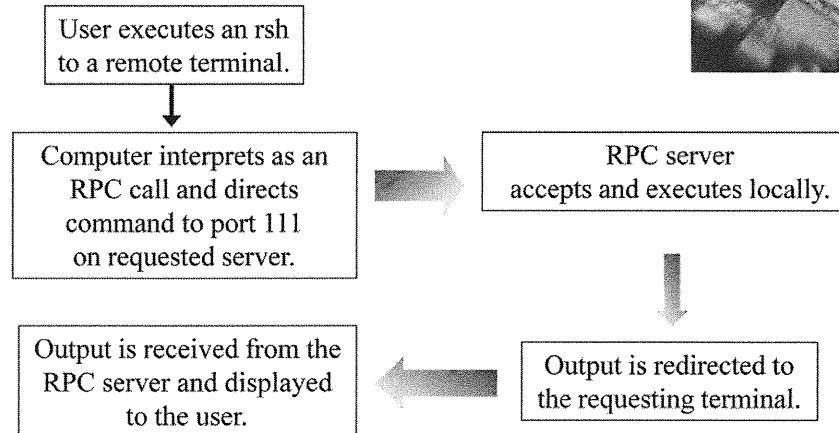
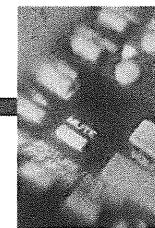
SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Remote Procedure Call

RPC (Remove Procedure is a powerful technique for constructing distributed, client-server based applications. An RPC is analogous to a function call. Like a function call, when an RPC is made, the calling arguments are passed to the remote procedure and the caller waits for a response to be returned from the remote procedure.

Remote Procedure Call

In Action



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Remote Procedure Call in Action

Diagram on the RPC call for information. In this example, it's requesting the function of a shell.

Port Mapper

- The RPC server programs containing the remote procedures use ephemeral ports
- A “registrar” keeps track of which RPC programs are using which ports
- Port Mapper uses UDP port 111 and TCP port 111
- Services register with the Port Mapper and clients query it



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Port Mapper

When an RPC starts on a computer, it registers with the portmapper service. In this way, portmapper tracks all RPCs running on the system. Portmapper can provide information on all of these RPCs to remote computers that desire their services.

Other RPC Services

- LOCKD: Run by both the client and server; it handles file locks
- STATD: Run by both the client and server; this daemon handles status of file locks
- AUTOMOUNTD: Mounts and unmounts NFS resources only when needed
- RSH: Allows a user to get a remote shell
- RCMD and REXD allow execution of programs or parts of programs remotely

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Other RPC Services

There are an array of other RPC services used in Unix systems.

Unix Printing

- History of buffer overflow problems
- Printing system often runs as privileged user and isn't careful about queue dirs
- Typical security problems have included:
 - Print any file (for example, /etc/shadow)
 - Overwrite any file on the system
 - Execute arbitrary commands with privilege
- Be sure to limit who can install printer daemons and control what is actually printed across the network

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Unix Printing

The Unix printing facility has historically had significant security problems. Many years ago these problems were magnified because the printing system ran with full administrative privileges. Thankfully, modern Unix systems now run the printing service as an unprivileged user to help reduce the impact of security issues.

Common security issues with Unix printing include:

- Buffer overflow problems in the Unix print daemon, which seem to have been particularly common lately. The Ramen worm propagated itself across Linux systems by exploiting vulnerable Linux print servers, among other things.
- In older eras, when disk space was expensive, the print system needed to be able to handle large print jobs that didn't fit in the system print queues. In these cases, the print system would simply make a link from the print queue directory to the actual file to be printed. If the print job actually resided in a user's home directory, or some other place a malicious user had access to, the user could simply replace the original file with a link to some other file on the system. The printing service would then print the file that it had been redirected to.
- When the printing service ran with administrative privileges, this meant that malicious users could print any file on the system—for example, the system password database! Now a malicious user can print only files that are readable by the special line printer user, lp, but this level of access can still be used to steal a user's print jobs flowing through the printer's queues. These print jobs can contain sensitive or proprietary information.
- The printing system may use print filters to convert output from one form to another before printing (for example, converting text files to Postscript). Some printing systems allowed the user to specify his own print filters. This would allow an attacker to run arbitrary commands as whatever user the print system is running as. Modern print systems generally restrict the choice of filters and the location of these files to the system administrator.

Common sense security rules apply to the Unix printing service:

- Disable the printing service if you have no need to print from your Unix systems. On most Linux systems, this is the CUPS (Common Universal Printing Service) software suite. Older, proprietary Unix variants (Solaris, HP-UX, and so on) will typically use the standard Unix lpd printing interface.
- Be sure to keep up-to-date the vendor patches.
- Consider deploying centralized print servers that are accessible only by administrators. Print jobs submitted from client systems should be immediately copied over to the print servers rather than being spooled directly from the clients. This shortens the time period that print jobs actually reside on client systems, and causes filter processing to be done on the more secure centralized print servers.

Web Servers

- To demonstrate Internet readiness, many systems ship with a Web server enabled
- If you're not serving Web pages, then you should shut the Web server off
- Practice secure Web server administration:
 - Run server as unprivileged user
 - Restrict access by IP address/password
- Beware CGIs and other executable code

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Web Servers

Since the popularization of the Web in the mid-90s, most Unix systems ship with a Web server as part of the base OS install—usually Apache. Often, this Web server is enabled at boot time by default. Even worse, in some cases vendors ship this Web server with some sample CGI applications that may contain security holes. This includes such holes as allowing remote users to run arbitrary commands on the Web server!

Obviously, if the machine does not need to serve up Web pages, then the Web server should be disabled.

Some of the more critical suggestions for securing a Web server include:

- Never run Web servers as the Unix administrative user, root. Always create a special unprivileged user for the Web server to run as.
- Where possible, restrict access to Web documents by IP address or by requiring a password.
- Beware CGI programs and other server side executables (PHP, Java, and so on). More Web security issues have been the result of these kinds of vulnerable scripts than any other single factor.
- Never put the source code for these executable programs anywhere in your document tree. Always locate CGIs in a special directory with restricted access. This is done by making sure you use the Apache ScriptAlias option to set up the CGI bin directory outside of the normal docroot (as opposed to using the ExecCGI option to put CGIs in the docroot). Then, make sure that the normal directory permissions on the CGI bin dir require special access to add CGIs.

E-mail

- 99.9% of machines don't need to be running an e-mail server:
 - Mail daemon receives mail from *other* hosts
 - Local mail clients run server binary from a disk
- Simplest configuration is best:
 - Config relays outgoing e-mail to a central server
 - Mail daemon is disabled to stop outside exploits
- Process the queue periodically via cron

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

E-mail

Typical Unix-like operating systems ship with an e-mail server enabled. In most cases, this is the standard Unix e-mail server, Sendmail (though some Unix distributions are starting to ship with the Postfix e-mail system instead, because many administrators have a perception that Postfix is more secure than Sendmail).

Because almost every Unix system ships with the Sendmail daemon enabled, many administrators think that you have to run Sendmail if users want to be able to send e-mail from the system. It turns out that this is completely incorrect.

When thinking about e-mail, it helps to conceptualize the sending of e-mail and reception/transfer of e-mail as two separate entities:

Nearly every Unix machine in your environment needs to be able to send out e-mail. This is not only e-mail generated by users on the system, but also automated e-mails from cron and other system services. The handling of newly generated e-mails is the responsibility of a process generally referred to as the Message Submission Process (MSP).

Only a small handful of machines in a typical environment are actually mail servers, machines that accept e-mail from other machines and deliver that e-mail into user mailboxes or transfer that e-mail on to other systems. Accepting e-mail from other machines and routing it to its final destination is the responsibility of the Mail Transfer Agent (MTA) process, which listens on port 25/tcp for incoming e-mail from other systems.

If all a given system ever does is send e-mail out to other machines, but never actually receives any incoming e-mail, why run an MTA process on this system at all? The only thing that the MTA process is doing on 99.9% of the Unix machines in most environments is listening on 25/tcp for the next remote Sendmail buffer overflow exploit. By disabling the MTA on most of the Unix systems in your environment, you will get ahead from a security perspective. Now you have to focus only on the e-mail security of the few machines in your environment that are actually mail servers and accepting e-mail from other machines.

Routing

- By default, many Unix systems start up a *dynamic routing daemon*
- The routing daemon modifies the route table based on information received via the network
- Route updates are usually accepted without authentication or verification
- Static routing avoids these issues entirely

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Routing

The Unix operating system kernel maintains a table of network routes that tells the machine how to direct network traffic leaving the host. By default, many Unix systems will start a dynamic routing daemon (routed or in.routed on many Unix systems) at boot time. This is a process that listens on the network for routing updates from nearby routers and changes the operating system routing table accordingly.

Unfortunately, in most cases these route updates have no built-in authentication mechanism. An attacker can send out bogus routing updates that cause the system to redirect traffic either to a non-existent router (denial-of-service attack) or to the attacker (man in the middle attack).

Rather than doing dynamic routing, the administrator has the option of configuring routes into the operating system routing table manually. This is typically referred to as static routing. Often the system only requires a single default route to the closest network router. On many Unix systems the IP address of this default router is usually placed in a configuration file to be set at boot time. This configuration file is often /etc/default/routing, though under Linux, the administrator sets the GATEWAY= parameter in either the /etc/sysconfig/network-scripts/ifcfg-* or the /etc/sysconfig/network files. Also note that if the system is configuring its network interfaces via DHCP, the machine will usually have its default route set as part of the DHCP configuration from the remote DHCP server.

Setting a default route at boot time will often automatically disable the dynamic routing daemon on the host. Sometimes this daemon will need to be disabled with a separate manual process.

Static routing is preferred from a security perspective but can make administration more difficult, and also make it harder for systems to automatically fail over to a backup router. However, most routers now implement some sort of "hot standby" protocol. The best advice seems to be to use a static default route on the end systems and let the routers handle fail-over.

SNMP

- SNMP is a remote system monitoring daemon many network management tools use
- It is critical to keep it patched and up to date
- If you must run an SNMP daemon, change the default "community string"

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

SNMP

Many Unix systems now ship with an SNMP daemon enabled by default. This allows administrators to monitor systems using common network management tools such as HP OpenView and MRTG. On the other hand, if the site is not currently using one of these tools, the only thing an SNMP daemon on the host accomplishes is giving potential attackers large amounts of data about the system and its configuration. Disable the SNMP daemon unless it is actively being used.

If the SNMP daemon is running on a host, experts recommend using something other than the default community string value of public. The community string is a weak password that the remote machine uses to request information from the local SNMP daemon. Typically, the administrator can set the community string in the file /etc/snmp/snmpd.conf. Also avoid allowing remote read-write access to the machine's SNMP daemon—permit only read-only access.

Useful / Good Services

- SSH for secure, encrypted logins
- NTP for time synchronization
- Correlate data from different logging sources:
 - Avoid disruptions due to incorrect time
 - Use cron for running tasks at scheduled times
- Syslog for capturing logging events

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Useful / Good Services

It may appear at this point that it is unsafe to run any service on a Unix system. However, there are a number of services that should generally always run on a machine:

SSH is the secure remote login and file transfer mechanism for Unix. SSH communications are encrypted, and SSH supports several alternate forms of authentication stronger than simple usernames and passwords. Free, portable SSH implementations are now available and most Unix vendors are starting to include SSH in their base operating system installs.

NTP is used to keep system clocks in synch with each other across a network. NTP is widely supported by Unix systems, Windows systems, and most network devices. After a security incident, when investigators are trying to correlate log events from hosts, routers, firewalls, and intrusion detection systems, it is critical that all of the log file timestamps match. Time synchronization is also important for time-based security software, such as Kerberos, or even when using a file-sharing protocol such as NFS.

crond automatically runs programs at times of the day specified by the system administrator or the users on the system. It is difficult to imagine running a Unix system without this feature, because without it, all administrative tasks would have to be done manually.

Syslog is the system logging daemon for Unix systems. It collects logging information from local and remote processes and either puts that information into log files on the local disk or relays the information to other systems.

Final Thought

- If you're not running any services from (x)inetd, turn the daemon off!
- Red Hat ships OS with xinetd enabled, but all services disabled...

Remember: If you don't need it, turn it off!

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Final Thought

Note that if no services end up being configured to run via inetd/xinetd, the administrator should simply stop the daemon from being started at boot time. Remember that all the administrator really needs to manage the system remotely is login access and file transfer ability. Because both of these tasks can be accomplished securely via SSH, it may not be necessary to use any inetd/xinetd-based services at all.

Summary

- Understand the Unix boot process.
- Understand common Unix services:
 - How to enable and disable
- Identify a method in providing defense in depth to our services
- Disabling unused services improves a system's security posture

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Summary

In this section, we identified how Unix systems boot from the kernel to the selection and function of individual services. Additionally, we addressed a method of providing defense in depth to our key services at the host level.

Disabling unused services is one of the most important tasks in helping to defend Unix systems from as yet unknown vulnerabilities. Remember the golden rules of service minimization: If a service is not needed, turn it off.

Disabling unused services improves a system's security posture, and it can also make the system perform better and be more reliable.

Of the services run by inetd/xinetd, the most commonly used services are the standard login and file transfer protocols: Telnet, FTP, and the BSD *r*-commands. However, SSH is in all ways a better replacement for these protocols. This is primarily because SSH communications are encrypted, which prevents eavesdropping and session hijacking attacks. Organizations should seriously look into replacing all of the old standard clear-text login protocols with SSH.

Having done this, a site may find that you have no real reason to run inetd or xinetd and can just shut these services off completely. Remember that reducing configuration files so that they include only the absolutely required services makes systems much easier to audit on an ongoing basis and can help stop certain automated exploits.

Module 32: Securing Linux / Unix

LOGS AND LOG MANAGEMENT

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Module 32: Securing Linux/Unix LOGS AND LOG MANAGEMENT

This section intentionally left blank.

Objectives

- Understand general Unix logging to identify activity (normal and nefarious)

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Objectives

This section focuses on the logging mechanism of Unix systems. It identifies the most common logs and how they can be used in to ensure security of the system is maintained through monitoring, detection, and response.

Important Log Files

- wtmp/wtmpx
 - utmp/utmpx
 - lastlog
 - history files
 - sulog
 - httpd
 - syslogd
 - messages\syslog
 - secure
 - FTP logs
 - maillog
- } Binary

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Important Log Files

These are the main logs we are going to cover.

On most Unix Flavors, wtmp, utmp, and lastlog are binary files. All other logs are typically plain ASCII files.

WTMP Log

- It keeps track of logins and logouts
- It is similar to UTMP, but it grows in length and keeps historical data
- Because it is a binary file, a special utility called “last” is used to access the WTMP structure

/var/log/wtmp

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

WTMP Log

WTMP is often the most useful log when tracking an intruder’s activities because it records historical login and logout data as well as both shutdowns and reboots. The WTMP log is often stored in /var/log/wtmp.

The historical records saved by the WTMP log are useful for tracking a specific intruder. This log provides more information than the lastlog, which overwrites the previous login information when a new connection is made.

WTMP Log “last” Output

User ID	Term ID	Remote Host ID	Date ↓	Time On	Time Off	Session Time
jbrown	ttyp2	mordor	Fri Feb 26	10:05 - 12:07		(02:01)
jsmith	ftp	207.196.92.170	Fri Feb 26	09:20 - 09:31		(00:10)
bwaters	ttyp5	32.97.106.11	Fri Feb 26	09:10 - 09:10		(00:00)
jsmith	ttyp4	z.glue.alpha.edu	Fri Feb 26	08:17 - 09:31		(01:14)
bwaters	ttyp3	32.97.106.11	Fri Feb 26	08:11 - 09:34		(01:22)
bwaters	ttyp2	32.97.106.11	Fri Feb 26	08:09 - 09:34		(01:25)
jsmith	ttyp0	131.118.249.226	Fri Feb 26	05:09	still logged in	
	ttyp0					Network Connection
	ttyq0					Network Connection
	ttyr0					Network Connection
	ttyS0					Network Connection
	ttyS0					Serial Connection

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

WTMP Log “last” Output

This is a print out of lastlog on a system. The following is the breakdown of each column: 1st column shows the userID, the 2nd column shows the connection point, and the 3rd column shows where the connection is made from. The next few columns show the date and time of the connections. The last column shows the total time of the connections.

UTMP Log

- Keeps track of users currently logged into the system
- Provides output for the w, finger, and who command
- May be world writeable and inaccurate
- Updated by the login program

```
batman % who
bob          tttyp2      Sep   18  14:22:23
jillk        tttyp3      Sep   18  14:24:21
batman %
```

/var/run/utmp

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

UTMP Log

The UTMP log tracks users currently logged into the system as opposed to the wtmp log, which records only the user logon events.

utmp w Output

USER	TTY	FROM	LOGIN@	IDLE	JCPU	PCPU	WHAT
stud2	ttyp0	208.141.82.220	8:34am	0.00s	0.43s	0.15s	w
stud18	ttyp1	stud18.class	9:48am	0.00s	0.43s	0.43s	-bash
stud19	ttyp2	stud19.class	9:49am	1:27	0.53s	0.53s	-bash

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

utmp w Output

This is the output of the w command.

- The 1st column is userID.
- The 2nd column is the connection point.
- The 3rd column identifies the source.
- The 4th column is the time connected.
- The 5th column is how long a connection has been idle.
- The 6th column is the JCPU time, which is the time used by all processes attached to the tty.
- The 7th column is the PCPU time, which is the time used by the current process, named in the what field.
- The 8th column is what process is currently running.

Lastlog

- Keeps track of each user's most recent login time
- Records initiating IP address
- Displayed every time the login program is run

```
BSDI BSD/386 1.1  unixbox (ttyp3)
login: jdoe
password:
Last login: Mon Sep 18 14:21:23 from batman.edu
```

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Lastlog

Lastlog records a user's most recent login information. It records the date and time and the IP address of the machine being used. This information is displayed every time the login program is run.

SULOG (1)

- Records the usage of the switch user command su
- su is often used by hackers to switch to usernames that have *rlogin* access to other machines, or to su to root-level access.
- Exists on out of the box Solaris and Irix machines

/var/adm/sulog

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

SULOG (1)

Sulog is used to log the use of the switch user command (su).

SULOG (2)

Activity	Date	Time	Terminal	Starting User	Ending User
SU	06/26	14:45	-	jsmith	-root
SU	06/26	14:45	+	jsmith	-root
SU	07/31	11:36	+	mjohnson	-root
SU	07/31	13:22	-	mjohnson	-root
SU	07/31	13:22	+	mjohnson	-root

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

SULOG (2)

This slide shows a sulog report. Information is recorded regarding the time, date, and terminal the command was executed on.

- The 1st column is what command (always su).
- The 2nd and 3rd columns are the date and time, respectively.
- The 4th column is success(+) or failure (-).
- The 5th column is where the request is made from (source).
- The 6th column is what user to user change was requested or made.

HTTP Logs (1)

- Most World Wide Web servers maintain logs that track the originating IP address of each connection

```
129.2.98.138 -- [26/Feb/1999:09:19:22 -0500] "GET /mrtg/mrtg-m.gif HTTP/1.0"
200 6039
129.2.98.138 -- [26/Feb/1999:09:19:22 -0500] "GET /mrtg/mrtg-l.gif HTTP/1.0"
200 1216
129.2.98.138 -- [26/Feb/1999:09:19:22 -0500] "GET /mrtg/mrtg-r.gif HTTP/1.0"
200 2994
129.2.98.138 -- [26/Feb/1999:09:19:23 -0500] "GET /mrtg/207.196.92.129.4.html
HTTP/1.0" 200 7198
129.2.98.138 -- [26/Feb/1999:09:19:23 -0500] "GET /mrtg/207.196.92.129.4-
day.gif HTTP/1.0" 200 11961
```

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

HTTP Logs (1)

Most World Wide Web servers maintain logs that track the originating IP address of each connection. They also maintain information on which pages were viewed, the date and time, and actions.

HTTP Logs (2)

```
129.2.98.138 - [26/Feb/1999:09:19:22 -0500] "GET /mrtg/mrtg-m.gif"
HTTP/1.0" 200 6039
129.2.98.138 - - [26/Feb/1999:09:19:22 -0500] "GET /mrtg/mrtg-l.gif"
HTTP/1.0" 200 1216
129.2.98.138 - - [26/Feb/1999:09:19:22 -0500] "GET /mrtg/mrtg-r.gif"
HTTP/1.0" 200 2994
```

- The Originating IP
- Almost Never Used
- Username Used for Request
- Date and Time of the Request
- Actual Text of the Request
- The Error Code Returned to the Browser (200 - OK, 404 - "File Not Found")
- The Number of Bytes Returned to the Browser

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

HTTP Logs (2)

This slide shows an example of an HTTP log. The originating IP address for each connection and the attempted activity are recorded.

FTP Logs (1)

- The WU-FTP maintains extensive logs to track incoming connections
- Solaris and IRIX and other Unix variants typically log only FTP connections
- Typically shows the originating IP address of the FTP connection
- It is commonly called *xferlog*

/var/log/xferlog

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

FTP Logs (1)

The FTP log keeps track of all files transferred to and from the server, thus they are commonly known as the xferlog. These logs typically record the date and time of the transfer, information on the file transferred, and the IP address of the originating machine.

FTP Logs (2)

File transfers on SGI machines are logged in the
/var/adm/SYSLOG:

```
Dec  4 06:54:40 6D:greenmachine ftpd[17046]: connection from
208.141.82.111
Dec  4 06:54:43 6E:greenmachine ftpd[17046]: FTP LOGIN FROM
208.141.82.111 as garman
Dec 31 09:57:31 6D:greenmachine ftpd[8557]: connection from
208.141.82.111
Jan  5 09:36:55 6D:greenmachine ftpd[12331]: connection from
208.141.82.111
Jan  5 09:40:56 6D:greenmachine ftpd[12340]: connection from
208.141.82.111
Jan 13 08:42:20 6D:greenmachine ftpd[18108]: connection from
208.141.82.111
```

**Notice only the fact that an FTP connection that occurred is logged,
not what files were uploaded or downloaded.**

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

FTP Logs (2)

The following is the basic information provided by minimal logging of FTP traffic: date, time, system name reporting, process reporting, and what action transpired.

FTP Logs (3)

- It is possible to make small changes to the inetc.conf and the syslogd.conf to log FTP much more effectively:

Irix

Old /etc/inetc.conf

```
ftp stream tcp nowait root /usr/etc/ftpd ftpd -l
```

New /etc/inetc.conf

```
ftp stream tcp nowait root /usr/etc/ftpd ftpd -l -l
```

Notice the simple change

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

FTP Logs (3)

It is possible to make small changes to the inetc.conf and the syslogd.conf to log FTP much more effectively: by adding an extra `-l`, we can improve the logging.

FTP Logs (4)

```
Mar  6 15:57:47 6D:greenmachine ftpd[3606]: connection from gandalf
Mar  6 15:57:55 6E:greenmachine ftpd[3606]: FTP LOGIN FROM gandalf as jdoe
Mar  6 15:58:02 6D:greenmachine ftpd[3606]: mkdir /usr/people/jdoe/tools
Mar  6 15:58:10 6D:greenmachine ftpd[3606]: get /usr/people/jdoe/ufsrestore.c
```

By simply editing the `inetd.conf` and restarting the `inetd`, system administrators can drastically improve their FTP logging.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

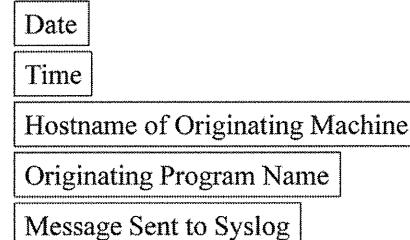
FTP Logs (4)

By simply editing the `inetd.conf` and restarting the `inetd`, system administrators can drastically improve their FTP logging.

This an example of extended logging of FTP.

Maillog (1)

- Maillog sends its logs to the syslogd. You can see the standard five fields



```
Feb 25 16:27:33 fs sendmail[2302]: QAA02300:  
to=jsmith@sso.sytexinc.com
```

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Maillog (1)

Some mail programs also record their logs to the syslog. This data is stored following the standard five field format: date, time, hostname, originating program, and message.

Maillog (2)

- The first field in the message (“QAA02300”) is the queue identification number assigned to this message. If there are multiple log entries pertaining to this message, this identification number will appear in each one of them.

```
Feb 25 16:27:33 fs sendmail[2302]: QAA02300:
```

**This field also appears in the message ID of the e-mail.
The following are the e-mail headers for the above-referenced e-mail.**

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Maillog (2)

The first field in the message (QAA02300) is the queue identification number assigned to this message. This field also appears in the message ID for the e-mail (see the next slide).

Maillog (3)

```
Received: from fs.sso.sytexinc.com [207.196.92.159] by sso.sytexinc.com with ESMTP  
(SMTPD32-4.06) id A158713C00DA; Thu, 25 Feb 1999 16:32:08 EDT  
Received: (from nobody@localhost)  
    by fs.sso.sytexinc.com (8.8.8/8.8.8) id QAA02300;  
    Thu, 25 Feb 1999 16:27:33 -0500 (EST)  
    (envelope-from lwaters@sso.sytexinc.com)  
Date: Thu, 25 Feb 1999 16:27:33 -0500 (EST)  
From: lwaters@sso.sytexinc.com  
Message-Id: <199902252127.QAA02300@fs.sso.sytexinc.com>  
X-Authentication-Warning: fs.sso.sytexinc.com: nobody set sender to lwaters@sso.sytexinc.com using -f  
To: jsmith@sso.sytexinc.com  
Errors-To: lwaters@sso.sytexinc.com  
Reply-To: lwaters@sso.sytexinc.com  
MIME-Version: 1.0  
Content-Type: text/plain  
Content-Transfer-Encoding: 7bit  
X-Mailer: IMP/PHP3 Imap webMail Program 1.99 beta  
Sender: lwaters@sso.sytexinc.com  
Subject: Re: how...??  
X-UIDL: 203714959  
Status: U
```

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Maillog (3)

This screen shows an e-mail header. The highlighted information is the queue identification number for this message. If this same identifier appears in the e-mail header and in the mail log, it can be used to correlate an e-mail with its associated log file entries.

Messages (SYSLOG)

- Records major events that take place
- SU to root
- Failed login attempts
- May need root-level access to view this log
- Generated by the syslogd

/var/log/messages

Irix	- /var/adm/SYSLOG
Solaris	- /var/adm/messages
HP	- /usr/adm/syslog

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Messages (SYSLOG)

The messages log (syslog) can record major events that take place on the system. Some major events that the administrator would want to be aware of would be an attempt to switch the user's privileges (su) to root as well as failed login attempts.

Messages

- There are five fields present on each line of a syslog-generated file

- 1 Date
- 2 Time
- 3 Hostname of Originating Machine
- 4 Originating Program Name
- 5 Message Sent to the Syslogd (follows “:”)

```
Jan 25 16:53:38 mjackson PAM[pamh[446]]: (login) session opened  
for user root on ttym1
```

```
Jan 25 16:53:38 mjackson login[446]: ROOT LOGIN ON ttym1
```

PAM = Pluggable Authentication Module

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Messages

Messages logged by syslog include these five fields: date, time, hostname, originating program, and the message with instructions for the syslogd.

The syslogd

- The syslog utility consists of a daemon that accepts incoming log messages and deals with them in accordance with the rules found in the /etc/syslog.conf

Any program which wants to generate log messages may do so through calls to the syslog interface

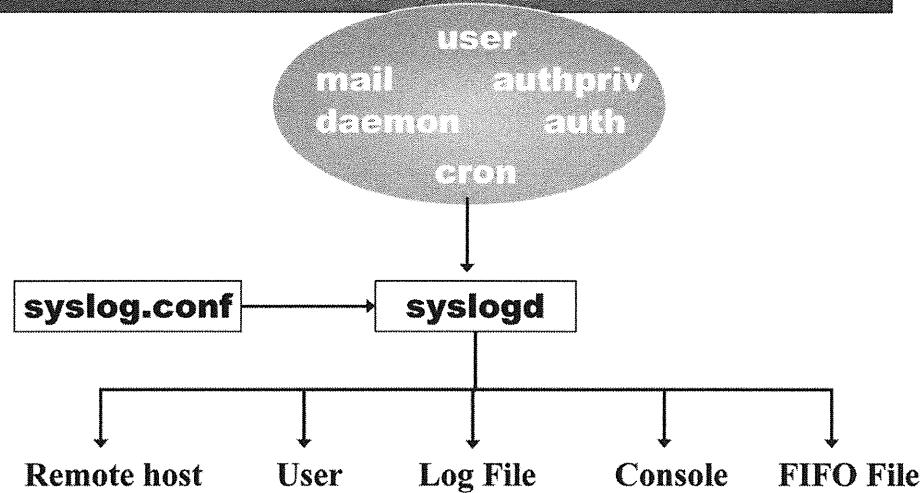
SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

The syslogd

Many Unix systems use syslogd to control log files. It uses a configuration file located at /etc/syslog.conf. The syslog.conf file contains information on logs and locations of the logs on the system.

The syslogd

System Facilities



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

The syslogd

This section intentionally left blank.

syslog.conf

.err;kern.debug;auth.notice;mail.crit	/dev/console
*.notice;kern.debug;lpr.info	/var/log/messages
mail.crit;news.err	/var/log/messages
mail.info	/var/log/maillog
authpriv.*	/var/log/secure
cron.*	/var/cron/log
mail.*	/var/log/maillog
*.alert	root
*.err	root
*.emerg	*

Facility

Level

Action

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

syslog.conf

This screen shows an example of a syslog.conf file. You'll notice two main fields: the selectors in the column on the left and the actions in the column on the right. The selector field has two parts, the facilities and the level of priority, for each action.

Facilities

auth	Authentication activity. Use authpriv.
authpriv	Authentication and PAM messages
cron	Messages associated with cron and at
daemon	Messages associated with daemons such as inetd
Kern	Kernel messages
Lpr	Messages related to printing services
Mail	E-mail (imap, pop, smtp)
News	Messages from News Server
Syslog	Messages from syslog
User	Messages from a user program
local0 – local16	For use with customized programs. SSH and remote machines (CISCO – Local 4).

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Facilities

The facility merely specifies how the message was produced in the syslog file. On the previous page, several logs were generated by the mail programs such as IMAP, POP, or SMTP. Those lines may be tagged with the mail keyword.

Levels

Emerg or panic	System is unusable
Alert	Situation needs to be fixed immediately
Crit	Error exists which will not allow proper execution of a program
Err	Error exists which will not allow proper execution of a component of a program
Warning	Warning message
Notice	Normal condition of significance
Info	Informational
Debug	A lot of information
None	Used to exempt a facility when using a wildcard
*	All levels except none

`*.info; mail.none;authpriv.none` `/var/log/messages`

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Levels

The second part of the selector is the priority level. The levels, listed here in descending order of priority, indicate the level of alert of the log message.

Actions

- **File**

*.info; mail.none;authpriv.none	/var/log/messages
---------------------------------	-------------------

- **Terminal or printer**

uucp, news.crit	/dev/console
uucp, news.crit	/dev/tty1
uucp, news.crit	/dev/lp1

- **Remote host: Remote host must have invoked syslogd with -r.**

authpriv.*	@192.168.1.100
------------	----------------

- **Username**

mail.*	root
*.emerg	*

- **Fifo file (made with the *mkfifo* command) for script processing**

authpriv.*	!/path/to/fifofile
------------	--------------------

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Actions

The actions column specifies how specific messages should be handled. Low priority informational messages might be stored to a file, whereas critical messages might be sent to a terminal or output on a printer. The administrator has control over specifying how the messaging information should be handled.

Configuring Logrotate

- /etc/logrotate.conf
- Directives:
 - include
 - daily, weekly, monthly, size
 - missingok
 - rotate <n>
 - create <perms> <owner> <group>

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Configuring Logrotate

Logrotate is configured in the aptly named /etc/logrotate.conf. Logrotate.conf starts with definitions of some global options. These options will be used as defaults for later entries. Order counts in logrotate.conf. The later definitions will overwrite the earlier definitions. This means that if you like, you can set some global values, follow them by the entries you wish to apply them to, and then set new global values, followed by more file entries. However, keep in mind that the more complicated your file, the easier it is to make mistakes. Logrotate has an include command that allows you to put your configuration commands in multiple files and read them in. This can be used to reduce complexity. You can name a file or a directory to be read in. When a directory is given, all files in the directory are read in alphabetical order. Linux packages expect to be able to include logrotate configuration files in the directory /etc/logrotate.d, so that directory is usually included after the global options. Entries for logfiles to rotate consist of the filename (or names), followed by any file specific directives in curly braces {}. So you can get an example of how this looks, the default /etc/logrotate.conf is included below. In addition to the above mentioned items, it includes entries to rotate two non-syslog, non-package logfiles: /var/log/wtmp and /var/log/btmp.

```
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 rotations worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
#compress
```

```
# packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own wtmp, or btmp -- we'll rotate them here
/var/log/wtmp {
missingok
monthly
create 0664 root utmp
rotate 1
}

/var/log/btmp {
missingok
monthly
create 0664 root utmp
rotate 1
}

# system-specific logs may be configured here
```

A few directives in this example are not explained in the comments. For example, missingok tells logrotate to continue on to the next file if the file doesn't exist, rather than quit with an error. The create directive gives the file permissions owner and group to create the file with after rotating. The rotate directive tells how many copies of the logfile to keep.

The example shows two of the directives that specify how long to keep the file, weekly and monthly. There is also a daily directive and the size directive, which specify that the file should be rotated when it reaches a certain size. The size directive is followed by the size in bytes (kilobytes can be specified with a k and megabytes with an M.) For instance, size 300M will rotate the log file whenever logrotate runs and finds that it is larger than 300 megabytes.

logrotate.conf for Syslog Files

- Closing and reopening the logfile
 - postrotate ... endscript
 - sharedscripts
- Remove /etc/cron.daily/sysklogd
- Additional directives for compression and e-mailing files

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

logrotate.conf for Syslog Files

In Unix systems, a program that has a file open for writing will continue to write to that file, even if it is moved or removed. Many programs will open their log files when they start and not close until they shut down. Most long-running programs, including syslog, have a method to tell them to close their log file. Most often, this involves sending the process a HUP signal (kill -1 <pid>). Logrotate has a number of options to allow you to run commands before or after rotating the files, which can be used to facilitate this. For syslog files, you want to use the postrotate directive to give a command to send an HUP to syslog. Then, you use the endscript directive to. If you use one entry for all your syslog files, you can use the sharedscripts directive to run this command only once.

To use logrotate instead of /etc/cron.daily/sysklogd to manage syslog files on non-Red Hat based systems, add the following entry to the end of your logrotate.conf file:

```
/var/log/syslog /var/log/messages {  
create 0640 root adm  
daily  
rotate 7  
sharedscripts  
postrotate  
/bin/kill -HUP 'cat /var/run/syslogd.pid 2>/dev/null' 2>/dev/null || true  
endscript  
}
```

Include all the files listed by /usr/sbin/syslogd-listfiles and remove /etc/cron.daily/sysklogd. This uses the same rotation schedule as /etc/cront.daily/sysklogd. Feel free to alter it to better suit your system. Make sure that logrotate is called by cron at least as often as you wish to rotate files.

Logrotate has a number of other directives that affect how log files are processed. Of particular interest are directives that affect how files are compressed and that e-mail files before removal. There are also options to use for testing and debugging logrotate. For further information on these and other options, check out the logrotate man page logrotate, which has descriptions of all the directives and other examples.

Centralized Logging

- Protects against log wiping
- Denial of service possibility
- Needs a lot of disks for large environments
- One machine holds a lot of sensitive information
- Easy to search and scan

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Centralized Logging

Introduction

In any environment that has multiple machines, centralized logging should be examined. Not only does it greatly simplify the life of the system administrator, it has the added security benefit of preventing malicious attackers from clearing their tracks behind them (unless they compromise the log server, too).

Protects Against Log Wiping

The main advantage to centralized logging is that it makes it difficult for a remote attack to wipe or otherwise corrupt the system logs. Any logs generated from their attacks will be sent immediately to another machine, which will store the data. Assuming the syslog server does not in turn get hacked, the information will remain there to be discovered by the system administrator and can be used in the recovery process.

Denial of Service Possibility

One key vulnerability of this design is that it makes it possible to either cause the syslog client machines to send so much log data it overwhelms the central syslog server, or an attacker can send false messages to the central syslog server directly to clutter up the logs. This means that a central syslog server should be configured to receive system logs from only those machines that will be legitimately sending logs to it and it should be protected by a firewall from all other machines.

Need Lots of Disks

The only protection against a syslog client that is sending lots of information is to have a syslog server that has lots of disks to store that information. Logs can be rotated quicker if needed, and several partitions can be used for logging to segment the risk of bringing down all the logging in an environment. There are some built-in mechanisms to aggregate repeated messages. That is, syslog messages will show strings such as, "last message repeated X times." In the end, though, the only protection against a malicious user causing logs to fill up is to use a firewall to block the attacker from reaching the environment, rotate logs quicker, or have enough disks to weather the attack.

One Machine Has Lots of Sensitive Information

The consequence of having centralized system logs is that all the logs are now residing in one place. Many logs will contain information that may provide valuable clues as to what is important in an environment. Some logs may even contain privileged information. With this information all in one place, that machine now becomes a valuable machine to target. The central syslog server becomes a machine that is critical to protect. It should only run the system logging service to receive messages and have an SSH daemon that will accept only a limited amount of logins from a small set of IP addresses belonging to administrators.

Easy to Search and Scan

With all the system logs being in one place, it makes it easy to have various log alerting programs (logwatch, logsurfer, swatch) to run in only one place and send only one set of alerts for all the events in an environment. For instance, some log-scanning programs will scan log files and send daily reports of the notable events on a system. Such a program would do so for each machine it is installed on to give a picture of the entire environment. Running the program on a central log server, however, will generate only one single report for the entire environment and all the systems. To track down problems, only one machine needs to be accessed to find error logs, even if the problem spans multiple machines (such as following an e-mail as it travels through an environment).

Security Issues with Syslog

- Denial of service:
 - Anybody can spam your syslog port
 - Overwhelm daemon, fill up logging area
 - Block access to 514/udp
- No authentication of messages
- Can be triggered via other apps and without direct system access by an attacker

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Security Issues with Syslog

As mentioned earlier, one of the biggest problems with syslog is that it will accept messages over the network from any host that is able to reach the syslog port (UDP port 514) on the local system. This means that an attacker can continuously send a large volume of messages at the syslog daemon non-stop until either the daemon is overwhelmed and cannot log any other messages or until the partition where the logging is happening fills up. Make sure that network firewalls block outside access to UDP port 514. If the vendor provides a syslogd option to prevent the daemon from listening on UDP port 514 entirely, then make use of that option in addition to employing network-based firewalls.

The syslog daemon has a history of buffer overflow problems. One fact that makes syslog buffer overflow issues so dangerous is that an attacker need not have direct network access to the system whose syslog daemon is being compromised. For example, Sendmail logs a plethora of data via syslog. There actually was an exploit where the attacker sent a specially crafted e-mail message to a remote mail server behind an organization's firewall. When Sendmail attempted to deliver the message, it would log information about that message to syslog. Because of the way the message was constructed, the logging data from Sendmail actually triggered a buffer overflow in the syslog daemon! It is critical to keep up-to-date on vendor patches for syslogd to help avoid exposure to these kinds of buffer overflow issues.

syslog-NG

- Replacement to syslog
- Additional filtering
- Sends data with TCP
- Can Support Microsoft Windows
- R-syslog also an option

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

syslog-NG

Introduction

There are some limitations to syslog that have led to the development of other system logging packages. Syslog Next Generation (or syslog-NG) was developed to add additional security to remote system logging and provide for additional filtering options for the log files. It is also designed to work with a wide variety of operating systems, including Microsoft Windows, which allows it to be the one logging solution throughout a heterogeneous environment.

Replaces Syslog

The syslog-NG application is a drop-in replacement for the traditional syslog that is included with Linux and Unix systems.[1] Many Linux variants will include a syslog-NG package in their distribution, but do not install it by default. For instance, in Fedora, an administrator can type yum install syslog-ng from the command line and it will install the package for you. To switch from syslog to syslog-NG, simply run chkconfig --del syslog and chkconfig --add syslog-ng. Then, stop the syslog daemon and start the syslog-ng daemon. The configuration for syslog-NG resides in /etc/syslog-ng and uses the same facilities and priorities as the default syslog daemon. It is possible to use a variety of syslog proxy servers to collect data and forward to a central loghost as well.[2]

Additional Filtering

In addition to the ability to filter by facility and priority, syslog-NG allows an administrator to filter by hostname and by the actual text of the log message using regular expressions.[3] It does this by first creating destinations (usually files), creating filters based on facility, priority, hostname, and the presence (or absence) of a string, and then it creates logging rules based on those filters and destinations. For example:

```
# This will create a destination with the alias of "messages" that will send data to /var/log/messages
destination messages { file("/var/log/messages"); };
```

```
# This will create a filter called "f_messages" that will contain a message between info and warn priorities,  
except those generated under the auth, authpriv, mail and news facilities  
filter f_messages { level(info..warn) and not facility(auth, authpriv, mail, news); };  
  
# This will log all messages captured by the "f_messages" filter above and send it to the destination "messages"  
log { source(src); filter(f_messages); destination(messages); };
```

It is important to note that syslog-NG cannot reuse the syslog.conf file from the traditional syslog. The main drawback of syslog-NG is its complexity in configuration.

Sends Data over TCP

When the traditional syslog was conceived, network links were not high-speed and not necessarily high quality. Using UDP allows the system logs to be sent out without having to worry about verifying receipt or dealing with transmission errors. Twenty years ago, this approach made sense. However, in modern environments with modern equipment, the load of dealing with TCP is trivial compared to the overall firepower available to the machine. Syslog-NG made the move to use TCP, so receipt of data can be verified so that it is guaranteed to make it to the remote syslog server. Although this comes at a performance price, it does ensure that important logs will not get lost in transit. In addition, sending the data over TCP allows the use of encryption with stunnel. Normally, syslog traffic is sent in the clear over a network, meaning anyone with a sniffer can read the logs. Using stunnel will allow for tunneling syslog data over an encrypted channel.

Can Support Windows Machines

One of the neat features of syslog-NG is that it can be installed on Windows machines and plugged into a syslog-NG environment. It uses the same central log server as every other machine does. This allows for the ability to correlate events across Linux and Windows environments by looking in the same place.

References

- [1] - <http://www.balabit.com/network-security/syslog-ng/> Syslog-NG homepage
- [2] - <http://sial.org/howto/logging/syslog-ng/> Logging with Syslog-NG
- [3] - <http://linux.cudeso.be/linuxdoc/syslog-ng.php> Syslog-NG examples

Summary

- Understand general Unix logging to identify activity (normal and nefarious)

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Summary

In this section, we discussed the logging mechanism of Unix systems and identified the most common logs and how they can be used to ensure security of the system is maintained through monitoring, detection, and response.

In summary, logging is good, and more logging is always better. Remember the point is to log as much as possible in order to have a better chance of attackers failing to cover all of their tracks. Small clues, such as a 75-cent accounting error, may add up when detecting attacks on local systems.

Logging to multiple destinations often is a good idea. Secure, centralized logging provides a backup for data that may have been deleted locally from the compromised system. Also, centralized logging can help spot patterns, such as network mapping attacks or concerted attacks against many systems in an enterprise, which would not be detected by just looking at the log files on a single machine.

Syslog can transmit log messages to another system over the network and be used to relay messages to a remote log server. However, these messages are received and stored by the remote system without any authentication, so an attacker can overwhelm network log servers without too much trouble. If possible, close off the syslog port (UDP port 514) on systems that are not log servers, and make sure firewalls limit access on this port, particularly from external systems.

System accounting, process accounting, and kernel-level auditing all provide valuable data about what's happening on your systems. However, properly configuring the systems to capture this data, and then doing something useful with the data once you have it requires extra administrative effort.

Module 33: Securing Linux/Unix

PATCH MANAGEMENT

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Module 33: Securing Linux/Unix

PATCH MANAGEMENT

This section intentionally left blank.

Objectives

- Understand good patch management and limitations

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Objectives

In this section, we discuss the basics of good patch-management applications.

Why Patch

- New vulnerabilities are being discovered every day
- Vendors are releasing patches to fix these vulnerabilities
- Unpatched systems are still the major reason systems get compromised

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Why Patch

An important item in keeping your system secure is to maintain the system patch level at all times. The top reason systems get compromised is still unpatched systems.

Be Careful

- Any new patch may impact the way your current system/application functions
 - Applications may stop working entirely or exhibit new and unexpected behaviors after the change
 - Some patches require a reboot
- Test patches on nonproduction systems first before distributing the patches widely

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Be Careful

The best way to ensure your system is always up to date is to use an automated patching program. Although this is ideal for end user systems, it may not be good for production systems. There is no way to know how patches will effect your system, especially third-party software. The best way to prevent this is to have an automated patching program announce the updates but not install them. This will allow you to test and evaluate the patch prior to deployment.

Finding Out About Patches

- Automated update:
 - Most OSes come with an automated updating system:
 - Yellow Dog Updates Manger (yum)
 - Synaptic
 - Red Hat Update Agent
 - On Production systems, it is not a good idea to auto update:
 - The best option is to have the update manger inform there are updates
- Vendor Web/FTP site:
 - You must remember to visit the site often
- Mail list:
 - Have a vendor or third-party (such as BUGTRAQ) e-mail update information

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Finding Out About Patches

Automated updates are the easiest way to keep up-to-date on system patches, but they are not the only way or best way. Other ways include visiting vendor web sites and subscribing to vendor mailing lists. Also, not all automated updates will ensure your third-party software is up-to-date.

Using apt

- Update tool for Debian-based systems, including Knoppix and Ubuntu
- Debian package files are normally named with a .deb extension:
 - apt-get update
 - Update local DB of available packages
 - apt-show-versions –u
 - Check and see which packages need updating
 - apt-get upgrade
 - Download and install upgrades

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Using apt

apt is a Debian-based patching tool that allows for automated patching of Debian-based Unix systems.

Note: Although some Linux distributions use the .deb format for packages, the RPM format used by Red Hat seems to be more popular.

apt

– **apt-show-versions –a <pkgname>**

- Displays all available versions of a given package

– **apt-get install <pkgname>**

- Installs new software

– **apt-get remove <pkgname>**

- Removes software

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

apt

This slide depicts the basic command options and usages for apt.

Installing Programs with yum

- Is it installed already?
`rpm -q gkrellm`
- Install it:
`yum install gkrellm`
- Check for and install patches:
`yum check-update`
`yum update`
 - A lot of bandwidth; might be worth doing later

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Installing Programs with yum

yum is the smart software installation tool for Centos and other Linux distributions. When asked to install or upgrade a package, yum knows how to identify and install any supporting packages. For example, if you install a new spelling check program, yum would know how to find and install the needed dictionary. Before yum, the administrator was often given an error message about the missing dictionary that didn't even list which package contained the needed file.

In this example, we're installing a system monitoring tool called gkrellm. First, let's find out if the program is already installed. Whenever an rpm package is installed on the system, a record is kept in a database. The `rpm -q gkrellm` command asks this database if the package is installed and either gives you package details if it is, or an error if it's not.

To install it, run `yum install gkrellm`. It may take a minute or two if your vm has to pull down some information files, but it should be relatively quick and painless.

yum is also smart enough to look for patches (aka updated software) for all the rpm packages installed on your system. `yum check-update` will give you a list of what needs to be updated. `yum update` will actually pull down the updates and install them along with any needed support files and libraries.

Patching a virtual machine that was made a few months back tends to pull down a lot of updates, using a lot of bandwidth in the process. This might be a better task to do outside of class.

GUI Tools

- Tools that install and update packages by the use of rpm or apt in a *graphical user interface (GUI)*:
 - yum/PackageKit
 - Use rpm to install, upgrade, remove packages
 - synaptic/ QWinApt
 - Use apt to install, upgrade, remove packages

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

GUI Tools

There are other package managers available. Two popular ones are yum and synaptic. Yum stands for yellow dog update manager, and synaptic is a graphical frontend for apt. Red Hat also has another tool called up-to-date.

Other Operating Systems

- Solaris
 - Utilities are pkgadd/pkgrm/pkgchk.
- HP-UX
 - Use swinstall/swremove/swverify.
- BSD
 - Use pkg_add/pkg_delete.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Other Operating Systems

Most operating systems have their own package managers.

Summary

- Understand good patch management and limitations
- One of the most common methods of attack is an un-patched system
- Because Unix systems typically run on high-end hardware, there is a higher chance they will not be patched

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Summary

In this section, we discussed the basics of good patch management applications.

Module 34: Securing Linux/Unix

SECURITY ENHANCEMENT UTILITIES

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Module 34: Securing Linux/Unix SECURITY ENHANCEMENT UTILITIES

This section intentionally left blank.

Objectives

- Understand integrity checkers
- Understand host-based firewalls and how they are managed to provide security
- Understand other tools for increasing security
- Understand security enhancement applications for Unix

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Objectives

In this section, we discuss some security enhancement utilities, capabilities, and patch management applications.

Tripwire®

- Intrusion detection through integrity checking
- Two versions:
 - Commercial: www.tripwire.com
 - Open Source: www.tripwire.org
- Creates a “secure” database of file and directory attributes
- Can include MD5 signatures for verification

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Tripwire®

Integrity checkers are a valuable tool that allow administrators to take a check-and-balance look at their systems and be presented with a snapshot of any modifications that may have occurred.

Tripwire® is both a commercial and open source integrity tool. It creates a digital snapshot of files and/or directories and places. This is a portable database. The database should be maintained offsite and it should be secured. Should an incident occur, or if an organization would like to audit systems, Tripwire® will provide a quick glance at any changes made to systems since the snapshot was taken. This can allow responders to immediately highlight any significant changes made to a system.

Tripwire® allows for the inclusion of both MD5 signatures for verification of data.

Tripwire® Common Commands

Create a Tripwire® database:

```
tripwire -m i -v ## this runs tripwire in init phase with verbose output
```

Check a system against the Tripwire® database:

```
tripwire -m c -v ## this runs tripwire in check phase with verbose output
```

Read a Tripwire® report:

```
twprint -m r --twrfile <filename>
```

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Tripwire® Common Commands

The most common commands for Tripwire® are the most basic.

Create the digital snapshot or database once all configurations have been made and to establish the baseline for future modification-tracking opportunities.

Check the system against a previous snapshot to highlight any changes to the system configuration files or access issues.

Report or view reports on completed integrity-checking actions.

Updating and maintaining the integrity database are as important as storing and protecting the snapshot offsite. If required changes are made to a system, an update to the integrity database must be performed. However, it is recommended that prior to any updates, an integrity check validation be performed to ensure the system has not been compromised and that access levels are okay. Then, an update to the system is performed, followed by an update to the integrity database to establish a new baseline snapshot.

Again, the steps are:

1. Create and update a system to the desired operational state.
2. Create an integrity snapshot to be the baseline for future verifications.
3. Occasionally check systems against their baselines to watch for modifications and access issues.
4. Perform an integrity check prior to updating a system.
5. Update the system with desired patches.
6. Update the integrity database and store offsite in a secure location.

IP Tables

- Built-in, host-based firewall for Linux:
 - Very powerful and customizable
- Many free scripts and GUIs available for simplifying configuration and maintenance
- Stateful firewall with NAT capability

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

IP Tables

The Linux kernel includes IP Tables, which is a built-in firewall with capabilities that are equivalent to modern, commercial firewall products, though IP Tables on COTS hardware may not be able to handle the same packet throughput as a specialized firewall hardware platform. In fact, many sites use a Linux machine running IP Tables instead of a commercial firewall product. All you need is Linux running on a hardware platform with multiple network interface cards and somebody who understands how to configure IP Tables to enforce the policy you want. Creating these kinds of quick-and-dirty network-layer firewalls was the original design impetus behind the IP Tables project.

Over the past few years, though, sites have been looking at IP Tables more as a host-based firewall to protect individual systems. Because IP Tables is a complete packet inspection system, you can implement much more powerful filters than just the simple kind of source address filtering provided by TCP Wrappers. The problem is that all of that powerful functionality comes with a lot of complexity. Learning how to write IP Tables filters to do what you want requires not only learning a lot about how firewalls are configured, but also overcoming the sometimes bizarre and confusing IP Tables rule language itself.

Just to give you a quick impression of the IP Tables command-line interface for building firewall rules, here's a series of IP Tables commands for creating a simple policy that allows incoming SSH traffic on port 22/tcp and allows all outgoing TCP, UDP, and ICMP traffic:

```
# Flush any existing rules
iptables -F

# Set our default policy to drop all packets
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

```
# Don't interfere with loopback traffic
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

# Inbound Rules: Allow SSH (22/tcp) and packets from
# sessions that we have initiated from this system
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -p tcp -m state \
--state ESTABLISHED -j ACCEPT
iptables -A INPUT -p udp -m state \
--state ESTABLISHED -j ACCEPT
iptables -A INPUT -p icmp -m state \
--state ESTABLISHED -j ACCEPT

# Outbound Rules: Any TCP, UDP, ICMP is OK
iptables -A OUTPUT -p tcp -m state \
--state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p udp -m state \
--state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p icmp -m state \
--state NEW,ESTABLISHED -j ACCEPT
```

You might be able to use the previous rules on a very simply configured system, such as a personal laptop, but for any real application, your firewall rule sets would have to be much more complicated than those shown above. You can see how the above command-line syntax would tend to discourage users and administrators from making full use of IP Tables.

The good news is that simple GUIs have been developed to allow even novice users to take advantage of IP Tables functionality. For example, Red Hat-based systems have a simple GUI widget that you can access via the *Applications... System Settings... Security Level...* menu choice. This widget enables you to turn the firewall on the system on and off and to select from a few different services that you want to allow outsiders to be able to access on your machine.

IP Tables Rules

```
iptables -A INPUT -p tcp --dport ssh -j  
ACCEPT
```

1. Append this rule to the input chain (-A INPUT), so we look at incoming traffic
2. Check to see whether it is TCP (-p tcp)
3. If so, check to see whether the input goes to the SSH port (--dport ssh)
4. If so, accept the input (-j ACCEPT)

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Rules

The rules in chains are what make IP Tables work. All packets go down the chain until they find a matching rule to follow, which may be another chain or accept/deny.

Basic IP Tables Rules Options

-A: Append this rule to a rule chain. Valid chains for what we're doing are INPUT, FORWARD, and OUTPUT, but we mostly deal with INPUT in this tutorial, which affects only incoming traffic.

-L: List the current filter rules.

-m state: Allow filter rules to match based on connection state. Permits the use of the **--state** option.

--state: Define the list of states for the rule to match on. Valid states are:

NEW: The connection has not yet been seen.

RELATED: The connection is new, but is related to another connection already permitted.

ESTABLISHED: The connection is already established.

INVALID: The traffic couldn't be identified for some reason.

-m limit: Require the rule to match only a limited number of times. Allows the use of the **--limit** option. Useful for limiting logging rules.

--limit: The maximum matching rate, given as a number followed by /second, /minute, /hour, or /day depending on how often you want the rule to match. If this option is not used and **-m limit** is used, the default is 3/hour.

-p: The connection protocol used.

--dport: The destination port(s) required for this rule. A single port may be given, or a range may be given as start:end, which will match all ports from start to end, inclusive.

IP Tables: Setting Up a Service Network

```
iptables -A FORWARD ... -j ACCEPT around each:  
-m state --state ESTABLISHED,RELATED  
-d web -p tcp --dport 80  
-d web -p tcp --dport 443  
-d mail -p tcp --dport 25  
-s mail -p tcp --dport 25  
-d dns -p udp --dport 53  
-d dns -p tcp --dport 53  
-s dns -p udp --dport 53  
-s dns -p tcp --dport 53  
-s admin -d 10.2.3.0/24 -p tcp --dport 22  
iptables -A FORWARD -j DROP
```

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

IP Tables: Setting Up a Service Network

Here's a sample set of firewall rules for setting up a service network. This network has a web server, a dns server, and a mail server, each with its own line in the firewall's /etc/hosts file (such as 192.168.1.7 web).

In IP Tables, the ESTABLISHED state matches packets that are part of an existing conversation; the second, third, and further packets in a TCP or UDP conversation. RELATED packets are those that have a connection to an existing conversation; these include FTP data channels and ICMP error messages, for example. We allow these first because this one rule will match the vast majority of the traffic in the firewall. Placing it first makes the firewall much more efficient.

For the rest of the firewall, then, we get to focus just on the initial packet of a connection. Both --dport 80 and --dport 443 let HTTP and HTTPS traffic in to our web server. For the mail server, we need to let SMTP mail both in and out, so we have rules for SMTP to (-d mail) and from -s mail our mail server. DNS also needs rules for traffic both to and from the machine named DNS. Because DNS traffic is carried on both TCP and UDP, we need to set up 4 rules.

We'll manage the systems over ssh, so we allow ssh traffic (--dport 22) from a single machine called admin to the entire subnet, 10.2.3.0/24. Finally, we block any other traffic.

This isn't a complete firewall, but does demonstrate the standard parts of setting up the service network.

chroot()

- Applications may call chroot() to isolate themselves to a particular directory
- If an attacker compromises an application, he will have only limited access to system
- Some apps have built-in chroot(): TFTP, (anonymous) FTP, BIND, and SSH
- For other apps such as Apache, use the chroot wrapper program

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

What Is chroot()?

The Unix chroot() restriction is an application isolation feature that must be enabled on an application-by-application basis.

An application developer may choose to have a program invoke chroot(), typically early in the setup and configuration phase of the application before the application starts accepting requests from external users. When an application calls chroot(), it is basically specifying to the OS kernel a directory in the file system where the application wants to isolate itself. From that point forward, the kernel enforces this isolation and doesn't let the application "see" any of the file system outside of the directory that the application specified when making the chroot() call. From the application's perspective it is as if this directory and all the subdirectories and files below it are now the only files and directories on the system (the system call is named chroot() because it effectively "changes the root" of the file system from the application's perspective).

The advantage from a security perspective is that if a vulnerability is discovered and exploited in the application, then the attacker can only access the parts of the file system where the application has chroot()ed itself too. This prevents the attacker from modifying other configuration files and binaries in the main operating system directories and compromising the rest of the system. In fact, the attacker's exploit may fail completely because it relies on other operating system binaries that are not present in the chroot() directory used by the application. This is particularly useful for applications like BIND and SSH, which are often Internet-facing applications and are complex enough that it's likely there are going to be more remote exploits discovered over time. Running these applications with the chroot() restriction helps mitigate the potential damage from future security events.

Aside from mitigating unknown exploits, another reason to employ chroot() is to simply restrict the amount of access a given application may have to the file system. For example, TFTP allows unauthenticated file transfers from a system, so administrators would like to restrict TFTP to only serving files from certain authorized directories. This is why modern TFTP daemons chroot() themselves to a directory like /tftpboot so that they will

only serve files from this area. Similarly, anonymous FTP servers are configured to automatically invoke chroot() so that they only provide access to software archives specifically configured by the system administrator.

However, it's important to realize that the decision to use chroot() is left to the individual application developer. You might want to chroot() your Apache web server to help protect yourself from directory traversal attacks that allow outsiders to escape from your standard HTML document tree and access other files on the system. Unfortunately, the Apache developers have not included chroot() functionality in the application. For these sorts of applications, the OS includes a "wrapper" type program called chroot. In the case of Apache, you would run the chroot "wrapper" program and tell it the directory where you want the chroot() to happen and specify the normal command-line invocation for Apache. The chroot program will perform the chroot() system call and then start the Apache daemon in this isolated environment.

The Problem with chroot()

- Have to discover and copy all application dependencies into chroot() directory:
 - Helper programs and shared libraries
 - Application and system configuration files
 - Device files (tricky!)
- Look for pre-configured directories provided with OS (BIND, FTP)
- Applications with complex dependencies may be impossible to chroot()

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

The Problem with chroot()

So why doesn't every application run under a chroot() restriction by default? Wouldn't that sort of rigid compartmentalization improve the overall security of the system? The answer is yes. This would improve security, but the difficulty is that there is significant administrative effort involved in configuring the directory structure where the application will chroot() itself to.

Remember that as far as the application is concerned, the chroot() directory and the directories below it are the entire file system. This means if the application needs to run another program, read a configuration file, or access a system device, then a copy of that program, configuration file, or device must be provided underneath the chroot()'ed directory structure in the same relative file location that the application expects. For example, if a BIND name server runs chroot()'ed to /var/named/chroot and it wants to look at its /etc/named.conf configuration file, the administrator has to create /var/named/chroot/etc/named.conf. If that same program wants to access the /dev/null system device, then the administrator must create a copy of this device called /var/named/chroot/dev/null, and so on.

What makes this more difficult is that there is not a tool you can run on a given application to find all of the external dependencies it might require. Instead, you end up going through a process of repeated failures where you attempt to run the application under chroot() restrictions, wait for the application to break due to some external dependency, figure out from the error messages what that dependency is, and then add copies of the required file(s), program(s), and so on in the chroot() area. Eventually, you achieve a configuration where the application functions properly. However, then you have an ongoing maintenance issue where patches and other software updates also need to be reflected in the copies of the programs, libraries, and configuration files that have been copied to the chroot() areas on your system.

The good news is that vendors are starting to provide pre-configured chroot() areas for common applications. For example, the TFTP daemon on every modern Unix system is already set up to properly run chroot()'ed in /tftpboot. Red Hat provides a pre-configured /var/named/chroot area for running BIND safely, although BIND under Red Hat is not configured to use this setup by default.

Ultimately, though, an application may have so many external dependencies that creating a working chroot() directory tree for the application means basically making an almost complete copy of the operating system under the chroot() directory. In these cases, just isolating the application on its own server is almost equivalent. Virtual machine technology, such as VMWare or Xen, is another mechanism you can employ to provide isolated "sandbox" type systems for specific applications.

Chkrootkit

- Unix equivalent of antivirus:
 - Looks for rootkits, sniffers, deleted logs, trojans, and kernel modules
- Quiet mode's only problems:
`chkrootkit -q`

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Chkrootkit

chkrootkit is a free malware-scanning tool for the various Unix flavors. It runs on Linux, FreeBSD, OpenBSD, NetBSD, Solaris, HP-UX, Tru64, BSDI, and Mac OS/X (which is FreeBSD-based).

Windows, with its market penetration and far more common malware, requires frequent updates to antimalware packages. These have thousands of signatures and need 24-hour coverage from paid employees to keep up with the new strains of malware. The Unixes aren't immune from malware by any stretch of the imagination, but because there are fewer signatures, these can be covered by a volunteer group. Nelson Murilo and a worldwide team have been maintaining this program since 1997.

To check a system, simply run chkrootkit. This mode is a good way to run it for the first time; it tells you what it's checking for and what it's found and not found. Feeding the output into less (`chkrootkit -q | less`) lets you scroll up and down through the checks.

For regular checks, especially checks from cron, it's better to run it in quiet mode (`chkrootkit -q`) where it shows only potential problems.

CIS Hardening Guides

- How do I harden my platform?
- Consensus guides with steps to make system secure
- Hardening tools
- Scoring tools
- Free for your use

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

CIS Hardening Guides

Wouldn't it be great if we could get the world's experts on securing Windows (/Linux/IOS/VMware/.....) all together in one room, get them to come up with the steps to secure the platform, and publish that list? Could we publish the list for anyone to use at no cost?

The Center for Internet Security is a volunteer project that does exactly that. Security gurus work together on a mailing list to come up with the steps they think should be followed in securing that platform.

The resulting document will include the steps they can largely agree on. The guide needs to be conservative; the suggested changes should not cause significant problems in almost all situations. For that reason, the more aggressive security settings that run a risk of disabling needed features tend to be left out. In the end, that's a good thing; people feel they can trust the suggestions.

For some of the platforms, CIS also has published hardening tools that will make some or all of the recommended changes automatically. They may also have scoring tools, programs that rate your system on how secure it is. After you've made some lockdown changes, the scoring tool can be run again with a hope that the number has gone up. The absolute number doesn't have a direct meaning, but it does allow a relative comparison between systems.

All of the guides, hardening tools, and scoring tools are free for your use with nothing more than registration.

Scoring Tools

Description	Items		Score	
	Passed	Failed	Actual	Max
1 Patches, Packages and Initial Lockdown	2	1	7.407	11.111
2 Minimize xinetd network services	6	2	8.333	11.111
3 Minimize boot services	11	10	5.320	11.111
4 Kernel Tuning/Network Parameter Modifications	0	2	0.000	11.111
5 Logging	2	2	5.556	11.111
6 File/Directory Permissions/Access	2	7	2.469	11.111
7 System Access, Authentication, and Authorization	2	9	2.020	11.111
8 User Accounts and Environment	5	7	4.630	11.111
9 Warning Banners	0	3	0.000	11.111
9.1 Reboot	0	0	0.000	0.000
10 Anti-Virus Consideration	0	0	0.000	0.000
11 Remove Backup Files	0	0	0.000	0.000
Overall Score:		30	43	36.240

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Scoring Tools

This is the summary from a sample CIS benchmark report. Because Centos Linux is based on Red Hat Linux, we used the Red Hat benchmark to create it.

The installation takes a little prep work. The current Red Hat benchmark requires Sun's Java Runtime Environment, and specifically version 1.5 (5.0). Don't get anything older or newer. Get the JRE from <http://java.sun.com/j2se/1.5.0/download.jsp>. Otherwise, the installation instructions are relatively straightforward.

Below the summary are category sections, with specific details on each potential issue. Each test provides a passed/failed/not tested, as well as more details about the test and what should be done to fix the issue.

The final score is not an absolute number, but a score that can be vaguely compared to other CIS Benchmark scored systems. It is also intended to show improvement. After you fix some of the issues mentioned, rerun the benchmark to hopefully see an improved score.

For more details and benchmark downloads, see the Center for Internet Security at <http://www.cisecurity.org/>.

Bastille Linux

- Hardening program:
 - Reports on how secure the system is
 - Shows security issues:
 - Educates admin
 - Optionally fixes issues:
 - Changes can be reverted
 - Linux, HP-UX, and Mac OS/X

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Bastille Linux

Bastille goes one step beyond the scoring tools. Instead of just reporting on what might be security issues, Bastille will explain the issues and optionally fix them.

This perl script inspects the system for common security mistakes. With each one, Bastille explains what the issue is, why it should be fixed, and what side effects the fix might have. This is a major improvement over a script that simply makes changes to the system—the administrator can make an informed decision about what to change. Bastille keeps track of these changes and can revert them later if needed. If you choose to do this, it's better to do this sooner than later. Reverting the changes can also remove any manual changes to these configuration files.

Bastille runs on multiple Unixes, including Mac OS/X, which is based on FreeBSD.

Security Enhancement Applications

- SELinux
 - Enhance the default DAC (discretionary access control) security of a Unix system with the inclusion and management of a MAC (mandatory access control) security effort
- AppArmor
 - Is SELinux the only answer? No.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Security Enhancement Applications

The next few slides address two of the most common security applications designed to enhance the already present DAC (discretionary access control) capability of Unix operating systems.

Security-Enhanced Linux (SELinux)

- SELinux is a Linux security enhancement feature
- It uses security-based policies
- It is based on the U.S. Department of Defense style mandatory access controls (MAC)
- It uses Linux Security Modules (LSM) in the Linux kernel

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Security-Enhanced Linux (SELinux)

SELinux uses Mandatory Access Controls (MAC) to allow administrators the ability to control all interactions of software on the system. The security model is based on least privilege and starts with users having no rights at all. Administrators must grant users access through the use of security policies.

Original Linux distributions follow the concept of Discretionary Access Control (DAC). DAC allows users full security access over their installed and owned applications. This can lead to security risks to systems should a user's account be compromised.

SELinux is not designed to replace existing security for Linux. It is designed only to enhance it.

References

http://en.wikipedia.org/wiki/Mandatory_access_control

<http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/selinux-guide/selg-preface-0011.html>

DAC and SELinux Policy

- Discretionary Access Control (DAC)

- Deny access

- Will not use SELinux access on object

- Allow or default

- SELinux access policy applies

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

DAC and SELinux Policy

Most Linux variants use a security measure called Discretionary Access Control (DAC). This allows the sysadmin and all users to manage the security of files they own or manage. There is one flaw in this method and that is any exploit that allows access to the system as any potential user may be able to elevate themselves or processes they own and defeat or circumvent security. SELinux takes this security control and enhances it with the capability to afford Mandatory Access Controls (MAC) along with the default DAC. What does this mean? If the system is using DAC and has a default DENY access set up, there is no need to utilize any other security, and SELinux is not a player. However, if the default is allow and you desire more finite control over the security permissions and context access of your files, directories, and processes, then SELinux is the answer.

MLS/MCS

- Multi-level security
 - Ranges written as:
 - Low-level and high-level (if levels differ)
 - Lowlevel (if levels are identical, i.e s0-s0 = s0)
- Multi-category security:
 - s0:c0=CompanyConfidential
 - s0:c1=PatientRecord
 - s0:c2=Unclassified
 - s0:c3=TopSecret
 - s0:c1,c3=CompanyConfidentialRedhat

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

MLS/MCS

Labels are an attribute of MLS and Multi-Category Security (MCS).

Each level is a sensitivity-category pair, with categories being optional:

- When using categories, the level is written as sensitivity:category-set.
- When not using categories, the level is written as sensitivity.

Categories can be abbreviated. For example, c0.c3 is the same as c0,c1,c2,c3.

Fedora 10 enforces MCS with targeted policy enforcement.

In MCS, there is one sensitivity, s0.

MCS, in Fedora 10, supports 1,024 different categories: c0 through to c1023.

s0-s0:c0.c1023 is sensitivity s0 and authorized for all categories.

MLS enforces the *Bell-LaPadula* Mandatory Access Model, and it is used in Labeled Security Protection Profile (LSPP) environments.

Other Approaches

- AppArmor

- It is an alternative to SELinux and also uses the LSM framework
- It is interchangeable with SELinux
- SELinux was viewed as too complex for typical users to manage
- It includes a MAC model fully configurable as well as a learning mode
- It is available on SUSE, OpenSUSE, and Ubuntu

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Other Approaches

SELinux is not the only MAC-capable framework on the market. AppArmor has distinguished itself as interchangeable with SELinux and is seen as an easier-to-manage alternative. Understanding the underpinnings of context still remains: Configuration is seen as easier, but it still affords the same level of security.

References

<http://www.ibm.com/developerworks/linux/library/l-selinux/>

Summary

- Understand integrity checkers and how they afford protection
- Understand host-based firewalls and how they are managed to provide security
- Understand security enhancement applications for Unix

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Summary

In this section, we discussed:

- Integrity checkers
- Host-based firewalls using rule-based methods
- Security enhancement applications

Course Summary

- Focus on risks to critical assets
- Core principles:
 - Know thy system
 - Least privilege
 - Defense in depth
 - Prevention is ideal, but detection is a must

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Course Summary

With any operating system, it is critical that it be as secure as possible. However, it is important to remember that there is always a balance between functionality and security. The trick is to understand what the risks are to the critical assets that reside on that computer and reduce those risks to an acceptable level.

The first way of accomplishing this is to give a system the least amount of access it needs to do its job. For a server, this means removing services and turning off ports that are not needed. It is also important to ensure the system is fully patched. In order to do this, you need to know the system and understand what runs on it.

Finally, you should deploy multiple measures of protection to keep the system secure, always remembering that prevention is ideal, but detection is a must.

Appendix: Key Unix Commands

- The following appendix provides details on the key Unix commands you need to know

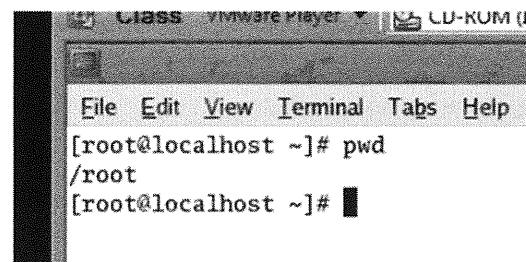
SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Appendix: Key Unix Commands

This section intentionally left blank.

pwd

- Print Working Directory



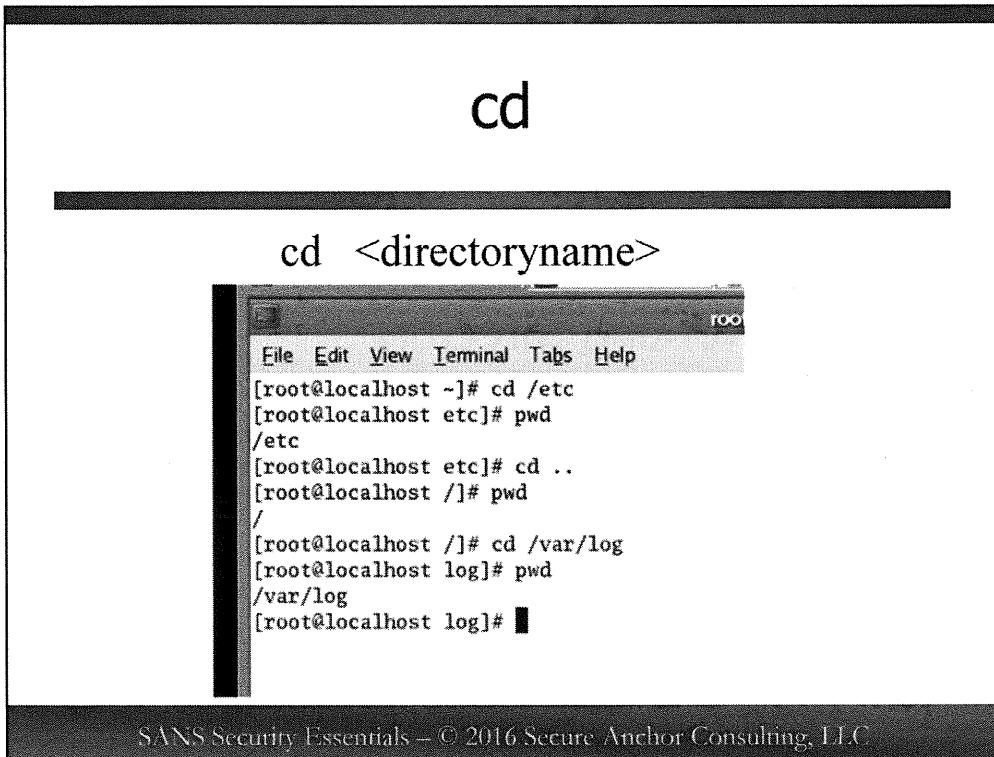
A screenshot of a terminal window titled "CLASS VMWARE Player". The window has a menu bar with "File", "Edit", "View", "Terminal", "Tabs", and "Help". The main area shows the command [root@localhost ~]# pwd followed by the output /root. The prompt [root@localhost ~]# is visible at the bottom.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

pwd

So, you're on the hard drive and you're lost. How do you find out what the full path name is?

The print working directory command (pwd) simply tells the user where he currently sits in the file system.



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

cd

The cd, or change directory command, is used to change location in the file structure.

NAME

cd - change working directory

SYNOPSIS

cd *directory*

DESCRIPTION

Directory is an absolute or relative pathname that becomes the new working directory.

If the *directory* operand does not begin with a slash (/) character, and the first component is not dot (.) or dot-dot (..), an empty string in place of a directory pathname represents the current directory.

If cd is invoked without arguments and the HOME environment variable exists and contains a directory name, that directory becomes the new working directory.

ls

Listing the Contents of a Directory

```
[jsmith@satcom jsmith]$ ls
lg
[jsmith@satcom jsmith]$

[jsmith@satcom jsmith]$ ls -l
total 1
drwxr-xr-x  3 jsmith  jsmith      1024 Feb 18 14:45 lg

[jsmith@satcom jsmith]$ ls -al
total 8
drwxrwxr-x  3 jsmith  jsmith      1024 Feb 22 15:45 .
drwxr-xr-x  30 root    root       1024 Feb 22 15:40 ..
-rw-rw-r--  1 jsmith  jsmith     135 Feb 22 15:45 .bash_history
-rw-r--r--  1 jsmith  jsmith    674 Feb  5 1997 .bashrc
-rw-r--r--  1 jsmith  jsmith    602 Feb  5 1997 .cshrc
-rw-r--r--  1 jsmith  jsmith    116 Feb  5 1997 .login
-rw-r--r--  1 jsmith  jsmith   234 Feb  5 1997 .profile
drwxr-xr-x  3 jsmith  jsmith      1024 Feb 18 14:45 lg
[jsmith@satcom jsmith]$
```

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

ls

The ls command is used to list the contents of the current directory. This is similar to the dir command in the DOS world.

NAME

ls - list directory contents

SYNOPSIS

ls [*OPTION*]... [*FILE*]...

OPTIONS

-a, --all: Do not hide entries starting with.

-l: Use a long listing format.

-q, --hide-control-chars: Print ? instead of nongraphic characters.

DESCRIPTION

List information about the files (the current directory by default).

touch / clear

- Touch can be used to create files or update timestamps. For example:

`touch filename`

- Clear clears the screen of all output

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

touch / clear

The touch command simply creates an empty file. It can also be used to change the timestamps of a file.

NAME

touch - change file timestamps

SYNOPSIS

touch [*OPTION*]... *FILE*...

OPTIONS

-t [[CC]YY]MMDDhhmm[.ss]: Use this instead of current time.

-a: Change only the access time.

-m: Change only the modification time.

-r <file>: Use this file's time instead of current time.

-t [[CC]YY]MMDDhhmm[.ss]: Use instead of current time.

-a: Change only the access time.

DESCRIPTION

Update the access and modification times of each file to the current time.

If FILE does not exist, then a new empty file is made.

The clear command clears the screen of all output, much like the cls command does in DOS.

NAME

clear - clear the terminal screen

SYNOPSIS

clear

DESCRIPTION

The clear command clears your screen.

cat

```
cat <filename>
cat <filename> | more
```

```
[root@localhost log]# cat /var/log/messages |more
Dec 22 11:34:17 localhost syslogd 1.4.1: restart.
Dec 22 11:37:14 localhost init: Trying to re-exec init
Dec 22 13:34:41 localhost shutdown: shutting down for system halt
Dec 22 13:34:42 localhost init: Switching to runlevel: 0
Dec 22 13:34:42 localhost gconfd (root-2150): Received signal 15, shutting down cleanly
Dec 22 13:34:42 localhost gconfd (root-2150): Exiting
Dec 22 13:34:43 localhost gdm(pam_unix)[2036]: session closed for user root
Dec 22 13:34:44 localhost xfs[1763]: terminating
Dec 22 13:34:44 localhost gpm[1734]: *** info [mice.c(1766)]:
Dec 22 13:34:44 localhost gpm[1734]: imps2: Auto-detected intellimouse PS/2
Dec 22 13:34:45 localhost xinetd[1726]: Exiting...
```

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

cat

The cat command is used to concatenate or display files. When given only a single file as an argument, it simply lists the contents to the screen. In order to keep the contents from scrolling off the screen, the output may be piped into the more command. This will allow the output to be displayed one page at a time.

NAME

cat - concatenate files and print on the standard output

SYNOPSIS

cat <Options> <File>

DESCRIPTION

The cat command writes the contents of standard input, or, a file if the filename is specified, to standard output.

mv

```
mv <filename> <newfilename>
```

```
CDE Edit View Terminal Help
[root@localhost ~]# ls -al .bash*
-rw------- 1 root root 5952 Dec 23 18:12 .bash_history
-rw-r--r-- 1 root root 24 Dec 3 2004 .bash_logout
-rw-r--r-- 1 root root 190 Sep 11 2006 .bash_profile
-rw-r--r-- 1 root root 176 Dec 3 2004 .bashrc
[root@localhost ~]# mv .bash_history .bash_history.old
[root@localhost ~]# ls -al .bash*
-rw------- 1 root root 5952 Dec 23 18:12 .bash_history.old
-rw-r--r-- 1 root root 24 Dec 3 2004 .bash_logout
-rw-r--r-- 1 root root 190 Sep 11 2006 .bash_profile
-rw-r--r-- 1 root root 176 Dec 3 2004 .bashrc
[root@localhost ~]#
```

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

mv

The mv, or move command, is used to move a file from one location to another. Because there is no rename function in Unix, this command is also used to rename a file by simply moving it from the old name to the new name.

NAME

mv - move (rename) files

SYNOPSIS

```
mv [OPTION]... SOURCE DEST
mv [OPTION]... SOURCE... DIRECTORY
```

DESCRIPTION

Rename *SOURCE* to *DEST* or move *SOURCE(s)* to *DIRECTORY*.

cp

```
cp <filename> <destination>
```

```
[root@localhost ~]# ls -al .bash*
-rw----- 1 root root 5952 Dec 23 18:12 .bash_history.old
-rw-r--r-- 1 root root 24 Dec 3 2004 .bash_logout
-rw-r--r-- 1 root root 190 Sep 11 2006 .bash_profile
-rw-r--r-- 1 root root 176 Dec 3 2004 .bashrc
[root@localhost ~]# cp .bash_history.old .bash_history
[root@localhost ~]# ls -al .bash*
-rw----- 1 root root 5952 Dec 24 14:36 .bash_history
-rw----- 1 root root 5952 Dec 23 18:12 .bash_history.old
-rw-r--r-- 1 root root 24 Dec 3 2004 .bash_logout
-rw-r--r-- 1 root root 190 Sep 11 2006 .bash_profile
-rw-r--r-- 1 root root 176 Dec 3 2004 .bashrc
[root@localhost ~]#
```

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

cp

The cp, or copy command, is used to copy a file from one location to another. Similar to the move command, the copy command differs only in that two copies of the file exist once the command is executed.

NAME

cp - copy files and directories

SYNOPSIS

```
cp [OPTION]... SOURCE DEST
cp [OPTION]... SOURCE... DIRECTORY
```

DESCRIPTION

Copy SOURCE to DEST or multiple SOURCE(s) to DIRECTORY.

mkdir

mkdir <newdirectoryname>

```
[root@localhost ~]# ls -l
total 100
-rw----- 1 root root 1952 Aug  4 2006 anaconda-ks.cfg
drwxr-xr-x 2 root root 4096 Dec 23 16:50 Desktop
-rw-r--r-- 1 root root 62725 Aug  4 2006 install.log
-rw-r--r-- 1 root root 5468 Aug  4 2006 install.log.syslog
[root@localhost ~]# mkdir new
[root@localhost ~]# ls -l
total 108
-rw----- 1 root root 1952 Aug  4 2006 anaconda-ks.cfg
drwxr-xr-x 2 root root 4096 Dec 23 16:50 Desktop
-rw-r--r-- 1 root root 62725 Aug  4 2006 install.log
-rw-r--r-- 1 root root 5468 Aug  4 2006 install.log.syslog
drwxr-xr-x 2 root root 4096 Dec 24 14:38 new
[root@localhost ~]#
```

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

mkdir

The `mkdir`, or make directory command, is used to create a new directory. The name of the new directory is simply provided as the only argument to the command.

NAME

`mkdir` - make directories

SYNOPSIS

`mkdir [OPTION] DIRECTORY...`

OPTION

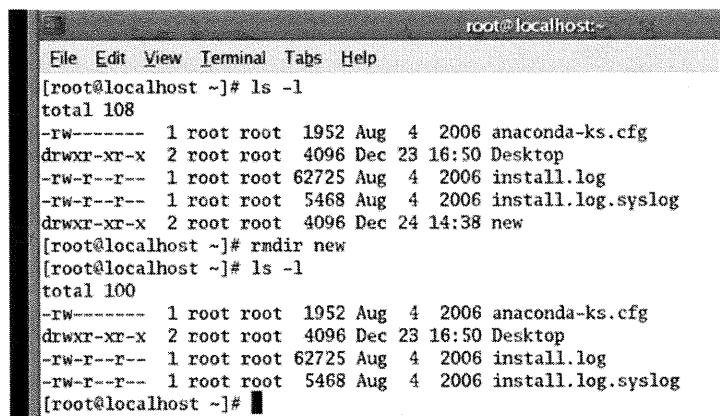
-p, --parents: No error; if existing, make parent directories as needed.

DESCRIPTION

Create the DIRECTORY(ies), if they do not already exist.

rmdir

rmdir <directoryname>



```
root@localhost ~]# ls -l
total 108
-rw----- 1 root root 1952 Aug  4 2006 anaconda-ks.cfg
drwxr-xr-x 2 root root 4096 Dec 23 16:50 Desktop
-rw-r--r-- 1 root root 62725 Aug  4 2006 install.log
-rw-r--r-- 1 root root 5468 Aug  4 2006 install.log.syslog
drwxr-xr-x 2 root root 4096 Dec 24 14:38 new
[root@localhost ~]# rmdir new
[root@localhost ~]# ls -l
total 100
-rw----- 1 root root 1952 Aug  4 2006 anaconda-ks.cfg
drwxr-xr-x 2 root root 4096 Dec 23 16:50 Desktop
-rw-r--r-- 1 root root 62725 Aug  4 2006 install.log
-rw-r--r-- 1 root root 5468 Aug  4 2006 install.log.syslog
[root@localhost ~]#
```

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

rmdir

The rmdir, or remove directory command, deletes an empty directory. It is important to note that this command will not delete a directory if it contains any files or other directories.

NAME

rmdir - remove empty directories

SYNOPSIS

rmdir [OPTION]... DIRECTORY...

DESCRIPTION

Remove the DIRECTORY(ies), if they are empty.

rm

rm <filename>

```

File Edit View Terminal Tabs Help
[root@localhost ~]# ls -al .bash*
-rw----- 1 root root 5952 Dec 24 14:36 .bash_history
-rw----- 1 root root 5952 Dec 23 18:12 .bash_history.old
-rw-r--r-- 1 root root 24 Dec 3 2004 .bash_logout
-rw-r--r-- 1 root root 190 Sep 11 2006 .bash_profile
-rw-r--r-- 1 root root 176 Dec 3 2004 .bashrc
[root@localhost ~]# rm .bash_history.old
rm: remove regular file '.bash_history.old'? y
[root@localhost ~]# ls -al .bash*
-rw----- 1 root root 5952 Dec 24 14:36 .bash_history
-rw-r--r-- 1 root root 24 Dec 3 2004 .bash_logout
-rw-r--r-- 1 root root 190 Sep 11 2006 .bash_profile
-rw-r--r-- 1 root root 176 Dec 3 2004 .bashrc
[root@localhost ~]#

```

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

rm

The rm, or remove command, is used to delete files or directories. Because Unix considers almost everything to be a file, the remove command is able to delete directories that are simply files containing lists of other files. This command, with the proper switch, can be used to delete directories and their contents recursively.

NAME

rm - remove files or directories

SYNOPSIS

rm [OPTION]... FILE...

OPTIONS

-f, --force ignore nonexistent files, never prompt

-i, --interactive prompt before any removal

-r, --recursive remove the contents of directories recursively

DESCRIPTION

The rm command removes each specified file. By default, it does not remove directories.

SU

SU <user>

```
[root@localhost ~]# id  
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel) context=r  
oot:system_r:unconfined_t  
[root@localhost ~]# su pooh  
[pooh@localhost root]$ id  
uid=503(pooh) gid=503(pooh) groups=503(pooh) context=user_u:system_r:unconfined_t  
[pooh@localhost root]$ su tigger  
Password:  
[tigger@localhost root]$ id  
uid=502(tigger) gid=502(tigger) groups=502(tigger) context=user_u:system_r:unconfined_t  
[tigger@localhost root]$
```

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

su

NAME

su - run a shell with substitute user and group IDs

SYNOPSIS

su <user>

DESCRIPTION

The **su** command allows one user to temporarily become another user. It runs a shell with the real and effective user ID, group ID, and supplemental groups of USER. If no USER is given, the default is root, the super-user. The shell run is taken from USER's password entry, or /bin/sh if none is specified there. If USER has a password, su prompts for the password unless run by a user with real user ID 0 (the super-user). su will substitute the new user's \$ENVIRONMENT as well.

sudo (1)

- Pronounced (su "do")
- Allows a system administrator to give certain users (or groups of users) the ability to run some (or all) commands as root while logging all commands and arguments
- Operates on a per-command basis and is not a replacement for the shell. It's features include:
 - The ability to restrict what commands a user may run on a per-host basis
 - Does copious logging of each command, providing a clear audit trail of who did what

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

sudo (1)

On some flavors of Unix, the root account password is locked, such as in Ubuntu. This means that you cannot log in as root directly or use the su: command to become the root user. However, since the root account exists, it is still possible to run programs with root-level privileges.

sudo adds a log entry of the command(s) run (in /var/log/auth.log). It allows easy transfer for admin rights, in a short term or long term period, by adding and removing users from groups, while not compromising the root account. Every cracker trying to *brute-force* their way into your box will know it has an account named root and will try that first. What they don't know is what the usernames of your other users are. Because the root account password is locked, this attack becomes essentially meaningless, give there isn't a password to crack or guess in the first place.

sudo (2)

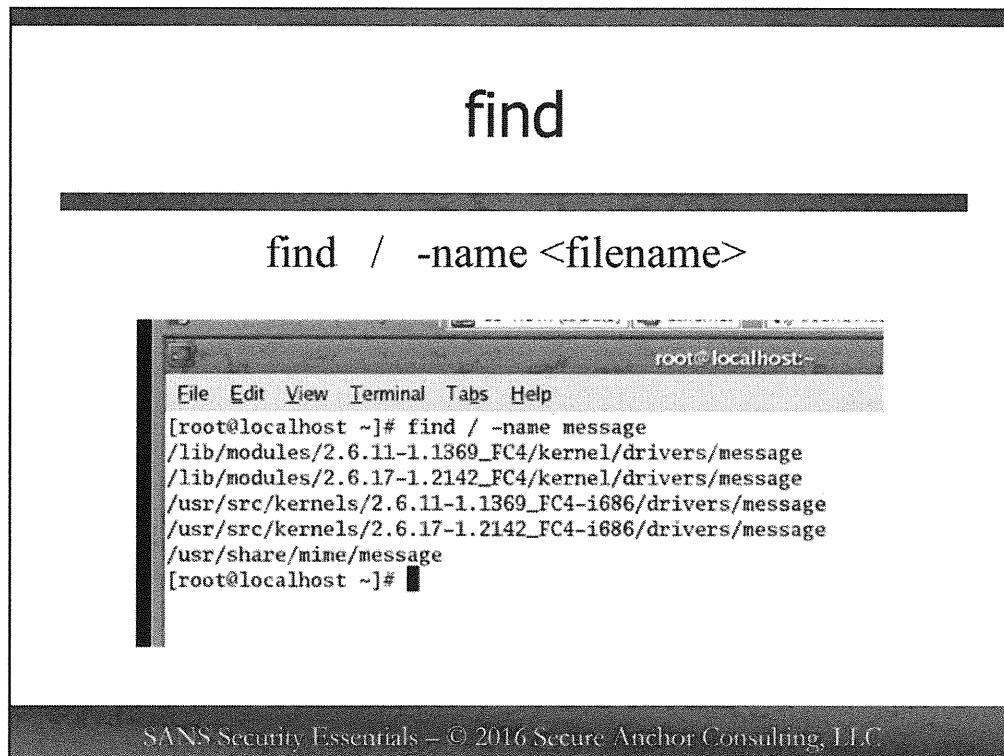
- Uses timestamp files to implement a "ticketing" system

```
File Edit View Terminal Tabs Help
[root@localhost ~]# su tigger
[tigger@localhost root]$ id
uid=502(tigger) gid=502(tigger) groups=10(wheel),502(tigger) context=user_u:syst
em_r:unconfined_t
[tigger@localhost root]$ useradd winnie
useradd: unable to lock password file
[tigger@localhost root]$ sudo useradd winnie
Password:
[tigger@localhost root]$ tail -n 1 /etc/passwd
winnie:x:504:504::/home/winnie:/bin/bash
[tigger@localhost root]$ id
uid=502(tigger) gid=502(tigger) groups=10(wheel),502(tigger) context=user_u:syst
em_r:unconfined_t
[tigger@localhost root]$
```

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

sudo (2)

The sudo command can be set to use ticketing, which allows the user to keep running the sudo command without having to input a password over and over again. This feature is based on the time the last sudo command was issued and after the time expires, will help protect the system should the individual forget to lock the system after using the command. When a user invokes *sudo* and enters a password, he is granted a ticket for 5 minutes.



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

find

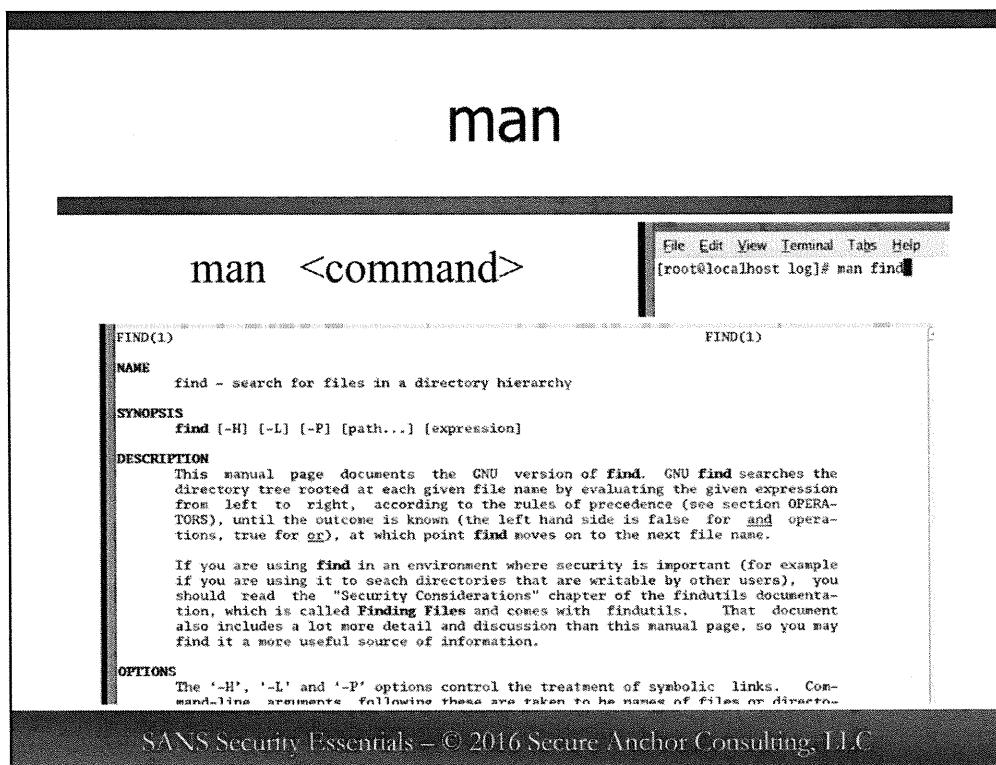
The **find** command is used to search for files within the file system. The first argument provided to this command is the location in which you wish the search to begin.

NAME

find - search for files in a directory hierarchy

SYNOPSIS

find [path...] [expression]



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

man

The man command, which is short for manual, can be used to read the manual page for any given command. By simply typing man followed by the name of the command for which you would like more information, the manual or instruction page for that command is displayed one page at a time.

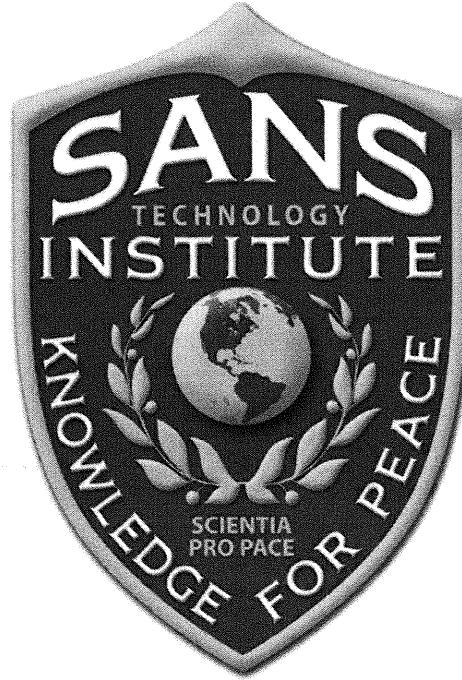
SECURITY 401 - SANS

Security Essentials

The End

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

This page intentionally left blank.



This course is part of the SANS Technology Institute (STI) Master's Degree Curriculum.

If your brain is hurting from all you've learned in this class, but you still want more, consider applying for a Master's Degree from STI. We offer two hands-on, intensive Master's Degree programs:

- Master of Science in Information Security Engineering
- Master of Science in Information Security Management

If you have a bachelor's degree and are ready to pursue a graduate degree in information security, please visit www.sans.edu for more information.

www.sans.edu

855-672-6733

info@sans.edu

Glossary of Terms

2013 SANS Institute

Glossary of Terms

Word	Definition
3-way handshake	Machine A sends a packet with a SYN flag set to Machine B. B acknowledges A's SYN with a SYN/ACK. A acknowledges B's SYN/ACK with an ACK.
Access	Opportunity to make use of an information system (IS) resource.
Access Control	Access Control ensures that resources are only granted to those users who are entitled to them.
Access Control List (ACL)	A mechanism that implements access control for a system resource by listing the identities of the system entities that are permitted to access the resource.
Access Control Mechanism	Security safeguard designed to detect and deny unauthorized access and permit authorized access in an IS.
Access Control Service	A security service that provides protection of system resources against unauthorized access. The two basic mechanisms for implementing this service are ACLs and tickets.
Access Level	Hierarchical portion of the security level used to identify the sensitivity of IS data and the clearance or authorization of users. Access level, in conjunction with the nonhierarchical

	categories, forms the sensitivity label of an object. (See category.)
Access List	(IS) Compilation of users, programs, or processes and the access levels and types to which each is authorized. (COMSEC) Roster of individuals authorized admittance to a controlled area.
Access Management	Access Management is the maintenance of access information which consists of four tasks: account administration, maintenance, monitoring, and revocation.
Access Matrix	An Access Matrix uses rows to represent subjects and columns to represent objects with privileges listed in each cell.
Access Profile	Associates each user with a list of protected objects the user may access.
Access Type	Privilege to perform action on an object. Read, write, execute, append, modify, delete, and create are examples of access types. (See write.)
Account Harvesting	Account Harvesting is the process of collecting all the legitimate account names on a system.
Accountability	(IS) Process of tracing IS activities to a responsible source. (COMSEC) Principle that an individual is entrusted to safeguard and control equipment, keying material, and information and is answerable to proper authority for the loss or misuse of that equipment or information.
Accounting Legend Code (ALC)	Numeric code used to indicate the minimum accounting controls required for items of accountable COMSEC material within the COMSEC Material Control System.
Accounting Number	Number assigned to an item of COMSEC material to facilitate its control.
Accreditation	Formal declaration by a Designated Accrediting Authority (DAA) that an IS is approved to operate at an acceptable level of risk, based on the implementation of an approved set of technical, managerial, and procedural safeguards. (See security safeguards.)
Accreditation Boundary	1. (IA) - Identifies the information resources covered by an accreditation decision, as distinguished from separately accredited information resources that are interconnected or with which information is exchanged via messaging. (Synonymous with Security

	Perimeter)
	2. (IC) - For the purposes of identifying the Protection Level for confidentiality of a system to be accredited, the system has a conceptual boundary that extends to all intended users of the system, both directly and indirectly connected, who receive output from the system (DCID 6/3, 5 Jun 99)
Accreditation package	Product comprised of a System Security Plan (SSP) and a report documenting the basis for the accreditation decision.
Accrediting authority	Synonymous with Designated Accrediting Authority (DAA).
ACK piggybacking	ACK piggybacking is the practice of sending an ACK inside another packet going to the same destination.
Active Content	Program code embedded in the contents of a web page. When the page is accessed by a web browser, the embedded code is automatically downloaded and executed on the user's workstation. Ex. Java, ActiveX (MS)
Activity Monitors	Activity monitors aim to prevent virus infection by monitoring for malicious activity on a system, and blocking that activity when possible.
Add-on security	Incorporation of new hardware, software, or firmware safeguards in an operational IS.
Address Resolution Protocol (ARP)	Address Resolution Protocol (ARP) is a protocol for mapping an Internet Protocol address to a physical machine address that is recognized in the local network. A table, usually called the ARP cache, is used to maintain a correlation between each MAC address and its corresponding IP address. ARP provides the protocol rules for making this correlation and providing address conversion in both directions.
Ad-hoc Architecture	Ad-hoc architecture is a form of wireless networking used for peer-to-peer connections in an unstructured network.
Adequate security	Security commensurate with the risk and magnitude of harm resulting from the loss, misuse, or unauthorized access to or modification of information. This includes assuring that information systems operate effectively and provide appropriate confidentiality, integrity, and availability, through the use of cost-effective management, personnel, operational, and technical controls. (OMB Circular A-130)

Advanced Encryption Standard (AES)	An encryption standard being developed by NIST. Intended to specify an unclassified, publicly-disclosed, symmetric encryption algorithm.
Advisory	Notification of significant new trends or developments regarding the threat to the IS of an organization. This notification may include analytical insights into trends, intentions, technologies, or tactics of an adversary targeting ISs.
Alert	Notification that a specific attack has been directed at the IS of an organization.
Algorithm	A finite set of step-by-step instructions for a problem-solving or computation procedure, especially one that can be implemented by a computer.
Alternate COMSEC custodian	Individual designated by proper authority to perform the duties of the COMSEC custodian during the temporary absence of the COMSEC custodian.
Alternative work site	Government-wide, national program allowing Federal employees to work at home or at geographically convenient satellite offices for part of the work week (e.g., telecommuting).
Anti-jam	Measures ensuring that transmitted information can be received despite deliberate jamming attempts.
Anti-spoof	Measures taken to prevent the unauthorized use of legitimate Identification & Authentication (I&A) data, however it was obtained, to mimic a subject different from the attacker.
Applet	Java programs; an application program that uses the client's web browser to provide a user interface.
Application	Software program that performs a specific function directly for a user and can be executed without access to system control, monitoring, or administrative privileges.
ARPANET	Advanced Research Projects Agency Network, a pioneer packet-switched network that was built in the early 1970s under contract to the U.S. Government, led to the development of today's Internet, and was decommissioned in June 1990.
Assurance	Measure of confidence that the security features, practices, procedures, and architecture of an IS accurately mediates and enforces the security policy.

Assured Software	Software that has been designed, developed, analyzed and tested using processes, tools, and techniques that establish a level of confidence in its trustworthiness appropriate for its intended use.
Asymmetric Cryptography	Public-key cryptography ; A modern branch of cryptography in which the algorithms employ a pair of keys (a public key and a private key) and use a different component of the pair for different steps of the algorithm.
Asymmetric Warfare	Asymmetric warfare is the fact that a small investment, properly leveraged, can yield incredible results.
Attack	Attempt to gain unauthorized access to an IS's services, resources, or information, or the attempt to compromise an IS's integrity, availability, or confidentiality.
Attack Sensing & Warning (AS&W)	Detection, correlation, identification, and characterization of intentional unauthorized activity with notification to decision makers so that an appropriate response can be developed.
Audit	Independent review and examination of records and activities to assess the adequacy of system controls, to ensure compliance with established policies and operational procedures, and to recommend necessary changes in controls, policies, or procedures.
Audit trail	Chronological record of system activities to enable the reconstruction and examination of the sequence of events and/or changes in an event.
Auditing	Auditing is the information gathering and analysis of assets to ensure such things as policy compliance and security from vulnerabilities.
Authenticate	To verify the identity of a user, user device, or other entity, or the integrity of data stored, transmitted, or otherwise exposed to unauthorized modification in an IS, or to establish the validity of a transmission.
Authentication	Security measure designed to establish the validity of a transmission, message, or originator, or a means of verifying an individual's authorization to receive specific categories of information.
Authentication system	Cryptosystem or process used for authentication.
Authenticator	Means used to confirm the identity of a station, originator, or individual.
Authenticity	Authenticity is the validity and conformance of the original information.

Authorization	Authorization is the approval, permission, or empowerment for someone or something to do something.
Authorized vendor	Manufacturer of INFOSEC equipment authorized to produce quantities in excess of contractual requirements for direct sale to eligible buyers. Eligible buyers are typically U.S. Government organizations or U.S. Government contractors.
Authorized Vendor Program (AVP)	Program in which a vendor, producing an INFOSEC product under contract to NSA, is authorized to produce that product in numbers exceeding the contracted requirements for direct marketing and sale to eligible buyers. Eligible buyers are typically U.S. Government organizations or U.S. Government contractors. Products approved for marketing and sale through the AVP are placed on the Endorsed Cryptographic Products List (ECPL).
Automated security monitoring	Use of automated procedures to ensure security controls are not circumvented or the use of these tools to track actions taken by subjects suspected of misusing the IS.
Automatic remote rekeying	Procedure to rekey a distant crypto-equipment electronically without specific actions by the receiving terminal operator. (See manual remote rekeying.)
Autonomous system	<p>One network or series of networks that are all under one administrative control.</p> <p>An autonomous system is also sometimes referred to as a routing domain. An autonomous system is assigned a globally unique number, sometimes called an Autonomous System Number (ASN).</p>
Availability	Availability is the need to ensure that the business purpose of the system can be met and that it is accessible to those who need to use it.
Backdoor	A backdoor is a tool installed after a compromise to give an attacker easier access to the compromised system around any security mechanisms that are in place.
Backup	Copy of files and programs made to facilitate recovery, if necessary.
Bandwidth	Commonly used to mean the capacity of a communication channel to pass data through the channel in a given amount of time. Usually expressed in bits per second.

Banner	A banner is the information that is displayed to a remote user trying to connect to a service. This may include version information, system information, or a warning about authorized use. Display on an IS that sets parameters for system or data use.
Basic Authentication	Basic Authentication is the simplest web-based authentication scheme that works by sending the username and password with each request.
Bastion Host	A bastion host has been hardened in anticipation of vulnerabilities that have not been discovered yet
Benign	Condition of cryptographic data that cannot be compromised by human access.
Benign environment	Nonhostile environment that may be protected from external hostile elements by physical, personnel, and procedural security countermeasures.
BIND	BIND stands for Berkeley Internet Name Domain and is an implementation of DNS. DNS is used for domain name to IP address resolution.
Binding	Process of associating a specific communications terminal with a specific cryptographic key or associating two related elements of information.
Biometrics	Automated methods of authenticating or verifying an individual based upon a physical or behavioral characteristic. Biometrics use physical characteristics of the users to determine access.
Bit	The smallest unit of information storage; a contraction of the term "binary digit;" one of two symbols--"0" (zero) and "1" (one) -- that are used to represent binary numbers.
Bit error rate	Ratio between the number of bits incorrectly received and the total number of bits transmitted in a telecommunications system.
BLACK	Designation applied to information systems, and to associated areas, circuits, components, and equipment, in which national security information is encrypted or is not processed.
Block Cipher	A block cipher encrypts one block of data at a time.
Boot Record Infector	A boot record infector is a piece of malware that inserts malicious code into the boot sector of a disk.

Border Gateway Protocol (BGP)	An inter-autonomous system routing protocol. BGP is used to exchange routing information for the Internet and is the protocol used between Internet service providers (ISP).
Boundary	Software, hardware, or physical barrier that limits access to a system or part of a system.
Brevity list C.F.D.	List containing words and phrases used to shorten messages.
Bridge	A product that connects a local area network (LAN) to another local area network that uses the same protocol (for example, Ethernet or token ring).
British Standard 7799	A standard code of practice and provides guidance on how to secure an information system. It includes the management framework, objectives, and control requirements for information security management systems.
Broadcast	To simultaneously send the same message to multiple recipients. One host to all hosts on network.
Broadcast Address	An address used to broadcast a datagram to all hosts on a given network using UDP or ICMP protocol.
Browser	An client computer program that can retrieve and display information from servers on the World Wide Web.
Browsing	Act of searching through IS storage to locate or acquire information, without necessarily knowing the existence or format of information being sought.
Brute Force	A cryptanalysis technique or other kind of attack method involving an exhaustive procedure that tries all possibilities, one-by-one.
Buffer Overflow	A buffer overflow occurs when a program or process tries to store more data in a buffer (temporary data storage area) than it was intended to hold. Since buffers are created to contain a finite amount of data, the extra information - which has to go somewhere - can overflow into adjacent buffers, corrupting or overwriting the valid data held in them.
Bulk encryption	Simultaneous encryption of all channels of a multichannel telecommunications link.
Business Continuity Plan (BCP)	A Business Continuity Plan is the plan for emergency response, backup operations, and post-disaster recovery steps that will ensure the availability of critical resources and facilitate the

	continuity of operations in an emergency situation.
Business Impact Analysis (BIA)	A Business Impact Analysis determines what levels of impact to a system are tolerable.
Byte	A fundamental unit of computer storage; the smallest addressable unit in a computer's architecture. Usually holds one character of information and usually means eight bits.
Cache	Pronounced cash, a special high-speed storage mechanism. It can be either a reserved section of main memory or an independent high-speed storage device. Two types of caching are commonly used in personal computers: memory caching and disk caching.
Cache Cramming	Cache Cramming is the technique of tricking a browser to run cached Java code from the local disk, instead of the internet zone, so it runs with less restrictive permissions.
Cache Poisoning	Malicious or misleading data from a remote name server is saved [cached] by another name server. Typically used with DNS cache poisoning attacks.
Call back	Procedure for identifying and authenticating a remote IS terminal, whereby the host system disconnects the terminal and reestablishes contact. Synonymous with dial back.
Canister	Type of protective package used to contain and dispense keying material in punched or printed tape form.
Cascading	Downward flow of information through a range of security levels greater than the accreditation range of a system network or component.
Category	Restrictive label applied to classified or unclassified information to limit access.
CCI assembly	Device embodying a cryptographic logic or other COMSEC design that NSA has approved as a Controlled Cryptographic Item (CCI). It performs the entire COMSEC function, but depends upon the host equipment to operate.
CCI component	Part of a Controlled Cryptographic Item (CCI) that does not perform the entire COMSEC function but depends upon the host equipment, or assembly, to complete and operate the COMSEC function.
CCI equipment	Telecommunications or information handling equipment that embodies a Controlled Cryptographic Item (CCI) component or CCI assembly and performs the entire COMSEC

	function without dependence on host equipment to operate.
Cell	A cell is a unit of data transmitted over an ATM network.
Central office of Record (COR)	Office of a federal department or agency that keeps records of accountable COMSEC material held by elements subject to its oversight.
Certificate	Digitally signed document that binds a public key with an identity. The certificate contains, at a minimum, the identity of the issuing Certification Authority, the user identification information, and the user's public key.
Certificate-Based Authentication	Certificate-Based Authentication is the use of SSL and certificates to authenticate and encrypt HTTP traffic.
Certificate management	Process whereby certificates (as defined above) are generated, stored, protected, transferred, loaded, used, and destroyed.
Certificate revocation list (CRL)	List of invalid certificates (as defined above) that have been revoked by the issuer.
Certification	Comprehensive evaluation of the technical and nontechnical security safeguards of an IS to support the accreditation process that establishes the extent to which a particular design and implementation meets a set of specified security requirements.
Certification authority (CA)	(C&A) Official responsible for performing the comprehensive evaluation of the security features of an information system and determining the degree to which it meets its security requirements. (PKI) Trusted entity authorized to create, sign, and issue public key certificates. By digitally signing each certificate issued, the user's identity is certified, and the association of the certified identity with a public key is validated.
Certification authority workstation (CAW)	Commercial-off-the-shelf (COTS) workstation with a trusted operating system and special purpose application software that is used to issue certificates.
Certification package	Product of the certification effort documenting the detailed results of the certification activities.
Certification test & evaluation (CT&E)	Software and hardware security tests conducted during development of an IS.
Certified TEMPEST technical authority	An experienced, technically qualified U.S. Government employee who has met established certification requirements in accordance with

	CNSS (NSTIISCC)-approved criteria and has been appointed by a U.S. Government Department or Agency to fulfill CTTA responsibilities.
Certifier	Individual responsible for making a technical judgment of the system's compliance with stated requirements, identifying and assessing the risks associated with operating the system, coordinating the certification activities, and consolidating the final certification and accreditation packages.
Chain of Custody	Chain of Custody is the important application of the Federal rules of evidence and its handling.
Challenge and reply authentication	Peararranged procedure in which a subject requests authentication of another and the latter establishes validity with a correct reply.
Challenge-Handshake Authentication Protocol (CHAP)	The Challenge-Handshake Authentication Protocol uses a challenge/response authentication mechanism where the response varies every challenge to prevent replay attacks.
Checksum	A value that is computed by a function that is dependent on the contents of a data object and is stored or transmitted together with the object, for the purpose of detecting changes in the data.
Check word	Cipher text generated by cryptographic logic to detect failures in cryptography.
Cipher	Any cryptographic system in which arbitrary symbols or groups of symbols, represent units of plain text, or in which units of plain text are rearranged, or both.
Cipherlock	A cipher lock is a mechanical lock the accepts a specific key sequence to grant access.
Ciphertext	Ciphertext is the encrypted form of the message being sent.
Cipher text auto-key (CTAK)	Cryptographic logic that uses previous cipher text to generate a key stream.
Ciphony	Process of enciphering audio information, resulting in encrypted speech.
Circuit Switched Network	A circuit switched network is where a single continuous physical circuit connected two endpoints where the route was immutable once set up.
Classified information	Information that has been determined pursuant to Executive Order 12958 or any predecessor Order, or by the Atomic Energy Act of 1954, as amended, to require protection against unauthorized disclosure and is marked to indicate its classified status.

Classified information spillage	Security incident that occurs whenever classified data is spilled either onto an unclassified IS or to an IS with a lower level of classification.
Clearance	Formal security determination by an authorized adjudicative office that an individual is authorized access, on a need to know basis, to a specific level of collateral classified information (TOP SECRET, SECRET, CONFIDENTIAL).
Clearing	Removal of data from an IS, its storage devices, and other peripheral devices with storage capacity, in such a way that the data may not be reconstructed using common system capabilities (i.e., keyboard strokes); however, the data may be reconstructed using laboratory methods. Cleared media may be reused at the same classification level or at a higher level. Overwriting is one method of clearing. (See magnetic remanence.)
Client	Individual or process acting on behalf of an individual who makes requests of a guard or dedicated server. The client's requests to the guard or dedicated server can involve data transfer to, from, or through the guard or dedicated server.
Closed security environment	Environment providing sufficient assurance that applications and equipment are protected against the introduction of malicious logic during an IS life cycle. Closed security is based upon a system's developers, operators, and maintenance personnel having sufficient clearances, authorization, and configuration control.
Code	(COMSEC) System of communication in which arbitrary groups of letters, numbers, or symbols represent units of plain text of varying length.
Code book	Document containing plain text and code equivalents in a systematic arrangement, or a technique of machine encryption using a word substitution technique.
Code group	Group of letters, numbers, or both in a code system used to represent a plain text word, phrase, or sentence.
Code vocabulary	Set of plain text words, numerals, phrases, or sentences for which code equivalents are assigned in a code system.
Cold start	Procedure for initially keying crypto-equipment.
Collaborative computing	Applications and technology (e.g., whiteboarding, group conferencing) that allow two or more individuals to share information real time in an inter- or intra-enterprise environment.

Collision	A collision occurs when multiple systems transmit simultaneously on the same wire.
Command authority	Individual responsible for the appointment of user representatives for a department, agency, or organization and their key ordering privileges.
Commercial COMSEC Evaluation Program (CCEP)	Relationship between NSA and industry in which NSA provides the COMSEC expertise (i.e., standards, algorithms, evaluations, and guidance) and industry provides design, development, and production capabilities to produce a type 1 or type 2 product. Products developed under the CCEP may include modules, subsystems, equipment, systems, and ancillary devices.
Common Criteria	Provides a comprehensive, rigorous method for specifying security function and assurance requirements for products and systems. (International Standard ISO/IEC 5408, Common Criteria for Information Technology Security Evaluation [ITSEC])
Common fill device	One of a family of devices developed to read-in, transfer, or store key.
Communications cover	Concealing or altering of characteristic communications patterns to hide information that could be of value to an adversary.
Communications deception	Deliberate transmission, retransmission, or alteration of communications to mislead an adversary's interpretation of the communications. (See imitative communications deception and manipulative communications deception.)
Communications profile	Analytic model of communications associated with an organization or activity. The model is prepared from a systematic examination of communications content and patterns, the functions they reflect, and the communications security measures applied.
Communications Security (COMSEC)	Measures and controls taken to deny unauthorized individuals information derived from telecommunications and to ensure the authenticity of such telecommunications. Communications security includes cryptosecurity, transmission security, emission security, and physical security of COMSEC material.
Community risk	Probability that a particular vulnerability will be exploited within an interacting population and adversely impact some members of that population.
Compartmentalization	A nonhierarchical grouping of sensitive

	information used to control access to data more finely than with hierarchical security classification alone.
Compartmented mode	Mode of operation wherein each user with direct or indirect access to a system, its peripherals, remote terminals, or remote hosts has all of the following: (a) valid security clearance for the most restricted information processed in the system; (b) formal access approval and signed nondisclosure agreements for that information which a user is to have access; and (c) valid need-to-know for information which a user is to have access.
Compromise	Type of incident where information is disclosed to unauthorized individuals or a violation of the security policy of a system in which unauthorized intentional or unintentional disclosure, modification, destruction, or loss of an object may have occurred.
Compromising emanations	Unintentional signals that, if intercepted and analyzed, would disclose the information transmitted, received, handled, or otherwise processed by information systems equipment. (See TEMPEST.)
Competitive Intelligence	Competitive Intelligence is espionage using legal, or at least not obviously illegal, means.
Computer abuse	Intentional or reckless misuse, alteration, disruption, or destruction of information processing resources.
Computer cryptography	Use of a crypto-algorithm program by a computer to authenticate or encrypt/decrypt information.
Computer Emergency Response Team (CERT)	An organization that studies computer and network INFOSEC in order to provide incident response services to victims of attacks, publish alerts concerning vulnerabilities and threats, and offer other information to help improve computer and network security.
Computer network	A collection of host computers together with the sub-network or inter-network through which they can exchange data.
Computer security	Measures and controls that ensure confidentiality, integrity, and availability of IS assets including hardware, software, firmware, and information being processed, stored, and communicated.
Computer security incident	See incident.

Computer security subsystem	Hardware/software designed to provide computer security features in a larger system environment.
Computing environment	Workstation or server (host) and its operating system, peripherals, and applications.
COMSEC account	Administrative entity, identified by an account number, used to maintain accountability, custody, and control of COMSEC material.
COMSEC account audit	Examination of the holdings, records, and procedures of a COMSEC account ensuring all accountable COMSEC material is properly handled and safeguarded.
COMSEC aid	COMSEC material that assists in securing telecommunications and is required in the production, operation, or maintenance of COMSEC systems and their components. COMSEC keying material, callsign/frequency systems, and supporting documentation, such as operating and maintenance manuals, are examples of COMSEC aids.
COMSEC assembly	Group of parts, elements, subassemblies, or circuits that are removable items of COMSEC equipment.
COMSEC boundary	Definable perimeter encompassing all hardware, firmware, and software components performing critical COMSEC functions, such as key generation, handling, and storage.
COMSEC chip set	Collection of NSA approved microchips.
COMSET	Computer instructions or routines controlling or affecting the externally performed functions of key generation, key distribution, message encryption/decryption, or authentication.
COMSEC custodian	Individual designated by proper authority to be responsible for the receipt, transfer, accounting, safeguarding, and destruction of COMSEC material assigned to a COMSEC account.
COMSEC demilitarization	Process of preparing COMSEC equipment for disposal by extracting all CCI, classified, or CRYPTO marked components for their secure destruction, as well as defacing and disposing of the remaining equipment hulk.
COMSEC element	Removable item of COMSEC equipment, assembly, or subassembly; normally consisting of a single piece or group of replaceable parts.
COMSEC end-item	Equipment or combination of components ready for use in a COMSEC application.

COMSEC equipment	Equipment designed to provide security to telecommunications by converting information to a form unintelligible to an unauthorized interceptor and, subsequently, by reconverting such information to its original form for authorized recipients; also, equipment designed specifically to aid in, or as an essential element of, the conversion process. COMSEC equipment includes crypto-equipment, crypto-ancillary equipment, cryptoproduction equipment, and authentication equipment.
COMSEC facility	Authorized and approved space used for generating, storing, repairing, or using COMSEC material.
COMSEC incident	See incident.
COMSEC insecurity	COMSEC incident that has been investigated, evaluated, and determined to jeopardize the security of COMSEC material or the secure transmission of information.
COMSEC manager	Individual who manages the COMSEC resources of an organization.
COMSEC material	Item designed to secure or authenticate telecommunications. COMSEC material includes, but is not limited to key, equipment, devices, documents, firmware, or software that embodies or describes cryptographic logic and other items that perform COMSEC functions.
COMSEC Material Control System (CMCS)	Logistics and accounting system through which COMSEC material marked "CRYPTO" is distributed, controlled, and safeguarded. Included are the COMSEC central offices of record, cryptologic depots, and COMSEC accounts. COMSEC material other than key may be handled through the CMCS.
COMSEC modification	See information systems security equipment modification.
COMSEC module	Removable component that performs COMSEC functions in a telecommunications equipment or system.
COMSEC monitoring	Act of listening to, copying, or recording transmissions of one's own official telecommunications to analyze the degree of security.
COMSEC profile	Statement of COMSEC measures and materials used to protect a given operation, system, or organization.

COMSEC survey	Organized collection of COMSEC and communications information relative to a given operation, system, or organization.
COMSEC system data	Information required by a COMSEC equipment or system to enable it to properly handle and control key.
COMSEC training	Teaching of skills relating to COMSEC accounting, use of COMSEC aids, or installation, use, maintenance, and repair of COMSEC equipment.
Concept of Operations (CONOP)	Document detailing the method, act, process, or effect of using an IS.
Confidentiality	Confidentiality is the need to ensure that information is disclosed only to those who are authorized to view it.
Configuration Control	Process of controlling modifications to hardware, firmware, software, and documentation to ensure the IS is protected against improper modifications prior to, during, and after system implementation.
Configuration Management	Management of security features and assurances through control of changes made to hardware, software, firmware, documentation, test, test fixtures, and test documentation throughout the life cycle of an IS.
Confinement channel	See covert channel.
Contamination	Type of incident involving the introduction of data of one security classification or security category into data of a lower security classification or different security category.
Contingency key	Key held for use under specific operational conditions or in support of specific contingency plans. (See reserve keying material.)
Continuity of operations plan (COOP)	Plan for continuing an organization's (usually a headquarters element) essential functions at an alternate site and performing those functions for the duration of an event with little or no loss of continuity before returning to normal operations.
Controlled access area	Physical area (e.g., building, room, etc.) to which only authorized personnel are granted unrestricted access. All other personnel are either escorted by authorized personnel or are under continuous surveillance.
Controlled access protection	Minimum set of security functionality that enforces access control on individual users and makes them accountable for their actions through login procedures, auditing of security-relevant events, and resource isolation.

Controlled cryptographic item (CCI)	Secure telecommunications or information handling equipment, or associated cryptographic component, that is unclassified but governed by a special set of control requirements. Such items are marked "CONTROLLED CRYPTOGRAPHIC ITEM" or, where space is limited, "CCI."
Controlled interface	Mechanism that facilitates the adjudication of different interconnected system security policies (e.g., controlling the flow of information into or out of an interconnected system).
Controlled space	Three-dimensional space surrounding IS equipment, within which unauthorized individuals are denied unrestricted access and are either escorted by authorized individuals or are under continuous physical or electronic surveillance.
Controlling authority	Official responsible for directing the operation of a cryptonet and for managing the operational use and control of keying material assigned to the cryptonet.
Cookie	Data exchanged between an HTTP server and a browser (a client of the server) to store state information on the client side and retrieve it later for server use. An HTTP server, when sending data to a client, may send along a cookie, which the client retains after the HTTP connection closes. A server can use this mechanism to maintain persistent client-side state information for HTTP-based applications, retrieving the state information in later connections.
Cooperative key generation	Electronically exchanging functions of locally generated, random components, from which both terminals of a secure circuit construct traffic encryption key or key encryption key for use on that circuit. (See per-call key.)
Cooperative remote rekeying	Synonymous with manual remote rekeying.
Correctness proof	A mathematical proof of consistency between a specification and its implementation.
Corruption	A threat action that undesirably alters system operation by adversely modifying system functions or data.
Countermeasure	Action, device, procedure, technique, or other measure that reduces the vulnerability of an IS.
Cost Benefit Analysis	A cost benefit analysis compares the cost of implementing countermeasures with the value of the reduced risk.

Covert Channels	Unintended and/or unauthorized communications path that can be used to transfer information in a manner that violates an IS security policy. (See overt channel and exploitable channel.)
Covert channel analysis	Determination of the extent to which the security policy model and subsequent lower-level program descriptions may allow unauthorized access to information.
Covert storage channel	Covert channel involving the direct or indirect writing to a storage location by one process and the direct or indirect reading of the storage location by another process. Covert storage channels typically involve a finite resource (e.g., sectors on a disk) that is shared by two subjects at different security levels.
Covert timing channel	Covert channel in which one process signals information to another process by modulating its own use of system resources (e.g., central processing unit time) in such a way that this manipulation affects the real response time observed by the second process.
Credentials	Information, passed from one entity to another, used to establish the sending entity's access rights.
Critical infrastructures	System and assets, whether physical or virtual, so vital to the U.S. that the incapacity or destruction of such systems and assets would have a debilitating impact on security, national economic security, national public health or safety, or any combination of those matters. [Critical Infrastructures Protection Act of 2001, 42 U.S.C. 5195c(e)]
Cron	Cron is a Unix application that runs jobs for users and administrators at scheduled times of the day.
Cross domain solution	Information assurance solution that provides the ability to access or transfer information between two or more security domains. (See multi level security.)
Crossover Cable	A crossover cable reverses the pairs of cables at the other end and can be used to connect devices directly together.
Cryptanalysis	The mathematical science that deals with analysis of a cryptographic system in order to gain knowledge needed to break or circumvent the protection that the system is designed to provide. In other words, convert the cipher text to plaintext without knowing the key.

CRYPTO	Marking or designator identifying COMSEC keying material used to secure or authenticate telecommunications carrying classified or sensitive U.S. Government or U.S. Government-derived information.
Crypto-alarm	Circuit or device that detects failures or aberrations in the logic or operation of crypto-equipment. Crypto-alarm may inhibit transmission or may provide a visible and/or audible alarm.
Crypto-algorithm	Well-defined procedure or sequence of rules or steps, or a series of mathematical equations used to describe cryptographic processes such as encryption/decryption, key generation, authentication, signatures, etc.
Cryptographic Algorithm or Hash	An algorithm that employs the science of cryptography, including encryption algorithms, cryptographic hash algorithms, digital signature algorithms, and key agreement algorithms.
Crypto-ancillary equipment	Equipment designed specifically to facilitate efficient or reliable operation of crypto-equipment, without performing cryptographic functions itself.
Crypto-equipment	Equipment that embodies a cryptographic logic.
Cryptographic	Pertaining to, or concerned with, cryptography.
Cryptographic component	Hardware or firmware embodiment of the cryptographic logic. A cryptographic component may be a modular assembly, a printed wiring assembly, a microcircuit, or a combination of these items.
Cryptographic initialization	Function used to set the state of a cryptographic logic prior to key generation, encryption, or other operating mode.
Cryptographic logic	The embodiment of one (or more) crypto-algorithm(s) along with alarms, checks, and other processes essential to effective and secure performance of the cryptographic process(es).
Cryptographic randomization	Function that randomly determines the transmit state of a cryptographic logic.
Cryptography	Art or science concerning the principles, means, and methods for rendering plain information unintelligible and for restoring encrypted information to intelligible form.
Crypto-ignition key (CIK)	Device or electronic key used to unlock the secure mode of crypto-equipment.
Cryptology	Field encompassing both cryptography and cryptanalysis.
Cryptonet	Stations holding a common key.

Cryptoperiod	Time span during which each key setting remains in effect.
Cryptosecurity	Component of COMSEC resulting from the provision of technically sound cryptosystems and their proper use.
Cryptosynchronization	Process by which a receiving decrypting cryptographic logic attains the same internal state as the transmitting encrypting logic.
Cryptosystem	Associated INFOSEC items interacting to provide a single means of encryption or decryption.
Cryptosystem analysis	Process of establishing the exploitability of a cryptosystem, normally by reviewing transmitted traffic protected or secured by the system under study.
Cryptosystem evaluation	Process of determining vulnerabilities of a cryptosystem.
Cryptosystem review	Examination of a cryptosystem by the controlling authority ensuring its adequacy of design and content, continued need, and proper distribution.
Cryptosystem survey	Management technique in which actual holders of a cryptosystem express opinions on the system's suitability and provide usage information for technical evaluations.
Cut-Through	Cut-Through is a method of switching where only the header of a packet is read before it is forwarded to its destination.
Cyclic Redundancy Check (CRC)	Sometimes called "cyclic redundancy code." A type of checksum algorithm that is not a cryptographic hash but is used to implement data integrity service where accidental changes to data are expected.
Daemon	A program which is often started at the time the system boots and runs continuously without intervention from any of the users on the system. The daemon program forwards the requests to other programs (or processes) as appropriate. The term daemon is a Unix term, though many other operating systems provide support for daemons, though they're sometimes called other names. Windows, for example, refers to daemons and System Agents and services.
Data aggregation	Compilation of unclassified individual data systems and data elements that could result in the totality of the information being classified or of beneficial use to an adversary.

Data Custodian	A Data Custodian is the entity currently using or manipulating the data, and therefore, temporarily taking responsibility for the data.
Data Encryption Standard (DES)	A widely-used method of data encryption using a private (secret) key. There are 72,000,000,000,000,000 (72 quadrillion) or more possible encryption keys that can be used. For each given message, the key is chosen at random from among this enormous number of keys. Like other private key cryptographic methods, both the sender and the receiver must know and use the same private key.
Data flow control	Synonymous with information flow control.
Data integrity	Condition existing when data is unchanged from its source and has not been accidentally or maliciously modified, altered, or destroyed.
Data Mining	Data Mining is a technique used to analyze existing information, usually with the intention of pursuing new avenues to pursue business.
Data origination authentication	Corroborating the source of data is as claimed.
Data Owner	A Data Owner is the entity having responsibility and authority for the data.
Data security	Protection of data from unauthorized (accidental or intentional) modification, destruction, or disclosure.
Data transfer device (DTD)	Fill device designed to securely store, transport, and transfer electronically both COMSEC and TRANSEC key, designed to be backward compatible with the previous generation of COMSEC common fill devices, and programmable to support modern mission systems.
Data Warehousing	Data Warehousing is the consolidation of several previously independent databases into one location.
Datagram	Request for Comment 1594 says, "a self-contained, independent entity of data carrying sufficient information to be routed from the source to the destination computer without reliance on earlier exchanges between this source and destination computer and the transporting network." The term has been generally replaced by the term packet. Datagrams or packets are the message units that the Internet Protocol deals with and that the Internet transports. A datagram or packet needs to be self-contained without reliance on earlier exchanges because there is no connection of fixed duration between the two communicating points as there is, for example, in

	most voice telephone conversations. (This kind of protocol is referred to as connectionless.)
Decapsulation	Decapsulation is the process of stripping off one layer's headers and passing the rest of the packet up to the next higher layer on the protocol stack.
Decertification	Revocation of the certification of an IS item or equipment for cause.
Decipher	Convert enciphered text to plain text by means of a cryptographic system.
Decode	Convert encoded text to plain text by means of a code.
Decrypt	Generic term encompassing decode and decipher.
Decryption	Decryption is the process of transforming an encrypted message into its original plaintext.
Dedicated mode	IS security mode of operation wherein each user, with direct or indirect access to the system, its peripherals, remote terminals, or remote hosts, has all of the following: a. valid security clearance for all information within the system; b. formal access approval and signed nondisclosure agreements for all the information stored and/or processed (including all compartments, subcompartments, and/or special access programs); and c. valid need-to-know for all information contained within the IS. When in the dedicated security mode, a system is specifically and exclusively dedicated to and controlled for the processing of one particular type or classification of information, either for full-time operation or for a specified period of time.
Defacement	Defacement is the method of modifying the content of a website in such a way that it becomes "vandalized" or embarrassing to the website owner.
Default classification	Temporary classification reflecting the highest classification being processed in an IS. Default classification is included in the caution statement affixed to an object.
Defense-in-Depth	Defense-in-Depth is the approach of using multiple layers of security to guard against failure of a single security component. IA strategy integrating people, technology, and operations capabilities to establish variable barriers across multiple layers and dimensions of networks. Synonymous with security-in-depth.

Degaussing	Procedure that reduces the magnetic flux to virtual zero by applying a reverse magnetizing field. Also called demagnetizing.
Delegated development program	INFOSEC program in which the Director, NSA, delegates, on a case by case basis, the development and/or production of an entire telecommunications product, including the INFOSEC portion, to a lead department or agency.
Denial of Service	The prevention of authorized access to a system resource or the delaying of system operations and functions.
Description top-level specification (C.F.D.)	Top-level specification written in a natural language (e.g., English), an informal design notation, or a combination of the two. Descriptive top-level specification, required for a class B2 and B3 (as defined in the Orange Book, Department of Defense Trusted Computer System Evaluation Criteria, DoD 5200.28-STD) information system, completely and accurately describes a trusted computing base. (See formal top-level specification.)
Designated approval authority (DAA)	Official with the authority to formally assume responsibility for operating a system at an acceptable level of risk. This term is synonymous with authorizing official, designated accrediting authority, and delegated accrediting authority.
Dial back	Synonymous with call back.
Dictionary Attack	An attack that tries all of the phrases or words in a dictionary, trying to crack a password or key. A dictionary attack uses a predefined list of words compared to a brute force attack that tries all possible combinations.
Diffie-Hellman	A key agreement algorithm published in 1976 by Whitfield Diffie and Martin Hellman. Diffie-Hellman does key establishment, not encryption. However, the key that it produces may be used for encryption, for further key management operations, or for any other cryptography.
Digest Authentication	Digest Authentication allows a web client to compute MD5 hashes of the password to prove it has the password.
Digital Certificate	A digital certificate is an electronic "credit card" that establishes your credentials when doing business or other transactions on the Web. It is issued by a certification authority. It contains your name, a serial number, expiration dates, a copy of the certificate holder's public key (used

	for encrypting messages and digital signatures), and the digital signature of the certificate-issuing authority so that a recipient can verify that the certificate is real.
Digital Envelope	A digital envelope is an encrypted message with the encrypted session key.
Digital Signature	A digital signature is a hash of a message that uniquely identifies the sender of the message and proves the message hasn't changed since transmission.
Digital Signature Algorithm (DSA)	An asymmetric cryptographic algorithm that produces a digital signature in the form of a pair of large numbers. The signature is computed using rules and parameters such that the identity of the signer and the integrity of the signed data can be verified.
Digital Signature Standard (DSS)	The U.S. Government standard that specifies the Digital Signature Algorithm (DSA), which involves asymmetric cryptography.
Direct shipment	Shipment of COMSEC material directly from NSA to user COMSEC accounts.
Disassembly	The process of taking a binary program and deriving the source code from it.
Disaster Recovery Plan (DRP)	A Disaster Recovery Plan is the process of recovery of IT systems in the event of a disruption or disaster.
Discretionary Access Control (DAC)	Means of restricting access to objects based on the identity and need-to-know of users and/or groups to which the object belongs. Controls are discretionary in the sense that a subject with a certain access permission is capable of passing that permission (directly or indirectly) to any other subject. (See mandatory access control.)
Disruption	A circumstance or event that interrupts or prevents the correct operation of system services and functions.
Distance Vector	Distance vectors measure the cost of routes to determine the best route to all known networks.
Distinguished name	Globally unique identifier representing an individual's identity.
Distributed Scans	Distributed Scans are scans that use multiple source addresses to gather information.
DMZ (Demilitarized Zone)	Perimeter network segment that is logically between internal and external networks. Its purpose is to enforce the internal network's IA policy for external information exchange and to provide external, untrusted sources with restricted access to releasable information while

	shielding the internal networks from outside attacks. A DMZ is also called a "screened subnet."
Domain	A sphere of knowledge, or a collection of facts about some program entities or a number of network points or addresses, identified by a name. On the Internet, a domain consists of a set of network addresses. In the Internet's domain name system, a domain is a name with which name server records are associated that describe sub-domains or host. In Windows NT and Windows 2000, a domain is a set of network resources (applications, printers, and so forth) for a group of users. The user need only to log in to the domain to gain access to the resources, which may be located on a number of different servers in the network.
Domain Name	A domain name locates an organization or other entity on the Internet. For example, the domain name "www.sans.org" locates an Internet address for "sans.org" at Internet point 199.0.0.2 and a particular host server named "www". The "org" part of the domain name reflects the purpose of the organization or entity (in this example, "organization") and is called the top-level domain name. The "sans" part of the domain name defines the organization or entity and together with the top-level is called the second-level domain name.
Domain Hijacking	Domain hijacking is an attack by which an attacker takes over a domain by first blocking access to the domain's DNS server and then putting his own server up in its place.
Domain Name System (DNS)	The domain name system (DNS) is the way that Internet domain names are located and translated into Internet Protocol addresses. A domain name is a meaningful and easy-to-remember "handle" for an Internet address.
Drop accountability	Procedure under which a COMSEC account custodian initially receipts for COMSEC material, and provides no further accounting for it to its central office of record. Local accountability of the COMSEC material may continue to be required. (See accounting legend code.)
Due Care	Due care ensures that a minimal level of protection is in place in accordance with the best practice in the industry.

Due Diligence	Due diligence is the requirement that organizations must develop and deploy a protection plan to prevent fraud, abuse, and additional deploy a means to detect them if they occur.
DumpSec	DumpSec is a security tool that dumps a variety of information about a system's users, file system, registry, permissions, password policy, and services.
Dumpster Diving	Dumpster Diving is obtaining passwords and corporate directories by searching through discarded media.
Dynamic Link Library	A collection of small programs, any of which can be called when needed by a larger program that is running in the computer. The small program that lets the larger program communicate with a specific device such as a printer or scanner is often packaged as a DLL program (usually referred to as a DLL file).
Dynamic Routing Protocol	Allows network devices to learn routes. Ex. RIP, EIGRP Dynamic routing occurs when routers talk to adjacent routers, informing each other of what networks each router is currently connected to. The routers must communicate using a routing protocol, of which there are many to choose from. The process on the router that is running the routing protocol, communicating with its neighbor routers, is usually called a routing daemon. The routing daemon updates the kernel's routing table with information it receives from neighbor routers.
Eavesdropping	Eavesdropping is simply listening to a private conversation which may reveal information which can provide access to a facility or network.
Echo Reply	An echo reply is the response a machine that has received an echo request sends over ICMP.
Echo Request	An echo request is an ICMP message sent to a machine to determine if it is online and how long traffic takes to get to it.
Egress Filtering	Filtering outbound traffic.
Electronically generated key	Key generated in a COMSEC device by introducing (either mechanically or electronically) a seed key into the device and then using the seed, together with a software algorithm stored in the device, to produce the desired key.

Electronic Key Management System (EKMS)	Interoperable collection of systems being developed by services and agencies of the U.S. Government to automate the planning, ordering, generating, distributing, storing, filling, using, and destroying of electronic key and management of other types of COMSEC material.
Electronic messaging services	Services providing interpersonal messaging capability; meeting specific functional, management, and technical requirements; and yielding a business-quality electronic mail service suitable for the conduct of official government business.
Electronic security (ELSEC) (C.F.D.)	Protection resulting from measures designed to deny unauthorized individuals information derived from the interception and analysis of noncommunications electromagnetic radiations.
Electronic signature	See digital signature.
Emanations Analysis	Gaining direct knowledge of communicated data by monitoring and resolving a signal that is emitted by a system and that contains the data but is not intended to communicate the data.
Embedded computer (C.F.D.)	Computer system that is an integral part of a larger system.
Embedded cryptography (C.F.D.)	Cryptography engineered into an equipment or system whose basic function is not cryptographic.
Embedded cryptographic system (C.F.D.)	Cryptosystem performing or controlling a function as an integral element of a larger system or subsystem.
Emissions Security (EMSEC)	Protection resulting from measures taken to deny unauthorized individuals information derived from intercept and analysis of compromising emanations from crypto-equipment or an IS. (See TEMPEST.)
Encapsulation	The inclusion of one data structure within another structure so that the first data structure is hidden for the time being.
Encipher	Convert plain text to cipher text by means of a cryptographic system.
Enclave	Collection of computing environments connected by one or more internal networks under the control of a single authority and security policy, including personnel and physical security.
Enclave boundary	Point at which an enclave's internal network service layer connects to an external network's service layer, i.e., to another enclave or to a Wide Area Network (WAN).

Encode	Convert plain text to cipher text by means of a code.
Encrypt	Generic term encompassing encipher and encode.
Encryption	Cryptographic transformation of data (called "plaintext") into a form (called "cipher text") that conceals the data's original meaning to prevent it from being known or used.
Encryption algorithm	Set of mathematically expressed rules for rendering data unintelligible by executing a series of conversions controlled by a key.
End-item accounting	Accounting for all the accountable components of a COMSEC equipment configuration by a single short title.
End-to-end encryption	Encryption of information at its origin and decryption at its intended destination without intermediate decryption.
End-to-end security	Safeguarding information in an IS from point of origin to point of destination.
Endorsed for unclassified cryptographic item (EUCI) (C.F.D.)	Unclassified cryptographic equipment that embodies a U.S. Government classified cryptographic logic and is endorsed by NSA for the protection of national security information. (See type 2 product.)
Endorsement (C.F.D.)	NSA approval of a commercially developed product for safeguarding national security information.
Entrapment	Deliberate planting of apparent flaws in an IS for the purpose of detecting attempted penetrations.
Environment	Aggregate of external procedures, conditions, and objects affecting the development, operation, and maintenance of an IS.
Ephemeral Port	Also called a transient port or a temporary port. Usually is on the client side. It is set up when a client application wants to connect to a server and is destroyed when the client application terminates. It has a number chosen at random that is greater than 1023.
Erasure	Process intended to render magnetically stored information irretrievable by normal means.
Escrow Passwords	Escrow Passwords are passwords that are written down and stored in a secure location (like a safe) that are used by emergency personnel when privileged personnel are unavailable.
Ethernet	The most widely-installed LAN technology. Specified in a standard, IEEE 802.3, an Ethernet LAN typically uses coaxial cable or special grades of twisted pair wires. Devices are

	connected to the cable and compete for access using a CSMA/CD protocol.
Evaluation assurance level (EAL)	Set of assurance requirements that represent a point on the Common Criteria predefined assurance scale.
Event	An event is an observable occurrence in a system or network. Occurrence, not yet assessed, that may affect the performance of an IS.
Executive state (C.F.D.)	One of several states in which an IS may operate, and the only one in which certain privileged instructions may be executed. Such privileged instructions cannot be executed when the system is operating in other states. Synonymous with supervisor state.
Exercise key	Key used exclusively to safeguard communications transmitted over-the-air during military or organized civil training exercises.
Exploitable channel	Channel that allows the violation of the security policy governing an IS and is usable or detectable by subjects external to the trusted computing base. (See covert channel.)
Exponential Backoff Algorithm	An exponential backoff algorithm is used to adjust TCP timeout values on the fly so that network devices don't continue to timeout sending data over saturated links.
Exposure	A threat action whereby sensitive data is directly released to an unauthorized entity.
Extended ACLs (Cisco)	Extended ACLs are a more powerful form of Standard ACLs on Cisco routers. They can make filtering decisions based on IP addresses (source or destination), Ports (source or destination), protocols, and whether a session is established.
Extensible Authentication Protocol (EAP)	A framework that supports multiple, optional authentication mechanisms for PPP, including clear-text passwords, challenge-response, and arbitrary dialog sequences.
Exterior Gateway Protocol (EGP)	A protocol which distributes routing information to the routers which connect autonomous systems.
Extraction resistance	Capability of crypto-equipment or secure telecommunications equipment to resist efforts to extract key.
Extranet	Extension to the intranet allowing selected outside users access to portions of an organization's intranet.
Facial Thermogram	Facial thermogram systems analyze the face size and heat patterns to determine the identity of a person seeking access.

Fail safe	Automatic protection of programs and/or processing systems when hardware or software failure is detected.
Fail soft (C.F.D.)	Selective termination of affected nonessential processing when hardware or software failure is determined to be imminent.
Failure access	Type of incident in which unauthorized access to data results from hardware or software failure.
Failure control	Methodology used to detect imminent hardware or software failure and provide fail safe or fail soft recovery.
False Rejects	False Rejects are when an authentication system fails to recognize a valid user.
Fast File System	The first major revision to the Unix file system, providing faster read access and faster (delayed, asynchronous) write access through a disk cache and better file system layout on disk. It uses inodes (pointers) and data blocks.
Fault Line Attacks	Fault Line Attacks use weaknesses between interfaces of systems to exploit gaps in coverage.
File protection	Aggregate of processes and procedures designed to inhibit unauthorized access, contamination, elimination, modification, or destruction of a file or any of its contents.
File security	Means by which access to computer files is limited to authorized users only.
File Transfer Protocol (FTP)	A TCP/IP protocol specifying the transfer of text or binary files across the network.
Fill device	COMSEC item used to transfer or store key in electronic form or to insert key into a crypto-equipment.
Filter	A filter is used to specify which packets will or will not be used. It can be used in sniffers to determine which packets get displayed, or by firewalls to determine which packets get blocked.
Filtering Router	An inter-network router that selectively prevents the passage of data packets according to a security policy. A filtering router may be used as a firewall or part of a firewall. A router usually receives a packet from a network and decides where to forward it on a second network. A filtering router does the same, but first decides whether the packet should be forwarded at all, according to some security policy. The policy is implemented by rules (packet filters) loaded into the router.

Finger	A protocol to lookup user information on a given host. A Unix program that takes an e-mail address as input and returns information about the user who owns that email address. On some systems, finger only reports whether the user is currently logged on. Other systems return additional information, such as the user's full name, address, and telephone number. Of course, the user must first enter this information into the system. Many email programs now have a finger utility built into them.
Fingerprinting	Sending strange packets to a system in order to gauge how it responds to determine the operating system.
FIREFLY	Key management protocol based on public key cryptography.
Firewall	A logical or physical discontinuity in a network to prevent unauthorized access to data or resources.
Firmware	Program recorded in permanent or semipermanent computer memory.
Fixed COMSEC facility	COMSEC facility located in an immobile structure or aboard a ship.
Flaw	Error of commission, omission, or oversight in an IS that may allow protection mechanisms to be bypassed.
Flaw hypothesis methodology	System analysis and penetration technique in which the specification and documentation for an IS are analyzed to produce a list of hypothetical flaws. This list is prioritized on the basis of the estimated probability that a flaw exists, on the ease of exploiting it, and on the extent of control or compromise it would provide. The prioritized list is used to perform penetration testing of a system.
Flooding	An attack that attempts to cause a failure in (especially, in the security of) a computer system or other data processing entity by providing more input than the entity can process properly.
Forest	A forest is a set of Active Directory domains that replicate their databases with each other.
Fork Bomb	A Fork Bomb works by using the fork() call to create a new process which is a copy of the original. By doing this repeatedly, all available processes on the machine can be taken up.
Form-based Authentication	Form-based Authentication uses forms on a webpage to ask a user to input username and password information.

Formal access approval	Process for authorizing access to classified or sensitive information with specified access requirements, such as Sensitive Compartmented Information (SCI) or Privacy Data, based on the specified access requirements and a determination of the individual's security eligibility and need-to-know.
Formal development methodology	Software development strategy that proves security design specifications.
Formal method	Mathematical argument which verifies that the system satisfies a mathematically described security policy.
Formal proof (C.F.D.)	Complete and convincing mathematical argument presenting the full logical justification for each proof step and for the truth of a theorem or set of theorems.
Formal security policy	Mathematically precise statement of a security policy.
Formal top-level specification (C.F.D.)	Top-level specification written in a formal mathematical language to allow theorems, showing the correspondence of the system specification to its formal requirements, to be hypothesized and formally proven.
Formal verification (C.F.D.)	Process of using formal proofs to demonstrate the consistency between formal specification of a system and formal security policy model (design verification) or between formal specification and its high-level program implementation (implementation verification).
Forward Lookup	Forward lookup uses an Internet domain name to find an IP address
Forward Proxy	Forward Proxies are designed to be the server through which all requests are made.
Fragment Offset	The fragment offset field tells the sender where a particular fragment falls in relation to other fragments in the original larger packet.
Fragment Overlap Attack	A TCP/IP Fragmentation Attack that is possible because IP allows packets to be broken down into fragments for more efficient transport across various media. The TCP packet (and its header) is carried in the IP packet. In this attack the second fragment contains incorrect offset. When packet is reconstructed, the port number will be overwritten.
Fragmentation	The process of storing a data file in several

	"chunks" or fragments rather than in a single contiguous sequence of bits in one place on the storage medium.
Frames	Data that is transmitted between network points as a unit complete with addressing and necessary protocol control information. A frame is usually transmitted serial bit by bit and contains a header field and a trailer field that "frame" the data. (Some control frames contain no data.)
Frequency hopping	Repeated switching of frequencies during radio transmission according to a specified algorithm, to minimize unauthorized interception or jamming of telecommunications.
Front-end security filter (C.F.D.)	Security filter logically separated from the remainder of an IS to protect system integrity. Synonymous with firewall.
Full Duplex	A type of duplex communications channel which carries data in both directions at once. Refers to the transmission of data in two directions simultaneously. Communications in which both sender and receiver can send at the same time.
Fully-Qualified Domain Name	A Fully-Qualified Domain Name is a server name with a hostname followed by the full domain name.
Full maintenance	Complete diagnostic repair, modification, and overhaul of COMSEC equipment, including repair of defective assemblies by piece part replacement. (See limited maintenance.)
Functional proponent (C.F.D.)	See network sponsor.
Functional testing	Segment of security testing in which advertised security mechanisms of an IS are tested under operational conditions.
Gateway	Interface providing compatibility between networks by converting transmission speeds, protocols, codes, or security measures.
gethostbyaddr	The gethostbyaddr DNS query is when the address of a machine is known and the name is needed.
gethostbyname	The gethostbyname DNS quest is when the name of a machine is known and the address is needed.
Global Information Grid	The globally interconnected, end-to-end set of information capabilities, associated processes, and personnel for collecting, processing, storing, disseminating, and managing information on demand to war fighters, policy makers, and support personnel. (DoD Directive 8100.1, 19 Sept. 2002)
Global information Infrastructure (GII)	Worldwide interconnections of the information

	systems of all countries, international and multinational organizations, and international commercial communications.
GNU	GNU is a Unix-like operating system that comes with source code that can be copied, modified, and redistributed. The GNU project was started in 1983 by Richard Stallman and others, who formed the Free Software Foundation.
Gnutella	An Internet file sharing utility. Gnutella acts as a server for sharing files while simultaneously acting as a client that searches for and downloads files from other users.
Guard	Mechanism limiting the exchange of information between systems.
Hacker	Unauthorized user who attempts to or gains access to an IS.
Handshaking procedures	Dialogue between two IS's for synchronizing, identifying, and authenticating themselves to one another.
Hard copy key	Physical keying material, such as printed key lists, punched or printed key tapes, or programmable, read-only memories (PROM).
Hardening	Hardening is the process of identifying and fixing vulnerabilities on a system.
Hardwired key	Permanently installed key.
Hash Function	An algorithm that computes a value based on a data object thereby mapping the data object to a smaller data object.
Hash total	Value computed on data to detect error or manipulation. (See checksum.)
Hashing	Computation of a hash total.
Hashword	Memory address containing hash total.
Header	A header is the extra information in a packet that is needed for the protocol stack to process the packet.
High assurance guard (HAG)	Device comprised of both hardware and software that is designed to enforce security rules during the transmission of X.400 message and X.500 directory traffic between enclaves of different classification levels (e.g., UNCLASSIFIED and SECRET).
Hijack Attack	A form of active wiretapping in which the attacker seizes control of a previously established communication association.
Honeypot	Programs that simulate one or more network services that you designate on your computer's ports. An attacker assumes you're running

	vulnerable services that can be used to break into the machine. A honey pot can be used to log access attempts to those ports including the attacker's keystrokes. This could give you advanced warning of a more concerted attack.
Hops	A hop is each exchange with a gateway a packet takes on its way to the destination.
Host	Any computer that has full two-way access to other computers on the Internet. Or a computer with a web server that serves the pages for one or more Web sites.
Host-Based ID	Host-based intrusion detection systems use information from the operating system audit records to watch all operations occurring on the host that the intrusion detection software has been installed upon. These operations are then compared with a pre-defined security policy. This analysis of the audit trail imposes potentially significant overhead requirements on the system because of the increased amount of processing power which must be utilized by the intrusion detection system. Depending on the size of the audit trail and the processing ability of the system, the review of audit data could result in the loss of a real-time analysis capability.
HTTP Proxy	An HTTP Proxy is a server that acts as a middleman in the communication between HTTP clients and servers.
HTTPS	When used in the first part of a URL (the part that precedes the colon and specifies an access scheme or protocol), this term specifies the use of HTTP enhanced by a security mechanism, which is usually SSL.
Hub	A hub is a network device that operates by repeating data that it receives on one port to all the other ports. As a result, data transmitted by one host is retransmitted to all other hosts on the hub.
Hybrid Attack	A Hybrid Attack builds on the dictionary attack method by adding numerals and symbols to dictionary words.
Hybrid Encryption	An application of cryptography that combines two or more encryption algorithms, particularly a combination of symmetric and asymmetric encryption.
Hyperlink	In hypertext or hypermedia, an information

	object (such as a word, a phrase, or an image; usually highlighted by color or underscoring) that points (indicates how to connect) to related information that is located elsewhere and can be retrieved by activating the link.
Hypertext Markup Language (HTML)	The set of markup symbols or codes inserted in a file intended for display on a World Wide Web browser page.
Hypertext Transfer Protocol (HTTP)	The protocol in the Internet Protocol (IP) family used to transport hypertext documents across an internet.
IA architecture	Activity that aggregates the functions of developing IA operational, system, and technical architecture products for the purpose of specifying and implementing new or modified IA capabilities within the IT environment. (DoD Directive 8100.1, 19 Sept 2002)
IA-enabled information technology product	Product or technology whose primary role is not security, but which provides security services as an associated feature of its intended operating capabilities. Examples include such products as security-enabled web browsers, screening routers, trusted operating systems, and security-enabled messaging systems.
Identification	Process an IS uses to recognize an entity.
Identity	Identity is whom someone or what something is, for example, the name by which something is known.
Identity token	Smart card, metal key, or other physical object used to authenticate identity.
Identity validation	Tests enabling an IS to authenticate users or resources.
Imitative communications deception	Introduction of deceptive messages or signals into an adversary's telecommunications signals. (See communications deception and manipulative communications deception.)
Impersonating	Form of spoofing.
Implant	Electronic device or electronic equipment modification designed to gain unauthorized interception of information-bearing emanations.
Inadvertant disclosure	Type of incident involving accidental exposure of information to an individual not authorized access.
Incident	(IS) Assessed occurrence having actual or potentially adverse effects on an IS. (COMSEC) Occurrence that potentially jeopardizes the security of COMSEC material or

	the secure electrical transmission of national security information.
Incident Handling	Incident Handling is an action plan for dealing with intrusions, cyber-theft, denial of service, fire, floods, and other security-related events. It is comprised of a six step process: Preparation, Identification, Containment, Eradication, Recovery, and Lessons Learned.
Incomplete parameter checking	System flaw that exists when the operating system does not check all parameters fully for accuracy and consistency, thus making the system vulnerable to penetration.
Incremental Backups	Incremental backups only backup the files that have been modified since the last backup. If dump levels are used, incremental backups only backup files changed since last backup of a lower dump level.
Indicator	Recognized action, specific, generalized, or theoretical, that an adversary might be expected to take in preparation for an attack.
Individual accountability	Ability to associate positively the identity of a user with the time, method, and degree of access to an IS.
Inetd (xinetd)	Inetd (or Internet Daemon) is an application that controls smaller internet services like telnet, ftp, and POP.
Inference Attack	Inference Attacks rely on the user to make logical connections between seemingly unrelated pieces of information.
Informal security policy	Natural language description, possibly supplemented by mathematical arguments, demonstrating the correspondence of the functional specification to the high-level design.
Information assurance (IA)	Measures that protect and defend information and information systems by ensuring their availability, integrity, authentication, confidentiality, and non-repudiation. These measures include providing for restoration of information systems by incorporating protection, detection, and reaction capabilities.
Information assurance manager (IAM)	See information systems security manager.
Information assurance officer (IAO)	See information systems security officer.
Information assurance product	Product or technology whose primary purpose is to provide security services (e.g., confidentiality, authentication, integrity, access control, non-repudiation of data) correct known vulnerabilities; and/or provide layered defense against various categories of non-authorized or

	malicious penetrations of information systems or networks. Examples include such products as data/network encryptors, firewalls, and intrusion detection devices.
Information environment	Aggregate of individuals, organizations, or systems that collect, process, or disseminate information, also included is the information itself.
Information flow control	Procedure to ensure that information transfers within an IS are not made from a higher security level object to an object of a lower security level.
Information Operations (IO)	Actions taken to affect adversary information and ISs while defending one's own information and ISs.
Information owner	Official with statutory or operational authority for specified information and responsibility for establishing the controls for its generation, collection, processing, dissemination, and disposal.
Information security policy	Aggregate of directives, regulations, rules, and practices that prescribe how an organization manages, protects, and distributes information.
Information system (IS)	Set of information resources organized for the collection, storage, processing, maintenance, use, sharing, dissemination, disposition, display, or transmission of information.
Information systems security (INFOSEC)	Protection of information systems against unauthorized access to or modification of information, whether in storage, processing or transit, and against the denial of service to authorized users, including those measures necessary to detect, document, and counter such threats.
Information systems security engineering (ISSE)	Process that captures and refines information protection requirements and ensures their integration into IT acquisition processes through purposeful security design or configuration.
Information systems security engineering modification	Modification of any fielded hardware, firmware, software, or portion thereof, under NSA configuration control. There are three classes of modifications: mandatory (to include human safety); optional/special mission modifications; and repair actions. These classes apply to elements, subassemblies, equipment, systems, and software packages performing functions such as key generation, key distribution, message encryption, decryption, authentication, or those mechanisms necessary to satisfy security policy, labeling, identification, or accountability.

Information systems security manager (ISSM)	Individual responsible for a program, organization, system, or enclave's information assurance program.
Information systems security officer (ISSO)	Individual responsible to the ISSM for ensuring the appropriate operational IA posture is maintained for a system, program, or enclave.
Information systems security product	Item (chip, module, assembly, or equipment), technique, or service that performs or relates to information systems security.
Infrastructure Architecture	Infrastructure Architecture is the wireless network mode where a centralized base station provides access to a networking medium and handles security for many hosts.
Ingress Filtering	Ingress Filtering is filtering inbound traffic.
Initialize	Setting the state of a cryptographic logic prior to key generation, encryption, or other operating mode.
Inspectable space	Three dimensional space surrounding equipment that process classified and/or sensitive information within which TEMPEST exploitation is not considered practical or where legal authority to identify and remove a potential TEMPEST exploitation exists. Synonymous with zone of control.
Integrity	Quality of an IS reflecting the logical correctness and reliability of the operating system; the logical completeness of the hardware and software implementing the protection mechanisms; and the consistency of the data structures and occurrence of the stored data. Note that, in a formal security mode, integrity is interpreted more narrowly to mean protection against unauthorized modification or destruction of information.
Integrity check value	Checksum capable of detecting modification of an IS.
Interrupt	An Interrupt is a signal that informs the OS that something has occurred.
Information Warfare	Information Warfare is the competition between offensive and defensive players over information resources.
Input Validation Attacks	Input Validation Attacks are where an attacker intentionally sends unusual input in the hopes of confusing an application.
Integrity	Integrity is the need to ensure that information has not been changed accidentally or deliberately, and that it is accurate and complete.
Integrity Star Property	In Integrity Star Property a user cannot read data

	of a lower integrity level than their own.
Interconnection security agreement	Written management authorization to interconnect information systems based upon acceptance of risk and implementation of established controls.
Interface	Common boundary between independent systems or modules where interactions take place.
Interface control document	Technical document describing interface controls and identifying the authorities and responsibilities for ensuring the operation of such controls. This document is baselined during the preliminary design review and is maintained throughout the IS lifecycle.
Interim approval to operate (IA TO)	Temporary authorization granted by a DAA for an IS to process information based on preliminary results of a security evaluation of the system.
Interim approval to test (IATT)	Temporary authorization to test an information system in a specified operational information environment within the timeframe and under the conditions or constraints enumerated in the written authorization.
Internal security controls	Hardware, firmware, or software features within an IS that restrict access to resources only to authorized subjects.
Internet	A term to describe connecting multiple separate networks together.
Internet Control Message Protocol (ICMP)	An Internet Standard protocol that is used to report error conditions during IP datagram processing and to exchange other information concerning the state of the IP network.
Internet Engineering Task Force (IETF)	The body that defines standard Internet operating protocols such as TCP/IP. The IETF is supervised by the Internet Society Internet Architecture Board (IAB). IETF members are drawn from the Internet Society's individual and organization membership.
Internet Message Access Protocol (IMAP)	A protocol that defines how a client should fetch mail from and return mail to a mail server. IMAP is intended as a replacement for or extension to the Post Office Protocol (POP). It is defined in RFC 1203 (v3) and RFC 2060 (v4).
Internet Protocol (IP)	Standard protocol for transmission of data from source to destinations in packet-switched communications networks and interconnected systems of such networks.
Internet Protocol Security (IPsec)	A developing standard for security at the

	network or packet processing layer of network communication.
Internet Standard	A specification, approved by the IESG and published as an RFC, that is stable and well-understood, is technically competent, has multiple, independent, and interoperable implementations with substantial operational experience, enjoys significant public support, and is recognizably useful in some or all parts of the Internet.
Internetwork private line interface (C.F.D.)	Network cryptographic unit that provides secure connections, singularly or in simultaneous multiple connections, between a host and a predetermined set of corresponding hosts.
Intranet	A computer network, especially one based on Internet technology, that an organization uses for its own internal, and usually private, purposes and that is closed to outsiders.
Intruder	A malicious individual who has accessed resources that they are unauthorized to (i.e. a hacker gaining root access on a system), or in excess of the authorization they have.
Intrusion	Unauthorized act of bypassing the security mechanisms of a system.
Intrusion Detection	A security management system for computers and networks. An IDS gathers and analyzes information from various areas within a computer or a network to identify possible security breaches, which include both intrusions (attacks from outside the organization) and misuse (attacks from within the organization).
IP Address	A computer's inter-network address that is assigned for use by the Internet Protocol and other protocols. An IP version 4 address is written as a series of four 8-bit numbers separated by periods.
IP Flood	A denial of service attack that sends a host more echo request ("ping") packets than the protocol implementation can handle.
IP Forwarding	IP forwarding is an Operating System option that allows a host to act as a router. A system that has more than 1 network interface card must have IP forwarding turned on in order for the system to be able to act as a router.
IP Spoofing	The technique of supplying a false IP address.
ISO	International Organization for Standardization, a

	voluntary, non-treaty, non-government organization, established in 1947, with voting members that are designated standards bodies of participating nations and non-voting observer organizations.
Issue-Specific Policy	An Issue-Specific Policy is intended to address specific needs within an organization, such as a password policy.
ITU-T	International Telecommunications Union, Telecommunication Standardization Sector (formerly "CCITT"), a United Nations treaty organization that is composed mainly of postal, telephone, and telegraph authorities of the member countries and that publishes standards called "Recommendations."
Jitter	Jitter or Noise is the modification of fields in a database while preserving the aggregate characteristics of that make the database useful in the first place.
Jump Bag	A Jump Bag is a container that has all the items necessary to respond to an incident inside to help mitigate the effects of delayed reactions.
Kerberos	A system developed at the Massachusetts Institute of Technology that depends on passwords and symmetric cryptography (DES) to implement ticket-based, peer entity authentication service and access control service distributed in a client-server network environment.
Kernel	The essential center of a computer operating system, the core that provides basic services for all other parts of the operating system. A synonym is nucleus. A kernel can be contrasted with a shell, the outermost part of an operating system that interacts with user commands. Kernel and shell are terms used more frequently in Unix and some other operating systems than in IBM mainframe systems.
Key	Usually a sequence of random or pseudorandom bits used initially to set up and periodically change the operations performed in crypto-equipment for the purpose of encrypting or decrypting electronic signals, or for determining electronic counter-countermeasures patterns, or for producing other key.
Key-auto-key (KAK)	Cryptographic logic using previous key to produce key.
Key distribution center (KDC)	COMSEC facility generating and distributing

	key in electrical form.
Key-encryption-key (KEK)	Key that encrypts or decrypts other key for transmission or storage.
Key exchange	Process of exchanging public keys (and other information) in order to establish secure communications.
Key list	Printed series of key settings for a specific cryptonet. Key lists may be produced in list, pad, or printed tape format.
Key management infrastructure (KMI)	Framework and services that provide the generation, production, storage, protection, distribution, control, tracking, and destruction for all cryptographic key material, symmetric keys as well as public keys and public key certificates.
Key pair	Public key and its corresponding private key as used in public key cryptography.
Key production key (KPK)	Key used to initialize a keystream generator for the production of other electronically generated key.
Key recovery	Mechanisms and processes that allow authorized parties to retrieve the cryptographic key used for data confidentiality.
Key stream	Sequence of symbols (or their electrical or mechanical equivalents) produced in a machine or auto-manual cryptosystem to combine with plain text to produce cipher text, control transmission security processes, or produce key.
Key tag	Identification information associated with certain types of electronic key.
Key tape	Punched or magnetic tape containing key. Printed key in tape form is referred to as a key list.
Key updating	Irreversible cryptographic process for modifying key.
Keying material	Key, code, or authentication information in physical or magnetic form.
Label	See security label.
Labeled security protections (C.F.D.)	Elementary-level mandatory access control protection features and intermediate-level discretionary access control features in a TCB that uses sensitivity labels to make access control decisions.
Laboratory attack	Use of sophisticated signal recovery equipment in a laboratory environment to recover information from data storage media.
Lattice Techniques	Lattice Techniques use security designations to

	determine access to information.
Layer 2 Forwarding Protocol (L2F)	An Internet protocol (originally developed by Cisco Corporation) that uses tunneling of PPP over IP to create a virtual extension of a dial-up link across a network, initiated by the dial-up server and transparent to the dial-up user.
Layer 2 Tunneling Protocol (L2TP)	An extension of the Point-to-Point Tunneling Protocol used by an Internet service provider to enable the operation of a virtual private network over the Internet.
Least Privilege	Principle requiring that each subject be granted the most restrictive set of privileges needed for the performance of authorized tasks. Application of this principle limits the damage that can result from accident, error, or unauthorized use of an IS..
Legion	Software to detect unprotected shares.
Level of concern	Rating assigned to an IS indicating the extent to which protection measures, techniques, and procedures must be applied. High, Medium, and Basic are identified levels of concern. A separate Level-of-Concern is assigned to each IS for confidentiality, integrity, and availability.
Level of protection	Extent to which protective measures, techniques, and procedures must be applied to ISs and networks based on risk, threat, vulnerability, system interconnectivity considerations, and information assurance needs. Levels of protection are: 1. Basic: IS and networks requiring implementation of standard minimum security countermeasures. 2. Medium: IS and networks requiring layering of additional safeguards above the standard minimum security countermeasures. 3. High: IS and networks requiring the most stringent protection and rigorous security countermeasures.
Lightweight Directory Access Protocol (LDAP)	A software protocol for enabling anyone to locate organizations, individuals, and other resources such as files and devices in a network, whether on the public Internet or on a corporate intranet.
Limited maintenance	COMSEC maintenance restricted to fault isolation, removal, and replacement of plug-in assemblies. Soldering or unsoldering usually is prohibited in limited maintenance. (See full maintenance.)
Line conditioning	Elimination of unintentional signals or noise induced or conducted on a telecommunications or IS signal, power, control, indicator, or other

	external interface line.
Line conduction	Unintentional signals or noise induced or conducted on a telecommunications or IS signal, power, control, indicator, or other external interface line.
Link encryption	Encryption of information between nodes of a communications system.
Link State	With link state, routes maintain information about all routers and router-to-router links within a geographic area, and creates a table of best routes with that information.
List Based Access Control	List Based Access Control associates a list of users and their privileges with each object.
List-oriented	IS protection in which each protected object has a list of all subjects authorized to access it.
Loadable Kernel Modules (LKM)	Loadable Kernel Modules allow for the adding of additional functionality directly into the kernel while the system is running.
Local authority	Organization responsible for generating and signing user certificates.
Local Management Device/Key Processor (LMD/KP)	EKMS platform providing automated management of COMSEC material and generating key for designated users.
Lock and key protection system (C.F.D.)	Protection system that involves matching a key or password with a specific access requirement.
Logic Bomb	Resident computer program triggering an unauthorized act when particular states of an IS are realized.
Logical completeness measure	Means for assessing the effectiveness and degree to which a set of security and access control mechanisms meets security specifications.
Log Clipping	Log clipping is the selective removal of log entries from a system log to hide a compromise.
Logic Gate	A logic gate is an elementary building block of a digital circuit. Most logic gates have two inputs and one output. As digital circuits can only understand binary, inputs and outputs can assume only one of two states, 0 or 1.
Long title	Descriptive title of a COMSEC item.
Loopback Address	The loopback address (127.0.0.1) is a pseudo IP address that always refers back to the local host and are never sent out onto a network.
Low probability of detection	Result of measures used to hide or disguise intentional electromagnetic transmissions.
Low probability of intercept	Result of measures to prevent the intercept of intentional electromagnetic transmissions.

MAC Address	A physical address; a numeric value that uniquely identifies that network device from every other device on the planet.
Magnetic remanence	Magnetic representation of residual information remaining on a magnetic medium after the medium has been cleared. (See clearing.)
Maintenance hook	Special instructions (trapdoors) in software allowing easy maintenance and additional feature development. Since maintenance hooks frequently allow entry into the code without the usual checks, they are a serious security risk if they are not removed prior to live implementation.
Maintenance key	Key intended only for in-shop use.
Malicious applets	Small application programs automatically downloaded and executed that perform an unauthorized function on an IS.
Malicious Code	Software (e.g., Trojan horse) that appears to perform a useful or desirable function, but actually gains unauthorized access to system resources or tricks a user into executing other malicious logic.
Malicious Logic	Hardware, software, or firmware capable of performing an unauthorized function on an IS.
Malware	A generic term for a number of different types of malicious code.
Mandatory Access Control (MAC)	Means of restricting access to objects based on the sensitivity of the information contained in the objects and the formal authorization (i.e., clearance, formal access approvals, and need-to-know) of subjects to access information of such sensitivity. (See discretionary access control.)
Mandatory modification	Change to a COMSEC end-item that NSA requires to be completed and reported by a specified date. (See optional modification.)
Manipulative communications deception	Alteration or simulation of friendly telecommunications for the purpose of deception. (See communications deception and imitative communications deception.)
Manual cryptosystem	Cryptosystem in which the cryptographic processes are performed without the use of crypto-equipment or auto-manual devices.
Manual remote rekeying	Procedure by which a distant crypto-equipment is rekeyed electrically, with specific actions required by the receiving terminal operator. Synonymous with cooperative remote rekeying. (Also see automatic remote keying.)

Masquerade Attack	A type of attack in which one system entity illegitimately poses as (assumes the identity of) another entity.
Masquerading	See spoofing.
Master crypto-ignition key	Key device with electronic logic and circuits providing the capability for adding more operational CIKs to a keyset.
Measures of Effectiveness (MOE)	Measures of Effectiveness is a probability model based on engineering concepts that allows one to approximate the impact a give action will have on an environment. In Information warfare it is the ability to attack or defend within an Internet environment.
Memory scavenging	The collection of residual information from data storage.
Message authentication code	Data associated with an authenticated message allowing a receiver to verify the integrity of the message.
Message externals	Information outside of the message text, such as the header, trailer, etc.
Message indicator	Sequence of bits transmitted over a communications system for synchronizing crypto-equipment. Some off-line cryptosystems, such as the KL-51 and one-time pad systems, employ message indicators to establish decryption starting points.
Mimicking	See spoofing.
Mobile code	Software modules obtained from remote systems, transferred across a network, and then downloaded and executed on local systems without explicit installation or execution by the recipient.
Mode of operation	Description of the conditions under which an IS operates based on the sensitivity of information processed and the clearance levels, formal access approvals, and need-to-know of its users. Four modes of operation are authorized for processing or transmitting information: dedicated mode, system-high mode, compartmented/partitioned mode, and multilevel mode.
Monoculture	Monoculture is the case where a large number of users run the same software, and are vulnerable to the same attacks.
Morris Worm	A worm program written by Robert T. Morris, Jr. that flooded the ARPANET in November, 1988, causing problems for thousands of hosts.
Multi-cast	Broadcasting from one host to a given set of hosts.

Multi-homed	You are "multi-homed" if your network is directly connected to two or more ISP's.
Multilevel device	Equipment trusted to properly maintain and separate data of different security categories.
Multilevel mode	INFOSEC mode of operation wherein all the following statements are satisfied concerning the users who have direct or indirect access to the system, its peripherals, remote terminals, or remote hosts: a. some users do not have a valid security clearance for all the information processed in the IS; b. all users have the proper security clearance and appropriate formal access approval for that information to which they have access; and c. all users have a valid need-to-know only for information to which they have access.
Multilevel security (MLS)	Concept of processing information with different classifications and categories that simultaneously permits access by users with different security clearances and denies access to users who lack authorization. (See cross domain solution.)
Multiplexing	To combine multiple signals from possibly disparate sources, in order to transmit them over a single path.
Multi-security level (MSL)	Capability to process information of different security classifications or categories by using periods processing or peripheral sharing.
Mutual suspicion	Condition in which two ISs need to rely upon each other to perform a service, yet neither trusts the other to properly protect shared data.
National Information Assurance Partnership (NIAP)	Joint initiative between NSA and NIST responsible for security testing needs of both IT consumers and producers and promoting the development of technically sound security requirements for IT products and systems and appropriate measures for evaluating those products and systems.
National Information Infrastructure (NII)	Nationwide interconnection of communications networks, computers, databases, and consumer electronics that make vast amounts of information available to users. It includes both public and private networks, the internet, the public switched network, and cable, wireless, and satellite communications.
National Institute of Standards and Technology (NIST)	National Institute of Standards and Technology, a unit of the US Commerce Department. Formerly known as the National Bureau of Standards, NIST promotes and maintains measurement standards. It also has active

	programs for encouraging and assisting industry and science to develop and use these standards.
Natural Disaster	Any "act of God" (e.g., fire, flood, earthquake, lightning, or wind) that disables a system component.
National security information (NSI)	Information that has been determined, pursuant to Executive Order 12958 (as amended) (Ref b.) or any predecessor order, to require protection against unauthorized disclosure.
National security system	<p>Any information system (including any telecommunications system) used or operated by an agency or by a contractor of any agency, or other organization on behalf of an agency, the function, operation, or use of which: I. involves intelligence activities; II. involves cryptologic activities related to national security; III. involves command and control of military forces; IV. involves equipment that is an integral part of a weapon or weapon system; or V. subject to subparagraph (B), is critical to the direct fulfillment of military or intelligence missions; or is protected at all times by procedures established for information that have been specifically authorized under criteria established by an Executive Order or an Act of Congress to be kept classified in the interest of national defense or foreign policy.</p> <p>(B). Does not include a system that is to be used for routine administrative and business applications (including payroll, finance, logistics, and personnel management applications).</p> <p>(Title 44 U.S. Code Section 3542, Federal Information Security Management Act of 2002.)</p>
Need-to-know	Necessity for access to, or knowledge or possession of, specific official information required to carry out official duties.
Need-to-know determination	Decision made by an authorized holder of official information that a prospective recipient requires access to specific official information to carry out official duties.
Netmask	32-bit number indicating the range of IP addresses residing on a single IP network/subnet/supernet. This specification displays network masks as hexadecimal numbers. For example, the network mask for a class C IP network is displayed as 0xffffffff00. Such a mask is often displayed elsewhere in the

Network	literature as 255.255.255.0. IS implemented with a collection of interconnected nodes.
Network Address Translation	The translation of an Internet Protocol address used within one network to a different IP address known within another network. One network is designated the inside network and the other is the outside.
Network-based IDS	A network-based IDS system monitors the traffic on its network segment as a data source. This is generally accomplished by placing the network interface card in promiscuous mode to capture all network traffic that crosses its network segment. Network traffic on other segments and traffic on other means of communication (like phone lines) can't be monitored. Network-based IDS involves looking at the packets on the network as they pass by some sensor. The sensor can only see the packets that happen to be carried on the network segment it's attached to. Packets are considered to be of interest if they match a signature. Network-based intrusion detection passively monitors network activity for indications of attacks. Network monitoring offers several advantages over traditional host-based intrusion detection systems. Because many intrusions occur over networks at some point, and because networks are increasingly becoming the targets of attack, these techniques are an excellent method of detecting many attacks which may be missed by host-based intrusion detection mechanisms.
Network front-end	Device implementing protocols that allow attachment of a computer system to a network.
Network Mapping	To compile an electronic inventory of the systems and the services on your network.
Network reference monitor	See reference monitor.
Network security	See information systems security.
Network security officer	See information systems security officer.
Network sponsor	Individual or organization responsible for stating the security policy enforced by the network, designing the network security architecture to properly enforce that policy, and ensuring the network is implemented in such a way that the policy is enforced.

Network system	System implemented with a collection of interconnected components. A network system is based on a coherent security architecture and design.
Network Taps	Network taps are hardware devices that hook directly onto the network cable and send a copy of the traffic that passes through it to one or more other networked devices.
Network weaving	Penetration technique in which different communication networks are linked to access an IS to avoid detection and trace-back.
No Lone Zone	Area, room, or space that, when staffed, must be occupied by two or more appropriately cleared individuals who remain within sight of each other. (See two-person integrity.)
Non-printable Character	A character that doesn't have a corresponding character letter to its corresponding ASCII code. Examples would be the Linefeed, which is ASCII character code 10 decimal, the Carriage Return, which is 13 decimal, or the bell sound, which is decimal 7. On a PC, you can often add non-printable characters by holding down the Alt key, and typing in the decimal value (i.e., Alt-007 gets you a bell). There are other character encoding schemes, but ASCII is the most prevalent.
Non-repudiation	Assurance the sender of data is provided with proof of delivery and the recipient is provided with proof of the sender's identity, so neither can later deny having processed the data.
Null	Dummy letter, letter symbol, or code group inserted into an encrypted message to delay or prevent its decryption or to complete encrypted groups for transmission or transmission security purposes.
Null Session	Known as Anonymous Logon, it is a way of letting an anonymous user retrieve information such as user names and shares over the network or connect without authentication. It is used by applications such as explorer.exe to enumerate shares on remote servers.
Object	Passive entity containing or receiving information. Access to an object implies access to the information it contains.
Object reuse	Reassignment and re-use of a storage medium containing one or more objects after ensuring no residual data remains on the storage medium.
Octet	A sequence of eight bits. An octet is an eight-bit byte.

Official information	All information in the custody and control of a U.S. Government department or agency that was acquired by U.S. Government employees as a part of their official duties or because of their official status and has not been cleared for public release.
Off-line cryptosystem	Cryptosystem in which encryption and decryption are performed independently of the transmission and reception functions.
One-part code	Code in which plain text elements and their accompanying code groups are arranged in alphabetical, numerical, or other systematic order, so one listing serves for both encoding and decoding. One-part codes are normally small codes used to pass small volumes of low-sensitivity information.
One-time cryptosystem	Cryptosystem employing key used only once.
One-time pad	Manual one-time cryptosystem produced in pad form.
One-time tape	Punched paper tape used to provide key streams on a one-time basis in certain machine cryptosystems.
On-line cryptosystem	Cryptosystem in which encryption and decryption are performed in association with the transmitting and receiving functions.
One-way Encryption	Irreversible transformation of plaintext to cipher text, such that the plaintext cannot be recovered from the cipher text by other than exhaustive procedures even if the cryptographic key is known.
One-way Function	A (mathematical) function, f, which is easy to compute the output based on a given input. However given only the output value it is impossible (except for a brute force attack) to figure out what the input value is.
Open Shortest Path First (OSPF)	Open Shortest Path First is a link state routing algorithm used in interior gateway routing. Routers maintain a database of all routers in the autonomous system with links between the routers, link costs, and link states (up and down).
Open storage	Storage of classified information within an accredited facility, but not in General Services Administration approved secure containers, while the facility is unoccupied by authorized personnel.
Operational key	Key intended for use over-the-air for protection of operational information or for the production or secure electrical transmission of key streams.

Operational vulnerability information	Information that describes the presence of a vulnerability within a specific operational setting or network.
Operational waiver	Authority for continued use of unmodified COMSEC end-items pending the completion of a mandatory modification.
Operations code	Code composed largely of words and phrases suitable for general communications use.
Operations Security (OPSEC)	Systematic and proven process by which potential adversaries can be denied information about capabilities and intentions by identifying, controlling, and protecting generally unclassified evidence of the planning and execution of sensitive activities. The process involves five steps: identification of critical information, analysis of threats, analysis of vulnerabilities, assessment of risks, and application of appropriate countermeasures.
Optional modification	NSA-approved modification not required for universal implementation by all holders of a COMSEC end-item. This class of modification requires all of the engineering/doctrinal control of mandatory modification but is usually not related to security, safety, TEMPEST, or reliability. (See mandatory modification.)
Organizational maintenance	Limited maintenance performed by a user organization.
Organizational Registration Authority (ORA)	Entity within the PKI that authenticates the identity and the organizational affiliation of the users.
OS Detection / OS Fingerprinting	The process by which an individual can determine which operating system a remote machine is using be the behavior it exhibits.
OSI	OSI (Open Systems Interconnection) is a standard description or "reference model" for how messages should be transmitted between any two points in a telecommunication network. Its purpose is to guide product implementers so that their products will consistently work with other products. The reference model defines seven layers of functions that take place at each end of a communication. Although OSI is not always strictly adhered to in terms of keeping related functions together in a well-defined layer, many if not most products involved in telecommunication make an attempt to describe themselves in relation to the OSI model. It is also valuable as a single reference view of communication that furnishes everyone a

common ground for education and discussion.

The main idea in OSI is that the process of communication between two end points in a telecommunication network can be divided into layers, with each layer adding its own set of special, related functions. Each communicating user or program is at a computer equipped with these seven layers of function. So, in a given message between users, there will be a flow of data through each layer at one end down through the layers in that computer and, at the other end, when the message arrives, another flow of data up through the layers in the receiving computer and ultimately to the end user or program. The actual programming and hardware that furnishes these seven layers of function is usually a combination of the computer operating system, applications (such as your Web browser), TCP/IP or alternative transport and network protocols, and the software and hardware that enable you to put a signal on one of the lines attached to your computer.

OSI divides telecommunication into seven layers. The layers are in two groups. The upper four layers are used whenever a message passes from or to a user. The lower three layers (up to the network layer) are used when any message passes through the host computer or router.

Messages intended for this computer pass to the upper layers. Messages destined for some other host are not passed up to the upper layers but are forwarded to another host.

The seven layers are:

Layer 7: The application layer...This is the layer at which communication partners are identified, quality of service is identified, user authentication and privacy are considered, and any constraints on data syntax are identified. (This layer is not the application itself, although some applications may perform application layer functions.)

Layer 6: The presentation layer...This is a layer, usually part of an operating system, that converts incoming and outgoing data from one presentation format to another (for example, from a text stream into a popup window with the newly arrived text). Sometimes called the syntax

OSI layers

	<p>layer.</p> <p>Layer 5: The session layer...This layer sets up, coordinates, and terminates conversations, exchanges, and dialogs between the applications at each end. It deals with session and connection coordination.</p> <p>Layer 4: The transport layer...This layer manages the end-to-end control (for example, determining whether all packets have arrived) and error-checking. It ensures complete data transfer.</p> <p>Layer 3: The network layer...This layer handles the routing of the data (sending it in the right direction to the right destination on outgoing transmissions and receiving incoming transmissions at the packet level). The network layer does routing and forwarding.</p> <p>Layer 2: The data-link layer...This layer provides synchronization for the physical level and does bit-stuffing for strings of 1's in excess of 5. It furnishes transmission protocol knowledge and management.</p> <p>Layer 1: The physical layer...This layer conveys the bit stream through the network at the electrical and mechanical level. It provides the hardware means of sending and receiving data on a carrier.</p>
Over-the-air key Distribution	Providing electronic key via over-the-air rekeying, over-the-air key transfer, or cooperative key generation.
Over-the-air key Transfer	Electronically distributing key without changing traffic encryption key used on the secured communications path over which the transfer is accomplished.
Over-the-air Rekeying (OTAR)	Changing traffic encryption key or transmission security key in remote crypto-equipment by sending new key directly to the remote crypto-equipment over the communications path it secures.
Overload	Hindrance of system operation by placing excess burden on the performance capabilities of a system component.
Overt	Communications path within a computer system or network designed for the authorized transfer of data. (See covert channel.)

Overwrite procedure	Process of writing patterns of data on top of the data stored on a magnetic medium.
Packet	A piece of a message transmitted over a packet-switching network. One of the key features of a packet is that it contains the destination address in addition to the data. In IP networks, packets are often called datagrams.
Packet Switched Network	A packet switched network is where individual packets each follow their own paths through the network from one endpoint to another.
Parity	Bit(s) used to determine whether a block of data has been altered.
Partitioned security mode	IS security mode of operation wherein all personnel have the clearance, but not necessarily formal access approval and need-to-know, for all information handled by an IS.
Partitions	Major divisions of the total physical hard disk space.
Password	Protected/private string of letters, numbers, and special characters used to authenticate an identity or to authorize access to data.
Password Authentication Protocol (PAP)	Password Authentication Protocol is a simple, weak authentication mechanism where a user enters the password and it is then sent across the network, usually in the clear.
Password Cracking	Password cracking is the process of attempting to guess passwords, given the password file information.
Password Sniffing	Passive wiretapping, usually on a local area network, to gain knowledge of passwords.
Patch	A patch is a small update released by a software manufacturer to fix bugs in existing programs.
Patching	Patching is the process of updating software to a different version.
Payload	Payload is the actual application data a packet contains.
Penetration	Gaining unauthorized logical access to sensitive data by circumventing a system's protections. See intrusion.
Penetration Testing	Security testing in which evaluators attempt to circumvent the security features of a system based on their understanding of the system design and implementation.
Per-call key	Unique traffic encryption key generated automatically by certain secure telecommunications systems to secure single voice or data transmissions. (See cooperative key)

	generation.)
Periods Processing	Processing of various levels of classified and unclassified information at distinctly different times. Under the concept of periods processing, the system must be purged of all information from one processing period before transitioning to the next.
Perimeter	Encompasses all those components of the system that are to be accredited by the DAA, and excludes separately accredited systems to which the system is connected.
Permutation	Permutation keeps the same letters but changes the position within a text to scramble the message.
Permuter	Device used in crypto-equipment to change the order in which the contents of a shift register are used in various nonlinear combining circuits.
Personal Firewalls	Personal firewalls are those firewalls that are installed and run on individual PCs.
Ping of Death	An attack that sends an improperly large ICMP echo request packet (a "ping") with the intent of overflowing the input buffers of the destination machine and causing it to crash.
Ping Scan	A ping scan looks for machines that are responding to ICMP Echo Requests.
Ping Sweep	An attack that sends ICMP echo requests ("pings") to a range of IP addresses, with the goal of finding hosts that can be probed for vulnerabilities.
Plain text	Ordinary readable text before being encrypted into ciphertext or after being decrypted.
Point-to-Point Protocol (PPP)	A protocol for communication between two computers using a serial interface, typically a personal computer connected by phone line to a server. It packages your computer's TCP/IP packets and forwards them to the server where they can actually be put on the Internet.
Point-to-Point Tunneling Protocol (PPTP)	A protocol (set of communication rules) that allows corporations to extend their own corporate network through private "tunnels" over the public Internet.
Poison Reverse	Split horizon with poisoned reverse (more simply, poison reverse) does include such routes in updates, but sets their metrics to infinity. In effect, advertising the fact that there routes are not reachable.

Policy approving authority (PAA)	First level of the PKI Certification Management Authority that approves the security policy of each PCA.
Policy certification authority (PCA)	Second level of the PKI Certification Management Authority that formulates the security policy under which it and its subordinate CAs will issue public key certificates.
Polyinstantiation	Polyinstantiation is the ability of a database to maintain multiple records with the same key. It is used to prevent inference attacks.
Polymorphism	Polymorphism is the process by which malicious software changes its underlying code to avoid detection.
Port	A port is nothing more than an integer that uniquely identifies an endpoint of a communication stream. Only one process per machine can listen on the same port number.
Port Scan	A port scan is a series of messages sent by someone attempting to break into a computer to learn which computer network services, each associated with a "well-known" port number, the computer provides. Port scanning, a favorite approach of computer cracker, gives the assailant an idea where to probe for weaknesses. Essentially, a port scan consists of sending a message to each port, one at a time. The kind of response received indicates whether the port is used and can therefore be probed for weakness.
Positive control material	Generic term referring to a sealed authenticator system, permissive action link, coded switch system, positive enable system, or nuclear command and control documents, material, or devices.
Possession	Possession is the holding, control, and ability to use information.
Post Office Protocol, Version 3 (POP3)	An Internet Standard protocol by which a client workstation can dynamically access a mailbox on a server host to retrieve mail messages that the server has received and is holding for the client.
Practical Extraction and Reporting Language (Perl)	A script programming language that is similar in syntax to the C language and that includes a number of popular Unix facilities such as sed, awk, and tr.
Preamble	A preamble is a signal used in network communications to synchronize the transmission timing between two or more systems. Proper timing ensures that all systems are interpreting the start of the information transfer correctly. A

	preamble defines a specific series of transmission pulses that is understood by communicating systems to mean "someone is about to transmit data". This ensures that systems receiving the information correctly interpret when the data transmission starts. The actual pulses used as a preamble vary depending on the network communication technology in use.
Preproduction model	Version of INFOSEC equipment employing standard parts and suitable for complete evaluation of form, design, and performance. Preproduction models are often referred to as beta models.
Pretty Good Privacy (PGP)™	Trademark of Network Associates, Inc., referring to a computer program (and related protocols) that uses cryptography to provide data security for electronic mail and other applications on the Internet.
Principal Accrediting Authority (PAA)	Senior official with authority and responsibility for all intelligence systems within an agency.
Print suppression	Eliminating the display of characters in order to preserve their secrecy.
Privacy system	Commercial encryption system that affords telecommunications limited protection to deter a casual listener, but cannot withstand a technically competent cryptanalytic attack.
Private Addressing	IANA has set aside three address ranges for use by private or non-Internet connected networks. This is referred to as Private Address Space and is defined in RFC 1918. The reserved address blocks are: 10.0.0.0 to 10.255.255.255 (10/8 prefix) 172.16.0.0 to 172.31.255.255 (172.16/12 prefix) 192.168.0.0 to 192.168.255.255 (192.168/16 prefix)
Privileged user	Individual who has access to system control, monitoring, or administration functions (e.g., system administrator, system ISSO, maintainers, system programmers, etc.)
Probe	Type of incident involving an attempt to gather information about an IS for the apparent purpose of circumventing its security controls.
Production model	INFOSEC equipment in its final mechanical and electrical form.
Program Infector	A program infector is a piece of malware that attaches itself to existing program files.

Program Policy	A Program policy is a high-level policy that sets the overall tone of an organization's security approach.
Promiscuous Mode	When a machine reads all packets off the network, regardless of who they are addressed to. This is used by network administrators to diagnose network problems, but also by unsavory characters who are trying to eavesdrop on network traffic (which might contain passwords or other information).
Proprietary Information	Material and information relating to or associated with a company's products, business, or activities, including but not limited to financial information; data or statements; trade secrets; product research and development; existing and future product designs and performance specifications; marketing plans or techniques; schematics; client lists; computer programs; processes; and know-how that has been clearly identified and properly marked by the company as proprietary information, trade secrets, or company confidential information. The information must have been developed by the company and not be available to the Government or to the public without restriction from another source.
Protected Distribution Systems (PDS)	Wire line or fiber optic distribution system used to transmit unencrypted classified national security information through an area of lesser classification or control.
Protection philosophy	Informal description of the overall design of an IS delineating each of the protection mechanisms employed. Combination of formal and informal techniques, appropriate to the evaluation class, used to show the mechanisms are adequate to enforce the security policy.
Protection profile	Common Criteria specification that represents an implementation-independent set of security requirements for a category of Target of Evaluations (TOE) that meets specific consumer needs.
Protection ring (C.F.D.)	One of a hierarchy of privileged modes of an IS that gives certain access rights to user programs and processes that are authorized to operate in a given mode.

Protective packaging	Packaging techniques for COMSEC material that discourage penetration, reveal a penetration has occurred or was attempted, or inhibit viewing or copying of keying material prior to the time it is exposed for use.
Protective technologies	Special tamper-evident features and materials employed for the purpose of detecting tampering and deterring attempts to compromise, modify, penetrate, extract, or substitute information processing equipment and keying material.
Protocol	A formal specification for communicating; an IP address the special set of rules that end points in a telecommunication connection use when they communicate. Protocols exist at several levels in a telecommunication connection.
Protocol Stacks (OSI)	A set of network protocol layers that work together.
Proxy	Software agent that performs a function or operation on behalf of another application or system while hiding the details involved.
Proxy Server	A server that acts as an intermediary between a workstation user and the Internet so that the enterprise can ensure security, administrative control, and caching service. A proxy server is associated with or part of a gateway server that separates the enterprise network from the outside network and a firewall server that protects the enterprise network from outside intrusion.
Public domain software	Software not protected by copyright laws of any nation that may be freely used without permission of, or payment to, the creator, and that carries no warranties from, or liabilities to the creator.
Public Key	The publicly-disclosed component of a pair of cryptographic keys used for asymmetric cryptography.
Public Key Certificate	Contains the name of a user, the public key component of the user, and the name of the issuer who vouches that the public key component is bound to the named user.
Public Key Cryptography (PKC)	Encryption system using a linked pair of keys. What one key encrypts, the other key decrypts.
Public Key Encryption	The popular synonym for "asymmetric cryptography".
Public Key Infrastructure (PKI)	A PKI (public key infrastructure) enables users of a basically unsecured public network such as the Internet to securely and privately exchange data and money through the use of a public and a

	private cryptographic key pair that is obtained and shared through a trusted authority. The public key infrastructure provides for a digital certificate that can identify an individual or an organization and directory services that can store and, when necessary, revoke the certificates.
Public-Key Forward Secrecy (PFS)	For a key agreement protocol based on asymmetric cryptography, the property that ensures that a session key derived from a set of long-term public and private keys will not be compromised if one of the private keys is compromised in the future.
Purging	Rendering stored information unrecoverable. (See sanitize.)
QAZ	A network worm.
QUADRANT	Short name referring to technology that provides tamper-resistant protection to crypto-equipment.
Race Condition	A Race condition exploits the small window of time between a security control being applied and when the service is used.
Radiation Monitoring	Radiation monitoring is the process of receiving images, data, or audio from an unprotected source by listening to radiation signals.
Randomizer	Analog or digital source of unpredictable, unbiased, and usually independent bits. Randomizers can be used for several different functions, including key generation or to provide a starting state for a key generator.
Read	Fundamental operation in an IS that results only in the flow of information from an object to a subject.
Read access	Permission to read information in an IS.
Real time reaction	Immediate response to a penetration attempt that is detected and diagnosed in time to prevent access.
Reconnaissance	Reconnaissance is the phase of an attack where an attacker finds new systems, maps out networks, and probes for specific, exploitable vulnerabilities.
Recovery procedures	Actions necessary to restore data files of an IS and computational capability after a system failure.
RED	Designation applied to an IS, and associated areas, circuits, components, and equipment in which unencrypted national security information is being processed.
RED/BLACK concept	Separation of electrical and electronic circuits, components, equipment, and systems that handle

	national security information (RED), in electrical form, from those that handle non-national security information (BLACK) in the same form.
Red team	Interdisciplinary group of individuals authorized to conduct an independent and focused threat-based effort as a simulated adversary to expose and exploit system vulnerabilities for the purpose of improving the security posture of information systems.
RED signal	Any electronic emission (e.g., plain text, key, key stream, subkey stream, initial fill, or control signal) that would divulge national security information if recovered.
Reference monitor	Concept of an abstract machine that enforces Target of Evaluation (TOE) access control policies.
Reflexive ACLs (Cisco)	Reflexive ACLs for Cisco routers are a step towards making the router act like a stateful firewall. The router will make filtering decisions based on whether connections are a part of established traffic or not.
Registry	The Registry in Windows operating systems is the central set of settings and information required to run the Windows computer.
Release prefix	Prefix appended to the short title of U.S.-produced keying material to indicate its foreign releasability. "A" designates material that is releasable to specific allied nations and "U.S." designates material intended exclusively for U. S. use.
Remanence	Residual information remaining on storage media after clearing. (See magnetic remanence and clearing.)
Remote access	Access for authorized users external to an enclave established through a controlled access point at the enclave boundary.
Remote rekeying	Procedure by which a distant crypto-equipment is rekeyed electrically. (See automatic remote rekeying and manual remote rekeying.)
Repair action	NSA-approved change to a COMSEC end-item that does not affect the original characteristics of the end-item and is provided for optional application by holders. Repair actions are limited to minor electrical and/or mechanical improvements to enhance operation, maintenance, or reliability. They do not require an identification label, marking, or control but must be fully documented by changes to the maintenance manual.

Request for Comment (RFC)	A series of notes about the Internet, started in 1969 (when the Internet was the ARPANET). An Internet Document can be submitted to the IETF by anyone, but the IETF decides if the document becomes an RFC. Eventually, if it gains enough interest, it may evolve into an Internet standard.
Reserve keying material	Key held to satisfy unplanned needs. (See contingency key.)
Residual risk	Portion of risk remaining after security measures have been applied.
Residue	Data left in storage after information processing operations are complete, but before degaussing or overwriting has taken place.
Resource encapsulation	Method by which the reference monitor mediates accesses to an IS resource. Resource is protected and not directly accessible by a subject. Satisfies requirement for accurate auditing of resource usage.
Resource Exhaustion	Resource exhaustion attacks involve tying up finite resources on a system, making them unavailable to others.
Response	A response is information sent that is responding to some stimulus.
Reverse Address Resolution Protocol (RARP)	RARP (Reverse Address Resolution Protocol) is a protocol by which a physical machine in a local area network can request to learn its IP address from a gateway server's Address Resolution Protocol table or cache. A network administrator creates a table in a local area network's gateway router that maps the physical machine (or Media Access Control - MAC address) addresses to corresponding Internet Protocol addresses. When a new machine is set up, its RARP client program requests from the RARP server on the router to be sent its IP address. Assuming that an entry has been set up in the router table, the RARP server will return the IP address to the machine which can store it for future use.
Reverse Engineering	Acquiring sensitive data by disassembling and analyzing the design of a system component.
Reverse Lookup	Find out the hostname that corresponds to a particular IP address. Reverse lookup uses an IP (Internet Protocol) address to find a domain name.
Reverse Proxy	Reverse proxies take public HTTP requests and pass them to back-end web servers to send the content to it, so the proxy can then send the content to the end-user.

Risk	Risk is the product of the level of threat with the level of vulnerability. It establishes the likelihood of a successful attack.
Risk Analysis	Examination of information to identify the risk to an IS.
Risk Assessment	Process of analyzing threats to and vulnerabilities of an IS, and the potential impact resulting from the loss of information or capabilities of a system. This analysis is used as a basis for identifying appropriate and cost-effective security countermeasures.
Risk Index (R.F.D.)	Difference between the minimum clearance or authorization of IS users and the maximum sensitivity (e.g.; classification and categories) of data processed by the system.
Risk Management	Process of managing risks to agency operations (including mission, functions, image, or reputation), agency assets, or individuals resulting from the operation of an information system. It includes risk assessment; cost-benefit analysis; the selection, implementation, and assessment of security controls; and the formal authorization to operate the system. The process considers effectiveness, efficiency, and constraints due to laws, directives, policies, or regulations. (NIST Special Pub 800-53)
Rivest-Shamir-Adleman (RSA)	An algorithm for asymmetric cryptography, invented in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman.
Role-based Access Control	Role-based access control assigns users to roles based on their organizational functions and determines authorization based on those roles.
Root	Root is the name of the administrator account in UNIX systems.
Rootkit	A collection of tools (programs) that a hacker uses to mask intrusion and obtain administrator-level access to a computer or computer network.
Router	Routers interconnect logical networks by forwarding information to other networks based upon IP addresses.
Routing Information Protocol (RIP)	Routing Information Protocol is a distance vector protocol used for interior gateway routing which uses hop count as the sole metric of a path's cost.
Routing Loop	A routing loop is where two or more poorly configured routers repeatedly exchange the same packet over and over.
RPC Scans	RPC scans determine which RPC services are running on a machine.

Rule Set Based Access Control (RSBAC)	Rule Set Based Access Control targets actions based on rules for entities operating on objects.
S/Key	A security mechanism that uses a cryptographic hash function to generate a sequence of 64-bit, one-time passwords for remote user login. The client generates a one-time password by applying the MD4 cryptographic hash function multiple times to the user's secret key. For each successive authentication of the user, the number of hash applications is reduced by one.
Safeguard	1.) Protection included to counteract a known or expected condition. 2.) Incorporated countermeasure or set of countermeasures within a base release.
Safeguarding statement	Statement affixed to a computer output or printout that states the highest classification being processed at the time the product was produced and requires control of the product, at that level, until determination of the true classification by an authorized individual. Synonymous with banner.
Safety	Safety is the need to ensure that the people involved with the company, including employees, customers, and visitors, are protected from harm.
Sanitize	Process to remove information from media such that data recovery is not possible. It includes removing all classified labels, markings, and activity logs. (See purging.)
Scavenging	Searching through data residue in a system to gain unauthorized knowledge of sensitive data.
Secure communications	Telecommunications deriving security through use of type 1 products and/or PDSs.
Secure Electronic Transactions (SET)	Secure Electronic Transactions is a protocol developed for credit card transactions in which all parties (customers, merchant, and bank) are authenticated using digital signatures, encryption protects the message and provides integrity, and provides end-to-end security for credit card transactions online.
Secure hash standard	Specification for a secure hash algorithm that can generate a condensed message representation called a message digest.
Secure Shell (SSH)	A program to log into another computer over a network, to execute commands in a remote machine, and to move files from one machine to another.

Secure Sockets Layer (SSL)	A protocol developed by Netscape for transmitting private documents via the Internet. SSL works by using a public key to encrypt data that's transferred over the SSL connection.
Secure State	Condition in which no subject can access any object in an unauthorized manner.
Secure Subsystem	Subsystem containing its own implementation of the reference monitor concept for those resources it controls. Secure subsystem must depend on other controls and the base operating system for the control of subjects and the more primitive system objects.
Security controls	Management, operational, and technical controls (i.e., safeguards or countermeasures) prescribed for an information system to protect the confidentiality, integrity, and availability of the system and its information. (NIST Special Pub 800-53)
Security Fault Analysis (SFA)	Assessment, usually performed on IS hardware, to determine the security properties of a device when hardware fault is encountered.
Security Features Users Guide (SFUG) (C.F.D.)	Guide or manual explaining how the security mechanisms in a specific system work.
Security filter	IS trusted subsystem that enforces security policy on the data passing through it.
Security-in-Depth	Synonymous with defense-in-depth.
Security inspection	Examination of an IS to determine compliance with security policy, procedures, and practices.
Security kernel	Hardware, firmware, and software elements of a trusted computing base implementing the reference monitor concept. Security kernel must mediate all accesses, be protected from modification, and be verifiable as correct.
Security label	Information representing the sensitivity of a subject or object, such as UNCLASSIFIED or its hierarchical classification (CONFIDENTIAL, SECRET, TOP SECRET) together with any applicable nonhierarchical security categories (e.g., sensitive compartmented information, critical nuclear weapon design information).
Security net control station	Management system overseeing and controlling implementation of network security policy.
Security perimeter	Boundary where security controls are in effect to protect assets.
Security Policy	A set of rules and practices that specify or regulate how a system or organization provides security services to protect sensitive and critical system resources.

Security Range	Highest and lowest security levels that are permitted in or on an IS, system component, subsystem, or network.
Security requirements	Types and levels of protection necessary for equipment, data, information, applications, and facilities to meet IS security policy.
Security requirements baseline	Description of the minimum requirements necessary for an IS to maintain an acceptable level of security.
Security safeguards	Protective measures and controls prescribed to meet the security requirements specified for an IS. Safeguards may include security features, management constraints, personnel security, and security of physical structures, areas, and devices. (See accreditation.)
Security specification	Detailed description of the safeguards required to protect an IS.
Security target	Common Criteria specification that represents a set of security requirements to be used as the basis of an evaluation of an identified Target of Evaluation (TOE).
Security test and evaluation (ST&E)	Examination and analysis of the safeguards required to protect an IS, as they have been applied in an operational environment, to determine the security posture of that system.
Security testing	Process to determine that an IS protects data and maintains functionality as intended.
Seed key	Initial key used to start an updating or key generation process.
Segment	Segment is another name for TCP packets.
Sensitive Compartmented Information (SCI)	Classified information concerning or derived from intelligence sources, methods, or analytical processes, which is required to be handled within formal access control systems established by the Director of Central Intelligence.
Sensitive Compartment Information Facility (SCIF)	Accredited area, room, or group of rooms, buildings, or installation where SCI may be stored, used, discussed, and/or processed.
Sensitive Information	Information, the loss, misuse, or unauthorized access to or modification of, that could adversely affect the national interest or the conduct of federal programs, or the privacy to which individuals are entitled under 5 U.S.C. Section 552a (the Privacy Act), but that has not been specifically authorized under criteria established by an Executive Order or an Act of Congress to be kept classified in the interest of national defense or foreign policy. (Systems that are not

	national security systems, but contain sensitive information, are to be protected in accordance with the requirements of the Computer Security Act of 1987 (P.L.100-235).)
Sensitivity label	Information representing elements of the security label(s) of a subject and an object. Sensitivity labels are used by the trusted computing base (TCB) as the basis for mandatory access control decisions.
Separation of Duties	Separation of duties is the principle of splitting privileges among multiple individuals or systems.
Server	A system entity that provides a service in response to requests from other system entities called clients.
Session	A session is a virtual connection between two hosts by which network traffic is passed.
Session Hijacking	Take over a session that someone else has established.
Session Key	In the context of symmetric encryption, a key that is temporary or is used for a relatively short period of time. Usually, a session key is used for a defined period of communication between two computers, such as for the duration of a single connection or transaction set, or the key is used in an application that protects relatively large amounts of data and, therefore, needs to be rekeyed frequently.
Shadow Password Files	A system file in which encryption user password are stored so that they aren't available to people who try to break into the system.
Share	A share is a resource made public on a machine, such as a directory (file share) or printer (printer share).
Shell	A Unix term for the interactive user interface with an operating system. The shell is the layer of programming that understands and executes the commands a user enters. In some systems, the shell is called a command interpreter. A shell usually implies an interface with a command syntax (think of the DOS operating system and its "C:>" prompts and user commands such as "dir" and "edit").
Shielded enclosure	Room or container designed to attenuate electromagnetic radiation, acoustic signals, or emanations.

Short title	Identifying combination of letters and numbers assigned to certain COMSEC materials to facilitate handling, accounting, and controlling.
Signals Analysis	Gaining indirect knowledge of communicated data by monitoring and analyzing a signal that is emitted by a system and that contains the data but is not intended to communicate the data.
Signature	A Signature is a distinct pattern in network traffic that can be identified to a specific tool or exploit.
Simple Integrity Property	In Simple Integrity Property a user cannot write data to a higher integrity level than their own.
Simple Network Management Protocol (SNMP)	The protocol governing network management and the monitoring of network devices and their functions. a set of protocols for managing complex networks.
Simple Security Property	In Simple Security Property a user cannot read data of a higher classification than their own. Bell-La Padula security model rule allowing a subject read access to an object, only if the security level of the subject dominates the security level of the object.
Single point keying	Means of distributing key to multiple, local crypto- equipment or devices from a single fill point.
Smartcard	A smartcard is an electronic badge that includes a magnetic strip or chip that can record and replay a set key.
Smurf	The Smurf attack works by spoofing the target address and sending a ping to the broadcast address for a remote network, which results in a large amount of ping replies being sent to the target.
Sniffer	A sniffer is a tool that monitors network traffic as it received on a network interface.
Sniffing	A synonym for "passive wiretapping."
Social Engineering	A euphemism for non-technical or low-technology means--such as lies, impersonation, tricks, bribes, blackmail, and threats--used to attack information systems.
Socket	The socket tells a host's IP stack where to plug in a data stream so that it connects to the right application.
Socket Pair	A way to uniquely specify a connection, i.e., source IP address, source port, destination IP address, destination port.
SOCKS	A protocol that a proxy server can use to accept requests from client users in a company's

	network so that it can forward them across the Internet. SOCKS uses sockets to represent and keep track of individual connections. The client side of SOCKS is built into certain Web browsers and the server side can be added to a proxy server.
Software	Computer programs (which are stored in and executed by computer hardware) and associated data (which also is stored in the hardware) that may be dynamically written or modified during execution.
Software assurance	Level of confidence that software is free from vulnerabilities, either intentionally designed into the software or accidentally inserted at anytime during its lifecycle, and that the software functions in the intended manner.
Software system test and evaluation process	Process that plans, develops, and documents the quantitative demonstration of the fulfillment of all baseline functional performance, operational, and interface requirements.
Source Port	The port that a host uses to connect to a server. It is usually a number greater than or equal to 1024. It is randomly generated and is different each time a connection is made.
Spam	Electronic junk mail or junk newsgroup postings.
Spanning Port	Configures the switch to behave like a hub for a specific port.
Special Access Program (SAP)	Sensitive program, approved in writing by a head of agency with original top secret classification authority, that imposes need-to-know and access controls beyond those normally provided for access to Confidential, Secret, or Top Secret information. The level of controls is based on the criticality of the program and the assessed hostile intelligence threat. The program may be an acquisition program, an intelligence program, or an operations and support program. (Joint Pub 1-02, 12 Apr 2001)
Special Access Program Facility (SAPF)	Facility formally accredited by an appropriate agency in accordance with DCID 6/9 in which SAP information may be processed.
Spillage	See classified information spillage.
Split Key	A cryptographic key that is divided into two or more separate data items that individually convey no knowledge of the whole key that results from combining the items.
Split Knowledge	Separation of data or information into two or more parts, each part constantly kept under

	control of separate authorized individuals or teams so that no one individual or team will know the whole data.
Split Horizon	Split horizon is a algorithm for avoiding problems caused by including routes in updates sent to the gateway from which they were learned.
Spooft	Attempt by an unauthorized entity to gain access to a system by posing as an authorized user.
Spoofing	Unauthorized use of legitimate Identification and Authentication (I&A) data, however it was obtained, to mimic a subject different from the attacker. Impersonating, masquerading, piggybacking, and mimicking are forms of spoofing.
Spread spectrum	Telecommunications techniques in which a signal is transmitted in a bandwidth considerably greater than the frequency content of the original information. Frequency hopping, direct sequence spreading, time scrambling, and combinations of these techniques are forms of spread spectrum.
SQL Injection	SQL injection is a type of input validation attack specific to database-driven applications where SQL code is inserted into application queries to manipulate the database.
Stack Smashing	Stack smashing is the technique of using a buffer overflow to trick a computer into executing arbitrary code.
Standard ACLs (Cisco)	Standard ACLs on Cisco routers make packet filtering decisions based on Source IP address only.
Star Property	In Star Property, a user cannot write data to a lower classification level without logging in at that lower classification level.
Start-up KEK	Key-encryption-key held in common by a group of potential communicating entities and used to establish ad hoc tactical networks.
State Machine	A system that moves through a series of progressive conditions.
Stateful Inspection	Also referred to as dynamic packet filtering. Stateful inspection is a firewall architecture that works at the network layer. Unlike static packet filtering, which examines a packet based on the information in its header, stateful inspection examines not just the header information but also the contents of the packet up through the application layer in order to determine more

	about the packet than just information about its source and destination.
Static Host Tables	Static host tables are text files that contain hostname and address mapping.
Static Routing	Static routing means that routing table entries contain information that do not change.
Stealthing	Stealthing is a term that refers to approaches used by malicious code to conceal its presence on the infected system.
Steganalysis	Steganalysis is the process of detecting and defeating the use of steganography.
Steganography	Methods of hiding the existence of a message or other data. This is different than cryptography, which hides the meaning of a message but does not hide the message itself. An example of a steganographic method is "invisible" ink.
Stimulus	Stimulus is network traffic that initiates a connection or solicits a response.
Storage Object	Object supporting both read and write accesses to an IS.
Store-and-Forward	Store-and-Forward is a method of switching where the entire packet is read by a switch to determine if it is intact before forwarding it.
Straight-Through Cable	A straight-through cable is where the pins on one side of the connector are wired to the same pins on the other end. It is used for interconnecting nodes on the network.
Stream Cipher	A stream cipher works by encryption a message a single bit, byte, or computer word at a time.
Strong authentication	Layered authentication approach relying on two or more authenticators to establish the identity of an originator or receiver of information.
Strong Star Property	In Strong Star Property, a user cannot write data to higher or lower classifications levels than their own.
Subassembly	Major subdivision of an assembly consisting of a package of parts, elements, and circuits that perform a specific function.
Subject	Generally an individual, process, or device causing information to flow among objects or change to the system state.
Subject security level	Sensitivity label(s) of the objects to which the subject has both read and write access. Security level of a subject must always be dominated by the clearance level of the user associated with the subject.

Sub Network	A separately identifiable part of a larger network that typically represents a certain limited number of host computers, the hosts in a building or geographic area, or the hosts on an individual local area network.
Subnet Mask	A subnet mask (or number) is used to determine the number of bits used for the subnet and host portions of the address. The mask is a 32-bit value that uses one-bits for the network and subnet portions and zero-bits for the host portion.
Superencryption	Process of encrypting encrypted information. Occurs when a message, encrypted off-line, is transmitted over a secured, on-line circuit, or when information encrypted by the originator is multiplexed onto a communications trunk, which is then bulk encrypted.
Supersession	Scheduled or unscheduled replacement of a COMSEC aid with a different edition.
Supervisor state (C.F.D.)	Synonymous with executive state of an operating system.
Suppression measure	Action, procedure, modification, or device that reduces the level of, or inhibits the generation of, compromising emanations in an IS.
Surrogate access	See discretionary access control.
Switch	A switch is a networking device that keeps track of MAC addresses attached to each of its ports so that data is only transmitted on the ports that are the intended recipient of the data.
Switched Network	A communications network, such as the public switched telephone network, in which any user may be connected to any other user through the use of message, circuit, or packet switching and control devices. 2. Any network providing switched communications service.
Syllabary (C.F.D.)	List of individual letters, combination of letters, or syllables, with their equivalent code groups, used for spelling out words or proper names not present in the vocabulary of a code. A syllabary may also be a spelling table.
Symbolic Links	Special files which point at another file.
Symmetric Cryptography	A branch of cryptography involving algorithms that use the same key for two different steps of the algorithm (such as encryption and decryption, or signature creation and signature verification). Symmetric cryptography is sometimes called "secret-key cryptography" (versus public-key cryptography) because the entities that share the key.

Symmetric Key	Encryption methodology in which the encryptor and decryptor use the same key, which must be kept secret.
SYN Flood	A denial of service attack that sends a host more TCP SYN packets (request to synchronize sequence numbers, used when opening a connection) than the protocol implementation can handle.
Synchronization	Synchronization is the signal made up of a distinctive pattern of bits that network hardware looks for to signal the start of a frame.
Synchronous Crypto-operation	Method of on-line crypto-operation in which crypto-equipment and associated terminals have timing systems to keep them in step.
Syslog	Syslog is the system logging facility for Unix systems.
System Administrator (SA)	Individual responsible for the installation and maintenance of an IS, providing effective IS utilization, adequate security parameters, and sound implementation of established IA policy and procedures.
System assets	Any software, hardware, data, administrative, physical, communications, or personnel resource within an IS.
System development methodologies	Methodologies developed through software engineering to manage the complexity of system development. Development methodologies include software engineering aids and high-level design analysis tools.
System high (C.F.D.)	Highest security level supported by an IS.
System high mode	IS security mode of operation wherein each user, with direct or indirect access to the IS, its peripherals, remote terminals, or remote hosts, has all of the following: a. valid security clearance for all information within an IS; b. formal access approval and signed nondisclosure agreements for all the information stored and/or processed (including all compartments, subcompartments and/or special access programs); and c. valid need-to-know for some of the information contained within the IS.
System indicator	Symbol or group of symbols in an off-line encrypted message identifying the specific cryptosystem or key used in the encryption.
System integrity	Attribute of an IS when it performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system.

System low (C.F.D.)	Lowest security level supported by an IS.
System profile	Detailed security description of the physical structure, equipment component, location, relationships, and general operating environment of an IS.
System security	See information systems security.
System Security Engineer	See information systems security engineering.
System Security Officer (SSO)	A person responsible for enforcement or administration of the security policy that applies to the system.
System Security Plan	Document fully describing the planned security tasks and controls required to meet system security requirements.
System-Specific Policy	A System-specific policy is a policy written for a specific system or device.
T1, T3	A digital circuit using TDM (Time-Division Multiplexing).
Tamper	To deliberately alter a system's logic, data, or control information to cause the system to perform unauthorized functions or services.
Target of evaluation (TOE)	IT product or system and its associated administrator and user guidance documentation that is the subject of an evaluation.
TCP Fingerprinting	TCP fingerprinting is the use of odd packer header combinations to determine a remote operating system.
TCP Full Open scan	TCP Full Open scans check each port by performing a full three-way handshake on each port to determine if it was open.
TCP Half Open scan	TCP Half Open scans work by performing the first half of a three-way handshake to determine if a port is open.
TCP Wrapper	A software package which can be used to restrict access to certain network services based on the source of the connection; a simple tool to monitor and control incoming network traffic.
TCP/IP	A synonym for "Internet Protocol Suite;" in which the Transmission Control Protocol and the Internet Protocol are important parts. TCP/IP is the basic communication language or protocol of the Internet. It can also be used as a communications protocol in a private network (either an intranet or an extranet).
TCPDump	TCPDump is a freeware protocol analyzer for Unix that can monitor network traffic on a wire.
Technical controls	Security controls (i.e., safeguards or countermeasures) for an information system that

	are primarily implemented and executed by the information system through mechanisms contained in the hardware, software, or firmware components of the system. (NIST Special Pub 800-53.)
Technical vulnerability information	Detailed description of a vulnerability to include the implementable steps (such as code) necessary to exploit that vulnerability.
Telecommunications	Preparation, transmission, communication, or related processing of information (writing, images, sounds, or other data) by electrical, electromagnetic, electromechanical, electro-optical, or electronic means.
TELNET	A TCP-based, application-layer, Internet Standard protocol for remote login from one host to another.
TEMPEST	Short name referring to investigation, study, and control of compromising emanations from IS equipment.
TEMPEST Test	Laboratory or on-site test to determine the nature of compromising emanations associated with an IS.
TEMPEST Zone	Designated area within a facility where equipment with appropriate TEMPEST characteristics (TEMPEST zone assignment) may be operated.
Test key	Key intended for testing of COMSEC equipment or systems.
Threat	A potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm.
Threat Analysis	Examination of information to identify the elements comprising a threat.
Threat Assessment	A threat assessment is the identification of types of threats that an organization might be exposed to.
Threat Model	A threat model is used to describe a given threat and the harm it could do to a system if it has a vulnerability.
Threat Monitoring	Analysis, assessment, and review of audit trails and other information collected for the purpose of searching out system events that may constitute violations of system security.
Threat Vector	The method a threat uses to get to the target.
Ticket-oriented (C.F.D.)	IS protection system in which each subject maintains a list of unforgeable bit patterns called tickets, one for each object a subject is

	authorized to access. (See list-oriented.)
Time bomb	Resident computer program that triggers an unauthorized act at a predefined time.
Time-compliance date	Date by which a mandatory modification to a COMSEC end-item must be incorporated if the item is to remain approved for operational use.
Time-dependent password	Password that is valid only at a certain time of day or during a specified interval of time.
Time to Live	A value in an Internet Protocol packet that tells a network router whether or not the packet has been in the network too long and should be discarded.
Tiny Fragment Attack	With many IP implementations it is possible to impose an unusually small fragment size on outgoing packets. If the fragment size is made small enough to force some of a TCP packet's TCP header fields into the second fragment, filter rules that specify patterns for those fields will not match. If the filtering implementation does not enforce a minimum fragment size, a disallowed packet might be passed because it didn't hit a match in the filter. STD 5, RFC 791 states: Every Internet module must be able to forward a datagram of 68 octets without further fragmentation. This is because an Internet header may be up to 60 octets, and the minimum fragment is 8 octets.
TOE Security Functions (TSF)	Set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the TSP.
TOE Security Policy (TSP)	Set of rules that regulate how assets are managed, protected, and distributed within the TOE.
Token-based Access Control	Token-based access control associates a list of objects and their privileges with each user. (The opposite of list based)
Token-based Devices	A Token-based device is triggered by the time of day, so every minute the password changes, requiring the user to have the token with them when they log in.
Token Ring	A token ring network is a local area network in which all computers are connected in a ring or star topology and a binary digit or token-passing scheme is used in order to prevent the collision of data between two computers that want to send messages at the same time.

Topology	The geometric arrangement of a computer system. Common topologies include a bus, star, and ring. The specific physical, i.e., real, or logical, i.e., virtual, arrangement of the elements of a network. Note 1: Two networks have the same topology if the connection configuration is the same, although the networks may differ in physical interconnections, distances between nodes, transmission rates, and/or signal types. Note 2: The common types of network topology are illustrated.
Traceroute (tracert.exe)	Traceroute is a tool that maps the route a packet takes from the local machine to a remote destination.
Traditional INFOSEC program	Program in which NSA acts as the central procurement agency for the development and, in some cases, the production of INFOSEC items. This includes the Authorized Vendor Program. Modifications to the INFOSEC end-items used in products developed and/or produced under these programs must be approved by NSA.
Traffic analysis (TA)	Study of communications patterns
Traffic encryption key (TEK)	Key used to encrypt plain text or to superencrypt previously encrypted text and/or to decrypt cipher text.
Traffic-flow security (TFS)	Measure used to conceal the presence of valid messages in an on-line cryptosystem or secure communications system.
Traffic padding	Generation of spurious communications or data units to disguise the amount of real data units being sent.
Tranquility	Property whereby the security level of an object cannot change while the object is being processed by an IS.
Transmission Control Protocol (TCP)	A set of rules (protocol) used along with the Internet Protocol to send data in the form of message units between computers over the Internet. While IP takes care of handling the actual delivery of the data, TCP takes care of keeping track of the individual units of data (called packets) that a message is divided into for efficient routing through the Internet. Whereas the IP protocol deals only with packets, TCP enables two hosts to establish a connection and exchange streams of data. TCP guarantees delivery of data and also guarantees that packets will be delivered in the same order in which they were sent.

Transmission security (TRANSEC)	Component of COMSEC resulting from the application of measures designed to protect transmissions from interception and exploitation by means other than cryptanalysis.
Transport Layer Security (TLS)	A protocol that ensures privacy between communicating applications and their users on the Internet. When a server and client communicate, TLS ensures that no third party may eavesdrop or tamper with any message. TLS is the successor to the Secure Sockets Layer.
Trap door	Synonymous with back door.
Triple DES	A block cipher, based on DES, that transforms each 64-bit plaintext block by applying the Data Encryption Algorithm three successive times, using either two or three different keys, for an effective key length of 112 or 168 bits.
Triple-wrapped	S/MIME usage: data that has been signed with a digital signature, and then encrypted, and then signed again.
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 program.
Trunking	Trunking is connecting switched together so that they can share VLAN information between them.
Trust	Trust determine which permissions and what actions other systems or users can perform on remote machines.
Trusted channel	Means by which a TOE Security Function (TSF) and a remote trusted IT product can communicate with necessary confidence to support the TOE Security Policy (TSP).
Trusted computer system	IS employing sufficient hardware and software assurance measures to allow simultaneous processing of a range of classified or sensitive information.
Trusted Computing Base (TCB)	Totality of protection mechanisms within a computer system, including hardware, firmware, and software, the combination responsible for enforcing a security policy.
Trusted distribution	Method for distributing trusted computing base (TCB) hardware, software, and firmware components that protects the TCB from modification during distribution.

Trusted foundry	Facility where both classified and unclassified parts can be produced with an extra level of assurance that the parts have not been tampered.
Trusted identification forwarding	Identification method used in IS networks whereby the sending host can verify an authorized user on its system is attempting a connection to another host. The sending host transmits the required user authentication information to the receiving host.
Trusted path	Means by which a user and a TOE Security Function (TSF) can communicate with necessary confidence to support the TOE Security Policy (TSP).
Trusted process	Process that has privileges to circumvent the system security policy and has been tested and verified to operate only as intended.
Trusted recovery	Ability to ensure recovery without compromise after a system failure.
Trusted software	Software portion of a trusted computing base (TCB).
Trusted Ports	Trusted ports are ports below number 1024 usually allowed to be opened by the root user.
TSEC nomenclature	System for identifying the type and purpose of certain items of COMSEC material.
Tunnel	A communication channel created in a computer network by encapsulating a communication protocol's data packets in (on top of) a second protocol that normally would be carried above, or at the same layer as, the first one. Most often, a tunnel is a logical point-to-point link -- i.e., an OSI layer 2 connection -- created by encapsulating the layer 2 protocol in a transport protocol (such as TCP), in a network or inter-network layer protocol (such as IP), or in another link layer protocol. Tunneling can move data between computers that use a protocol not supported by the network connecting them.
Tunneling	Technology enabling one network to send its data via another network's connections. Tunneling works by encapsulating a network protocol within packets carried by the second network.
Two-part code	Code consisting of an encoding section, in which the vocabulary items (with their associated code groups) are arranged in alphabetical or other systematic order, and a decoding section, in which the code groups (with their associated meanings) are arranged in a separate alphabetical or numeric order.

Two-person Control	Continuous surveillance and control of positive control material at all times by a minimum of two authorized individuals, each capable of detecting incorrect and unauthorized procedures with respect to the task being performed, and each familiar with established security and safety requirements.
Two-person Integrity (TPI)	System of storage and handling designed to prohibit individual access to certain COMSEC keying material by requiring the presence of at least two authorized individuals, each capable of detecting incorrect or unauthorized security procedures with respect to the task being performed. (See no-lone zone.)
Type certification	The certification acceptance of replica information systems based on the comprehensive evaluation of the technical and non-technical security features of an IS and other safeguards, made as part of and in support of the accreditation process, to establish the extent to which a particular design and implementation meet a specified set of security requirements.
Type 1 key	Generated and distributed under the auspices of NSA for use in a cryptographic device for the protection of classified and sensitive national security information.
Type 1 product	Cryptographic equipment, assembly or component classified or certified by NSA for encrypting and decrypting classified and sensitive national security information when appropriately keyed. Developed using established NSA business processes and containing NSA approved algorithms. Used to protect systems requiring the most stringent protection mechanisms.
Type 2 key	Generated and distributed under the auspices of NSA for use in a cryptographic device for the protection of unclassified national security information.
Type 2 product	Cryptographic equipment, assembly, or component certified by NSA for encrypting or decrypting sensitive national security information when appropriately keyed. Developed using established NSA business processes and containing NSA approved algorithms. Used to protect systems requiring protection mechanisms exceeding best commercial practices including systems used for the protection of unclassified national security information.

Type 3 key	Used in a cryptographic device for the protection of unclassified sensitive information, even if used in a Type 1 or Type 2 product.
Type 3 product	Unclassified cryptographic equipment, assembly, or component used, when appropriately keyed, for encrypting or decrypting unclassified sensitive U.S. Government or commercial information, and to protect systems requiring protection mechanisms consistent with standard commercial practices. Developed using established commercial standards and containing NIST approved cryptographic algorithms/modules or successfully evaluated by the National Information Assurance Partnership (NIAP).
Type 4 key	Used by a cryptographic device in support of its Type 4 functionality; i.e., any provision of key that lacks U.S. Government endorsement or oversight.
Type 4 product	Unevaluated commercial cryptographic equipment, assemblies, or components that neither NSA nor NIST certify for any Government usage. These products are typically delivered as part of commercial offerings and are commensurate with the vendor's commercial practices. These products may contain either vendor proprietary algorithms, algorithms registered by NIST, or algorithms registered by NIST and published in a FIPS.
UDP Scan	UDP scans perform scans to determine which UDP ports are open.
Unauthorized disclosure	Type of event involving exposure of information to individuals not authorized to receive it.
Unclassified	Information that has not been determined pursuant to E.O. 12958 or any predecessor order to require protection against unauthorized disclosure and that is not designated as classified.
Unicast	Broadcasting from host to host.
Uniform Resource Identifier (URI)	The generic term for all types of names and addresses that refer to objects on the World Wide Web.
Uniform Resource Locator (URL)	The global address of documents and other resources on the World Wide Web. The first part of the address indicates what protocol to use, and the second part specifies the IP address or the domain name where the resource is located. For example, http://www.pcwebopedia.com/index.html .

Unix	A popular multi-user, multitasking operating system developed at Bell Labs in the early 1970s. Created by just a handful of programmers, Unix was designed to be a small, flexible system used exclusively by programmers.
Unprotected Share	In Windows terminology, a "share" is a mechanism that allows a user to connect to file systems and printers on other systems. An "unprotected share" is one that allows anyone to connect to it.
Untrusted process	Process that has not been evaluated or examined for adherence to the security policy. It may include incorrect or malicious code that attempts to circumvent the security mechanisms.
Updating	Automatic or manual cryptographic process that irreversibly modifies the state of a COMSEC key, equipment, device, or system.
User	A person, organization entity, or automated process that accesses a system, whether authorized to do so or not. (PKI) Individual defined, registered, and bound to a public key structure by a certification authority (CA).
User ID	Unique symbol or character string used by an IS to identify a specific user.
User Contingency Plan	User contingency plan is the alternative methods of continuing business operations if IT systems are unavailable.
User Datagram Protocol (UDP)	A communications protocol that, like TCP, runs on top of IP networks. Unlike TCP/IP, UDP/IP provides very few error recovery services, offering instead a direct way to send and receive datagrams over an IP network. It's used primarily for broadcasting messages over a network. UDP uses the Internet Protocol to get a datagram from one computer to another but does not divide a message into packets (datagrams) and reassemble it at the other end. Specifically, UDP doesn't provide sequencing of the packets that the data arrives in.
User Partnership Program (UPP)	Partnership between the NSA and a U.S. Government agency to facilitate development of secure IS equipment incorporating NSA-approved cryptography. The result of this program is the authorization of the product or system to safeguard national security information in the user's specific application.

User representative	Individual authorized by an organization to order COMSEC keying material and interface with the keying system, provide information to key users, and ensure the correct type of key is ordered.
U.S.-controlled facility	Base or building to which access is physically controlled by U.S. individuals who are authorized U.S. Government or U.S. Government contractor employees.
U.S.-controlled space	Room or floor within a facility that is not a U.S.-controlled facility, access to which is physically controlled by U.S. individuals who are authorized U.S. Government or U.S. Government contractor employees. Keys or combinations to locks controlling entrance to U.S.-controlled spaces must be under the exclusive control of U.S. individuals who are U.S. Government or U.S. Government contractor employees.
U.S. person	U.S. citizen or a permanent resident alien, an unincorporated association substantially composed of U.S. citizens or permanent resident aliens, or a corporation incorporated in U.S., except for a corporation directed and controlled by a foreign government or governments.
Validated products list	List of validated products that have been successfully evaluated under the National Information Assurance Partnership (NIAP) Common Criteria Evaluation and Validation Scheme (CCEVS).
Validation	Process of applying specialized security test and evaluation procedures, tools, and equipment needed to establish acceptance for joint usage of an IS by one or more departments or agencies and their contractors.
Variant	One of two or more code symbols having the same plain text equivalent.
Verification	Process of comparing two levels of an IS specification for proper correspondence (e.g., security policy model with top-level specification, top-level specification with source code, or source code with object code).
Virtual Private Network (VPN)	A restricted-use, logical (i.e., artificial or simulated) computer network that is constructed from the system resources of a relatively public, physical (i.e., real) network (such as the Internet), often by using encryption (located at hosts or gateways), and often by tunneling links of the virtual network across the real network. For example, if a corporation has LANs at

	several different sites, each connected to the Internet by a firewall; the corporation could create a VPN by (a) using encrypted tunnels to connect from firewall to firewall across the Internet and (b) not allowing any other traffic through the firewalls. A VPN is generally less expensive to build and operate than a dedicated real network, because the virtual network shares the cost of system resources with other users of the real network.
Virus	A hidden, self-replicating section of computer software, usually malicious logic, that propagates by infecting -- i.e., inserting a copy of itself into and becoming part of - another program, leaving no sign of its presence. A virus cannot run by itself; it requires that its host program be run to make the virus active.
Vulnerability	A flaw or weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's security policy.
Vulnerability analysis	Examination of information to identify the elements comprising a vulnerability.
Vulnerability assessment	Formal description and evaluation of vulnerabilities of an IS.
WAP gap	Before WAP 2.0 wireless communications used different encryption than what was used on the Internet. The WAP gateway provided the decryption and re-encryption for the traffic, but it remained possible to capture the plaintext information if the gateway were compromised.
War chalking	War chalking is marking areas, usually on sidewalks with chalk, that receive wireless signals that can be accessed.
War Dialer	A computer program that automatically dials a series of telephone numbers to find lines connected to computer systems, and catalogs those numbers so that a cracker can try to break into the systems.
War Dialing	War dialing is a simple means of trying to identify modems in a telephone exchange that may be susceptible to compromise in an attempt to circumvent perimeter security.
War Driving	War driving is the process of traveling around looking for wireless access point signals that can be used to get network access.
Web of Trust	A web of trust is the trust that naturally evolves as a user starts to trust other's signatures, and the signatures that they trust.

Web risk assessment	Process for ensuring websites are in compliance with applicable policies.
Web Server	A software process that runs on a host computer connected to the Internet to respond to HTTP requests for documents from client web browsers.
WHOIS	An IP for finding information about resources on networks.
Windowing	A windowing system is a system for sharing a computer's graphical display presentation resources among multiple applications at the same time. In a computer that has a graphical user interface (GUI), you may want to use a number of applications at the same time (this is called task). Using a separate window for each application, you can interact with each application and go from one application to another without having to reinitiate it. Having different information or activities in multiple windows may also make it easier for you to do your work. A windowing system uses a window manager to keep track of where each window is located on the display screen and its size and status. A windowing system doesn't just manage the windows but also other forms of graphical user interface entities.
Windump	Windump is a freeware tool for Windows that is a protocol analyzer that can monitor network traffic on a wire.
Wired Equivalent Privacy	A security protocol for wireless local area networks defined in the standard IEEE 802.11b.
Wireless Application Protocol	A specification for a set of communication protocols to standardize the way that wireless devices, such as cellular telephones and radio transceivers, can be used for Internet access, including e-mail, the World Wide Web, newsgroups, and Internet Relay Chat.
Wireless technology	Permits the active or passive transfer of information between separated points without physical connection. Active information transfer may entail a transmit and/or receive emanation of energy, whereas passive information transfer entails a receive-only capability. Currently wireless technologies use IR, acoustic, RF, and optical but, as technology evolves, wireless could include other methods of transmission.
Wiretapping	Monitoring and recording data that is flowing between two points in a communication system.

Work factor	Estimate of the effort or time needed by a potential perpetrator, with specified expertise and resources, to overcome a protective measure.
World Wide Web ("the Web", WWW, W3)	The global, hypermedia-based collection of information and services that is available on Internet servers and is accessed by browsers using Hypertext Transfer Protocol and other information retrieval mechanisms.
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. See malicious code.
Wrap	To use cryptography to provide data confidentiality service for a data object.
Write	Fundamental operation in an IS that results only in the flow of information from a subject to an object. (See access type.)
Write access	Permission to write to an object in an IS.
Zero fill	To fill unused storage locations in an IS with the representation of the character denoting "0."
Zeroize	To remove or eliminate the key from a cryptoequipment or fill device.
Zone of control	Synonymous with inspectable space.
Zone transfer	A zone transfer is when a DNS server performs a complete dump of the database for a domain and sends the information from the primary DNS server to the secondary DNS servers.

References:

- <http://whatis.techtarget.com/>
- <http://www.google.com/>
- <http://www.pcwebopedia.com/>
- <http://www.its.blrdoc.gov/projects/telecomglossary2000>
- <http://www.freesoft.org/CIE/RFC/bynum.cgi?2828>
- <http://www.clock.org/~jss/glossary/index.html>
- <http://deer-run.com>, Hal Pomeranz
- **RFC 1858**
- **RFC 2080**

Acronym List

© 2013 SANS Institute

Acronym List

Acronym	Definition
3DES	Triple DES (NIST)
3G	Third Generation (telephony)
ABM	Asynchronous Balanced Mode
ACE	Access Control Entry
ACK	Acknowledgement Field Valid flag (TCP) or Acknowledgement number
ACL	Access control list
ACM	Association for Computing Machinery
AD	Active Directory (Microsoft)
ADCE	Active Directory Client Extensions
ADSL	Asymmetrical Digital Subscriber Line
AES	Advanced Encryption Standard (NIST)
AFS	Andrew File System
AH	Authentication Header (IPsec)
AIG (C.F.D.)	Address Indicator Group
AIN	Advance Intelligence Network
AK	Advanced Remote Rekeying
AKD/RCU	Automatic Key Distribution/Rekeying Control Unit
ALC	Accounting Legend Code
ALE	Annualized Loss Expectancy

ALT	ALTerate
AMAP	Application Mapping tool
AMD	AutoMounteD (Unix)
AMEX	American Express
AMS	1. Auto-Manual System 2. Autonomous Message Switch
ANDVT	Advanced Narrowband Digital Voice Terminal
ANSI	American National Standards Institute
AP	Access Point (WLAN)
APC	Adaptive Predictive Coding
APOP	Authenticated Post Office Protocol
APU	Auxiliary Power Unit
ARIN	American Registry for Internet Numbers
ARM	Asynchronous Response Mode
ARO	Annualized Rate of Occurrence
ARP	Address Resolution Protocol
ARPANET	Advanced Research Projects Agency Network
AS	Authentication Server
ASAP	As Soon As Possible
ASCII	American Standard Code for Information Interchange
ASIC	Application Specific Integrated Circuit
ASP	Active Server Page (Microsoft)
ASR	Automatic System Recovery
ASSIST Program	Automated Information System Security Incident Suport Team Program
AT	Administration Tools
ATM	Asynchronous Transfer Mode or Automatic Teller Machine
AUTODIN	Automatic Digital Network
AV	Anti-Virus
AVP	Authorized Vendor Program
AXFR	Zone Transfer
b	Bit
B	Byte (8 bits)
BCP	Business Continuity Plan
BDC	Backup Domain Controller (Microsoft Windows NT)
BER	Bit Error Rate
BGP	Border Gateway Protocol
BIA	Business Impact Analysis
BID	BlackICE Defender
BIND	Berkeley Internet Name Daemon

BIOS	Basic Input/Output System (Microsoft)
BITS	Background Intelligent Transfer Service
BMP	Bitmap File Format (Microsoft)
BO	Back Orifice
BOCA	Building Officials and Code Administrators International, Inc (Building Codes)
BOF	Back Officer Friendly
BOOTP	Bootstrap Protocol
Bps, b/s	Bits per second
BS	British Standard
BSD	Berkeley Software Distribution
BSI	British Standard Institute
BSS	Basic Service Set
C2	Command and Control
C3	Command, Control, and Communications
C3I	Command, Control, Communications, and Intelligence
C4	Command, Control, Communications, and Computers
CA	<ol style="list-style-type: none"> 1. Certificate Authority (PKI) 2. Controlling Authority 3. Cryptanalysis 4. COMSEC Account 5. Command Authority 6. Certification Authority
C&A	Certification and Accreditation
CACM	Communications of the ACM
CAST	Carlisle Adams, Stafford Tavares
CAT	Category
CAW	Certificate Authority Workstation
CBC	Cipher Block Chaining mode
CC	Common Criteria
CCEP	Commercial COMSEC Evaluation Program
CCEVS	Common Criteria Evaluation and Validation Scheme
CCI	Controlled Cryptographic Item
CCITT	Consultative Committee for International Telegraphy and Telephony
CCO	Circuit Control Officer
CCTV	Closed Circuit Television
CD	Compact Disk
CDE	Common Desktop Environment
CDFS	Compact Disk File System
CDMA	Code Division Multiple Access
CDPD	Cellular Digital Packet Data
CDROM	Compact Disk Read-Only Memory

CDS	Cross Domain Solution
CEO	Chief Executive Officer
CEOI	Communications Electronics Operating Instruction
CEPR	Compromising Emanation Performance Requirement
CER	<ol style="list-style-type: none"> 1. Crossover Error Rate 2. Cryptographic Equipment Room 3. Communication Equipment Room
CERN	A French acronym for the European Laboratory for Particle Physics
CERT	Computer Security Emergency Response Team
CFB	Cipher FeedBack mode
CFD	Common Fill Device
CGI	Common Gateway Interface
CHAP	Challenge-Handshake Authentication Protocol
CHARGEN	Character Generation Service
CIA	Confidentiality, Integrity, and Availability
CIAC	Computer Incident Assessment Capability
CID	Consensus Intrusion Database
CIDF	Common Intrusion Detection Framework
CIDR	Classless Interdomain Routing
CIFS	Common Internet File System
CIK	Crypto-Ignition Key
CIO	Chief Information Officer
CIRT	Computer Incident Response Team
CIS	Cerberus Information Security
CKG	Cooperative Key Generation
CLR	Common Language Runtime (Microsoft)
CMCS	COMSEC Material Control System
CN	Common Name
CNA	Computer Network Attack
CND	Computer Network Defense
CNN	Cable News Network
CNSS	Committee on National Security Systems
CNSSAM	Committee on National Security Systems Advisory Memorandum
CNSSD	Committee on National Security Systems Directive
CNSSI	Committee on National Security Systems Instruction
CNSSP	Committee on National Security Systems Policy
CO	Central Office
COM	Component Object Model (Microsoft)
COMEX	Commodity Exchange

COMPUSSEC	Computer Security
COMSEC	Communications Security
CONOP	Concept of Operations
COOP	Continuity of Operations Plan
COPS	Community Oriented Policing Services
COR	<ul style="list-style-type: none"> 1. Central Office of Record (COMSEC) 2. Contracting Officer Representative
COTS	Commercial-off-the-shelf
CPU	Central Processing Unit
CRC	Cyclical Redundancy Check
CRII	Code Red II Worm
CRL	Certificate Revocation List (PKI)
CRYPT	UNIX Password Algorithm
Crypt/Crypto	Cryptographic-related
CS	Code Segment
CSE	Communications Security Establishment (Canada)
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
CSO	Chief Security Officer
CSS	<ul style="list-style-type: none"> 1. COMSEC Subordinate Switch 2. Constant Surveillance Service (Courier) 3. Continuous Signature Service (Courier) 4. Coded Switch System
CSSO	Contractor Special Security Officer
CSTVRP	Computer Security Technical Vulnerability Report Program
CTAK	Cipher Text Auto-Key
CT&E	Certification Test and Evaluation
Ctrl	Control
CTTA	Certified TEMPEST Technical Authority
CUP	COMSEC Utility Program
CVE	Common Vulnerabilities and Exposures
CWR	Congestion Window Reduced
DAA	<ul style="list-style-type: none"> 1. Designated Accrediting Authority 2. Delegated Accrediting Authority 3. Designated Approval Authority
DAC	Discretionary Access Control
DACL	Discretionary Access Control List
DAD	Destruction, Alteration, and Disclosure
DAMA	Demand Assigned Multiple Access

DARPA	Defense Advanced Research Projects Agency (US)
DBS	DOS Boot Sector
dc	Domain Components
DC	Domain Controller (Microsoft)
DCE	Data Communications Equipment
DCID	Director Central Intelligence Directive
DCS	<ul style="list-style-type: none"> 1. Defense Communications System 2. Defense Courier Service
DCT	Discrete Cosine Transform
DDE	Dynamic Data Exchange
DDS	Dual Driver Service (courier)
DDoS	Distributed Denial of Service
DEA	Data Encryption Algorithm
DEC	Digital Equipment Corp. (now Compaq)
DeCSS	De-Contents Scrambling System
DEFCON	DEFense CONdition
DEL	DELete
DES	Data Encryption Standard (NIST)
DESTPORT	DESTination PORT
DF	Don't Fragment flag (IP)
DH	Diffie-Hellman
DHCP	Dynamic Host Configuration Protocol
DISN	Defense Information System Network
DITSCAP	DoD Information Technology Security Certification and Accreditation Process
DLCI	Data Link Connection Identifier
DLL	Dynamic Linked Library
DMA	Direct Memory Access
DMS	Defense Message System
DMZ	Demilitarized Zone
DN	Distinguished Name (PKI)
DNS	Domain Name System or Service
DNSSEC	Domain Name System Security
DOCSIS	Data Over Cable Interface Specification
DOJ	Department of Justice (US)
DoS	Denial of Service
DOS	Disk Operating System (PC)
DRP	Disaster Recovery Plan
DSA	<ul style="list-style-type: none"> 1. Data Signature Algorithm 2. Digital Signature Algorithm

DSDM	Dynamic Systems Development Method
DSL	Digital Subscriber Line
DSN	Defense Switched Network
DSS	Digital Signature Standard (NIST)
DSSS	Direct Sequence Spread Spectrum
DSVT	Digital Subscriber Voice Terminal
DTE	Data Terminal Equipment
DTK	Deception Toolkit (Cohen)
DTLS	Descriptive Top-Level Specification
DTD	Data Transfer Device
DTS	Diplomatic Telecommunications Service
DUA	Directory User Agent
DVD	Digital Versatile Disc
EAM	Emergency Action Message
EAS	Emergency Alert System
EAP	Extensible Authentication Protocol
EBCDIC	Extended Binary Coded Decimal Interchange Code (IBM)
ECB	Electronic Code Book mode
ECC	Elliptic Curve Cryptography
ECCM	Electronic Counter-Countermeasures
ECDLP	Elliptic Curve Discrete Logarithm Problem
ECDSA	Elliptic Curve Digital Signature Algorithm
ECE	ECN Echo
ECM	Electronic Countermeasures
ECN	Explicit Congestion Notification
ECPA	Electronic Communications Privacy Act
ECPL	Endorsed Cryptographic Products List (a section in the Information Systems Security Products and Services Catalogue)
EDAC	Error Direction and Correction
EDGAR	Education Department General Administrative Regulations
EER	Equal Error Rate
EF	Exposure Factor
EFD	Electronic Fill Device
EFS	Encrypting File System
EFTO (C.F.D.)	Encrypt for Transmission Only
EGP	Exterior Gateway Protocol
EGS	European Global System (wireless)
EIA	Electronic Industries Alliance (was Electronic Industries Association)
EICAR	European Institute for Computer Anti-Virus Research
EIGRP	Enhanced Interior Gateway Routing Protocol (Cisco)
EKMS	Electronic Key Management System

ELINT	Electronic Intelligence
E Model	Engineering Development Model
EMS	Enterprise Management System
EMSEC	Emission Security
EPL	Evaluated Products List (a section in the INFOSEC Products and Services Catalogue)
ERD	Emergency Repair Disk (Microsoft)
ERTZ	Equipment Radiation TEMPEST Zone
ESP	Encapsulating Security Payload (IPsec)
ESS	Extended Service Set
ETPL	Endorsed TEMPEST Products List
EU	European Union
EV	Event Viewer (Microsoft Windows NT/2000)
EVT	Event Viewer File Format (Microsoft)
FAA	Federal Aviation Administration (US)
FAQ	Frequently Asked Questions
FAR	False Accept Rate
FAT	File Allocation Table (Microsoft)
FBR	Floppy Boot Record
FC	File Compare Command (DOS)
FCS	Frame Check Sequence
FDDI	Fiber Distribution Data Interface (ANSI)
FDIU	Fill Device Interface Unit
FEC	Forward Error Correction
FFS	Standard Berkeley Fast File System
FHSS	Frequency-Hopping Spread Spectrum
FIFO	First In, First Out Queue
FIN	Finish Flag (TCP)
FIPS	Federal Information Processing Standard (US)
FOCI	Foreign Owned, Controlled or Influenced
FOUO	For Official Use Only
FQDN	Fully-Qualified Domain Name
FR	Frame Relay
FRS	File Replication Service (Microsoft)
FRR	False Reject Rate
FSRS	Functional Security Requirements Specification
FSTS	Federal Secure Telephone Service
FTP	File Transfer Protocol
FTS	Federal Telecommunications System
FTAM	File Transfer Access Management
FTLS	Formal Top-Level Specification
FW-1	Firewall-1 (Checkpoint)

FYI	For Your Information
G	Giga; $1,000,000,000 = 10^9$ (bit rate) or $1,073,741,824 = 2^{30}$ (storage)
GAO	Government Accounting Office (US)
Gb	Giga-bits
GCCS	Global Command and Control System
GCFW	GIAC Certified Firewall Analyst
GCHQ	Government Communication Headquarters (UK)
GECOS	General Electric Comprehensive Operating System
GETS	Government Emergency Telecommunications Service
GHz	Giga-Hertz
GIAC	Global Information Assurance Certification
GIAC-TC	Global Information Assurance Certification-Training Center
GID	Group Identifier (Unix)
GIF	Graphic Interchange Format (Compuserve)
GIG-E	Gigabit Ethernet
GIMP	GNU Image Manipulation Program
GLB	Gramm Leach Bliley Act. (US)
GNU	GNU's Not Unix
GOTS	Government-off-the-Shelf
GPL	GNU Public License
GPO	Group Policy Object (Microsoft)
GPS	Global Positioning System
grep	Get Regular Expression and Print
GRUB	Grand Unified Bootloader
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
GWEN	Ground Wave Emergency Network
HDLC	High-Level Data Link Control (ISO)
HIDS	Host-based Intrusion Detection System
HIPAA	Health Insurance Portability and Accountability Act
HKLM	HKEY_LOCAL_MACHINE (Microsoft)
HMAC	Hashed Message Authentication Code
HR	Human Resources
HSRP	Hot Standby Router Protocol (Cisco)
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP over SSL
HUP	Hang-Up
HVAC	Heating, Ventilation, And Cooling
Hz	Hertz; cycles per second
I/O	Input/Output

IA	Information Assurance
I&A	Identification and Authentication
IANA	Internet Assigned Numbers Authority
IASIW	Institute for the Advanced Study of Information Warfare
IATO	Interim Approval to Operate
IBAC	Identity Based Access Control
IBM	International Business Machines Corp.
IBSS	Independent Basic Service Set
IC	Intelligence Community
ICE	Information Concealment Engine (Encryption)
ICF	Internet Connection Firewall (Microsoft)
ICMP	Internet Control Message Protocol
ICQ	Internet Call to Quarters, derived from military and ham radio CQ, or "call to quarters" signal; also derived from phrase "I seek you"
ICSA	International Computer Security Association
ICU	Interface Control Unit
ICV	Integrity Check Value (IPsec)
ID	Identifier or Intrusion Detection
IDC	International Data Corp.
IDE	Integrated (or Intelligent) Drive Electronics
IDEA	International Data Encryption Algorithm
IDS	Intrusion Detection System
IE	Internet Explorer (Microsoft)
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IEMATS	Improved Emergency Message Automatic Transmission System
IETF	Internet Engineering Task Force
IFF	Identification, Friend or Foe
IFFN	Identification, Friend, Foe or Neutral
IGMP	Internet Group Management Protocol
IGP	Interior Gateway Protocol
IHL	Internet Header Length (IP)
IIS	Internet Information Server (Microsoft)
IKE	Internet Key Exchange (IPsec)
ILS	Integrated Logistics Support
IMAP	Internet Message Access Protocol
INFOSEC	Information Systems Security
IO	Information Operations
IOS	Internetwork Operating System (Cisco)
IP	Internet Protocol or Instruction Pointer
IPSO	Internet Protocol Security Option

IPsec	IP Security Protocol
IPv4	IP Version 4
IPv6	IP version 6
IPX	Internetwork Packet Exchange Protocol (Novell)
IQUERY	Inverse query
IRC	Internet Relay Chat
IRDP	Internet Router Discovery Protocol
ISAKMP	Internet Security Association and Key Management Protocol (IPsec)
ISDN	Integrated Services Digital Network
ISM	Internet Service Manager (Microsoft) or Internet System Manager
ISN	Initial Sequence Number (TCP)
IS	Information Systems
ISO	International Organization for Standardization or Internet Security Officer
ISP	Internet Service Provider
ISS	Internet Security Systems, Inc.
ISSE	Information Systems Security Engineering
ISSM	Information Systems Security Manager
ISSO	Information Systems Security Officer
IT	Information Technology
ITAR	International Traffic in Arms Regulation
ITSEC	Information Technology Security Evaluation Criteria
ITU	International Telecommunication Union (formerly CCITT)
ITU-T	International Telecommunication Union Telecommunication Standardization Sector
IW	Information Warfare
JPEG	Joint Photographic Experts Group (ISO)
k, K	kilo; $1,000 = 10^3$ (bit rate; usually 'k') or $1,024 = 2^{10}$ (storage; usually 'K')
KAK	Key-Auto-Key
KDC	Key DistributionCenter (Keberos)
KDE	K Desktop Environment
KEK	Key Encryption Key
KG	Key Generator
KMC	KeyManagement Center
KMI	Key Management Infrastructure
KMID	Key Management Identification Number
KMODC	Key Management Ordering and DistributionCenter
KMP	Key Management Protocol
KMS	Key Management System
KP	Key Processor
KPK	Key Production Key
KSD	Key Storage Device

L2F	Layer 2 Forwarding
L2TP	Layer 2 Tunneling Protocol
L6	Bell Telephone Laboratories Low-Level Linked List Language
LAN	Local Area Network
LC3	L0phtCrack v3
LDAP	Lightweight Directory Access Protocol
LEAD	Low-Cost Encryption/Authentication Device
LFSR	Linear Feedback Shift Register
LILO	Linux Loader
LKM	Loadable Kernel Modules
LM	LAN Manager (Microsoft)
LMD	Local Management Device
LMD/KP	Local Management Device/Key Processor
LOCK	Logical Co-Processing Kernel
LPC	Linear Predictive Coding
LPD	Low Probability of Detection
LPI	Low Probability of Intercept
LRIP	Limited Rate Initial Preproduction
LSI	Large Scale Integration
LSB	Least Significant Bit
LSOF	List Open Files (tool)
M	Mega; $1,000,000 = 10^6$ (bit rate) or $1,048,576 = 2^{20}$ (storage)
MAC	Mandatory Access Control
MAC	Message Authentication Code
MAC	Media Access Control
MAN	<ol style="list-style-type: none"> 1. Metropolitan Area Network 2. Metropolitan Area Network
MB	Mega Bytes
Mb	Mega-bit
Mbps	Mega-bit per second
MBR	Master Boot Record
MBSA	Microsoft Baseline Security Analyzer
MD2	Message Digest 2
MD4	Message Digest 4
MD5	Message Digest 5
ME	Windows ME
MER	Minimum Essential Requirements
MHS	Message Handling System
MI	Message Indicator
MIB	Management Information Base

MIME	Multipurpose Internet Mail Extensions
MINIX	MINi-unIX
MINTERM	Miniature Terminal
MISSI	Multilevel Information Systems Security
MIT	Massachusetts Institute of Technology
MLS	Multilevel Security
MMC	Microsoft Management Console (Microsoft)
MO	Method of Operations
MOE	Measure Of Effectiveness
MOM	Microsoft Operations Manager
MP3	MPEG Audio Layer 3
MPEG	Motion (or Moving) Picture Experts Group (ISO)
MPLS	Multiprotocol Label Switching
MS	Microsoft
MSAU	Multistation Access Unit
MSB	Most Significant Bits
MSE	Mobile Subscriber Equipment
MTA	Metropolitan Transit Authority (Boston)
MTU	Maximum Transmission Unit
NACAM	National COMSEC Advisory Memorandum
NACSI	National COMSEC Instruction
NACSIM	National COMSEC Information Memorandum
NAI	Network Associates, Inc.
NAK	Negative Acknowledge
NAPT	Network Address and Port Translation
NASL	Nessus Attack Scripting Language
NAT	Network Address Translation
NCCD	Nuclear Command and Control Document
NCS	<ol style="list-style-type: none"> 1. National Communications System 2. National Cryptologic School 3. Net Control Station
NCSC	National Computer Security Center
NCSA	National Computer Security Association (now the ICSA)
NDA	Non-Disclosure Agreement
NDS	NetWare Directory Services (Novell)
NetBIOS	Network Basic Input/Output System (Microsoft)
NFPA	National Fire Protection Association
NFR	Network Flight Recorder
NFS	Network File System

NIC	Network Interface Card
NIDS	Network-Based Intrusion Detection System
NIPC	National Infrastructure Protection Center
NIS	Network Information Service
NISAC	National Industrial Security Advisory Committee
NIST	National Institute of Standards and Technology (U.S.)
NLZ	No-Lone Zone
NMAP	Network Mapping Tool
NNTP	Network News Transfer Protocol
NRM	Normal Response Mode
NSA	National Security Agency (U.S.)
NSD	National Security Directive
NSDD	National Security Decision Directive
NSEP	National Security Emergency Preparedness
NSI	National Security Information
NSTAC	National Security Telecommunications Advisory Committee
NSTISSAM	National Security Telecommunications and Information Systems Security Advisory/Information Memorandum
NSTISSC	National Security Telecommunications and Information Systems Security Committee
NSTISSD	National Security Telecommunications and Information Systems Security Directive
NSTISSI	National Security Telecommunications and Information Systems Security Instruction
NSTISSP	National Security Telecommunications and Information Systems Security Policy
NSWC	Naval Surface WarfareCenter (U.S. Navy)
NT	Windows NT
NT4SP2	Windows NT 4.0 Service Pack 2
NTCB	Network Trusted Computing Base
NTFS	Windows NT File System (Microsoft Windows NT/2000)
NTIA	National Telecommunications and Information Administration
NTISSAM	National Telecommunications and Information Systems Security Advisory/Information Memorandum
NTISSD	National Telecommunications and Information Systems Security Directive
NTISSI	National Telecommunications and Information Systems Security Instruction
NTISSP	National Telecommunications and Information Systems Security Policy
NTLM	Windows NT LAN Manager (Microsoft)
NTLM2	Windows NT LAN Manager version 2 (Microsoft)
NTP	Network Time Protocol
NVRAM	Non-Volatile Random Access Memory
NYSE	New York Stock Exchange
OADR	Originating Agency's Determination Required
ODBC	Open Database Connectivity (Microsoft)
OEM	Original Equipment Manufacture

OFB	Output FeedBack Mode
OI	Order Information
ONB	Optimal Normal Base mathematics (encryption)
OOB	Out of Band
OPCODE	Operations Code
OPSEC	Operations Security
ORA	Organizational Registration Authority
OS	Operating System
OSI	Open Systems Interconnect
OSPF	Open Shortest Path First
OSR2	Windows 95 Service Release
OTAD	Over-the-Air Key Distribution
OTAR	Over-the-Air Rekeying
OTAT	Over-the-Air Key Transfer
OTP	One-Time Pad
OTT	One-Time Tape
OU	Organization Unit
OUI	Organizationally Unique Identifiers
PAA	(PKI) Policy Approving Authority (IC) Principal Accreditating Authority
PAE	Port Authentication Entity
PAL	Permissive Action Link
PAM	Pluggable Authentication Modules
PAN	Personal Area Network
PAP	Password Authentication Protocol
PARMS	Parallel Algebraic Recursive Multilevel Solver
PBR	Partition Boot Record
PBX	Private Branch Exchange
PC	Personal Computer
PCA	Policy Certification Authority
PCIPB	President's Critical Infrastructure Protection Board
PCMCIA	Personal Computer Memory Card International Association
PDA	1. Personal Data Assistant 2. Personal Digital Assistant
PDC	Primary Domain Controller (Microsoft Windows NT)
PDR	Preliminary Design Review
PDS	1. Protected Distribution Systems 2. Practices Dangerous to Security
PEAP	Protected Extensible Authentication Protocol

PEM	Privacy Enhanced Email
PES	Positive Enable System
PGP	Pretty Good Privacy
PHP	PHP: Hypertext Preprocessor
PI	Payment Information
PID	Process Identifier (Unix)
PIM	Personal Information Management
PIN	Personal Identification Number
PING	Packet InterNet Groper
PKC	Public Key Cryptography
PKI	Public Key Infrastructure
PKSD	Programmable Key Storage Device
P Model	Preproduction model
PNEK	Post-Nuclear Event Key
POP	Point Of Presence
POP3	Post Office Protocol v3
POS	Point-of-Sale
POST	Power On Self Test
PPL	Preferred Products List (a section in the INFOSEC Products and Services Catalogue)
PPP	Point-to-Point Protocol
PPTP	Point-to-Point Tunneling Protocol
PRBAC (C.F.D.)	Partition Rule Base Access Control
PROPIN	Proprietary Information
PSH	Push Flag (TCP)
PSYOP	Psychological Operation
PVC	Permanent Virtual Circuits
PWB	Programmers' Workbench
PWDSD	Protected Wireline Distribution System
QA	Quality Assurance
QoS	Quality of Service
qotd	Quote-of-the-Day Service (Unix)
QS	Quality System Requirements
R&D	Research and Development
RA	Risk Analysis or Risk Assessment
RADIUS	Remote Authentication Dial In User Service
RAID	Redundant Array of Independent Disks
RAM	Random Access Memory
RAMP	Rating Maintenance Program
RARP	Reverse Address Resolution Protocol
RAS	Remote Access Server (Microsoft Windows NT/2000)

RC4	Rivest Cipher (or Ron's Code) #4
RC6	Rivest Cipher (or Ron's Code) #6
RDP	Remote Desktop Protocol (Microsoft)
RDS	Remote Data Service (Microsoft)
RF	Radio Frequency
RFC	Request for Comments (IETF)
RFP	Request for Proposal
RIAA	Recording Industry Association of America
RID	Relative Identifier
RIP	Routing Information Protocol
RO	Read-Only
ROI	Return On Investment
ROM	Read-Only Memory
ROT	Rotation Forward (Ciphers)
RPC	Remote Procedure Call
RPII	Radiological Protection Institute of Ireland
RPM	Red Hat Package Manager
RRAS	Routing and Remote Access Service
RSA	Rivest, Shamir, and Adleman
RSBAC	Rule Set Based Access Control
RST	Reset Flag (TCP)
RW	Read-Write
S/KEY	S/KEY One-Time Password System (Bellcore, now Telcordia)
SA	<ol style="list-style-type: none"> 1. Security Associations 2. Security Administrator
SABI (C.F.D.)	Secret and Below Interoperability
SACL	System Access Control List
SAINT	Security Administrator's Integrated Network Tool
SAM	Security Account Manager (Microsoft Windows NT/2000)
SANS	SysAdmin, Network, Security
SAO	Special Access Office
SAP	<ol style="list-style-type: none"> 1. System Acquisition Plan 2. Special Access Program
SARA	Security Auditor's Research Assistant
SARK	SAVILLE Advanced Remote Keying
SAT	Security Access Token (Microsoft)
SATAN	Security Administrator's Tool for Analyzing
SBS	Small Business Server (Microsoft) or Step-by-Step (SANS)
SBU	Sensitive But Unclassified

SCA	Security Configuration and Analysis (Microsoft)
SCAT	Security Configuration and Analysis Tool (Microsoft Windows 2000)
SCCS	Source Code Control System
SCI	Sensitive Compartmented Information
SCIF	Sensitive Compartmented Information Facility
SCM	Security Configuration Manager
SCO	Santa Cruz Organization
SCSI	Small Computer System Interface (ANSI)
SCU	System Configuration Utility (Microsoft Windows NT/2000)
SDK	Software Development Kit
SDCL	Synchronous Data Link Control
SDNS	Secure Data Network System
SDR	System Design Review
SEC	Securities and Exchange Commission (U.S.)
SEQ	Sequence Number
SESAME	Secure European System for Applications in a Multi-vendor Environment
SET	Secure Electronic Transaction (MasterCard, Visa, et al.)
SF	Syn/Fin Data Flag
SFA	Security Fault Analysis
SFC	System File Checker
SFUG (C.F.D.)	Security Features Users Guide
SGI	Silicon Graphics Indy
SGID	Set Group Identifier
SHA	Secure Hash Algorithm (NIST)
SHS	Secure Hash Standard (NIST)
SI	Special Intelligence
SID	Security ID Number (Microsoft)
SIGGEN	Special Interest Group for natural language GENeration
SIP	Session Initiation Protocol
SISS	Subcommittee on Information Systems Security
SKC	Secret Key Cryptography
SKEME	Secure Key Exchange Mechanism
SKIP	Simple Key-Management for Internet Protocols (Sun Microsystems)
SLE	Single Loss Expectancy
SLIP	Serial Line IP
SMB	Server Message Block
S/MIME	Secure Multipurpose Internet Mail Extensions
SMTP	Simple Mail Transfer Protocol
SMU	Secure Mobile Unit
SNA	Systems Network Architecture
SNAPLEN	Snapshot Length

SNAT	Source Network Address Translation
SNMP	Simple Network Management Protocol
SOA	Start of Authority
SOHO	Small Office/Home Office
SOP	Standard Operating Procedure
SP	Service Pack (Microsoft)
SPF	Shortest Path First algorithm
SPI	Security Profile Inspector for Unix Networks
SPID	Service Profile Identifier
SPK	Single Point Key(ing)
SPX	Sequenced Packet Exchange
SQL	Structured Query Language
SRR	Security Requirements Review
SSH	Secure Shell
SSI	Server Side Includes
SSID	Service Set Identifier (IEEE 802.11b)
SSL	Secure Sockets Layer (Netscape)
SSP	System Security Plan
ST&E	Security Test and Evaluation
STE	Secure Terminal Equipment
STP	Shielded Twisted Pair
STS	<ol style="list-style-type: none"> 1. Station To Station 2. Subcommittee on Telecommunication Security
STU	Secure Telephone Unit
SUID	Set User Identifier
SUS	Software Update Services (Microsoft)
SVC	Switched Virtual Circuit
SVRx	Unix System V Revision x
SW	Software
SYN	Synchronize Sequence Number Flag (TCP)
Syslog	System Logger
SYSV	System V Unix
TA	Traffic Analysis
TACACS	Terminal Access Controller Access Control System
TACTERM	Tactical Terminal
TAG	TEMPEST Advisory Group
TCB	Trusted Computing Base
tar	Tape Archive (Unix)
TCP/IP	Transmission Control Protocol/Internet Protocol
TED	Trunk Encryption Device

TEK	Traffic Encryption Key
TEP	TEMPEST Endorsement Program
TFM	Trusted Facility Manual
TFN	Tribe Flood Network
TFN2K	Tribe Flood Network 2000
TFS	Traffic Flow Security
TFTP	Trivial File Transfer Protocol
TGS	Ticket Granting Server (Keberos)
TGT	Ticket Granting Ticket Request (Keberos)
TKIP	Temporal Key Integrity Protocol
TLS	<ol style="list-style-type: none"> 1. Transport Layer Security 2. Top-Level Specification
TOC	Time of Check
TOE	Target of Evaluation
TOS	Type of Service (IP)
TOU	Time of Use
TP	Transformation Procedure
TPC	Two-Person Control
TPEP	Trusted Products Evaluation Program
TPI	Two-Person Integrity
TRANSEC	Transmission Security
TRB	Technical Review Board
TRI-TAC	Tri-Service Tactical Communications System
TSABI (C.F.D.)	Top Secret and Below Interoperability
TSCM	Technical Surveillance Countermeasures
TSEC	Telecommunications Security
TSIG	Transaction signature (DNS)
TTAP	Trust Technology Assessment Program
TTL	Time To Live (IP)
TTY	Teletypewriter
UA	User Agent
UAPRSF	Urgent, Ack, Push, Reset, Syn, Finish flags (TCP)
UDP	User Datagram Protocol
UID	User Identifier (Unix)
UIS	User Interface System
UNC	Universal Naming Convention
UNIX	From UNICS (Uni-plexed Information and Computing System)
UPP	User Partnership Program
UPS	Uninterruptible Power Supply

URG	Urgent Data Flag (TCP)
URL	Uniform Resource Locator
US	United States
UTP	Unshielded Twisted Pair
VAX	Virtual Address eXtension
VBS	VisualBASIC Script (Microsoft)
VCI	Virtual Channel Identifier
VDSL	Very high bit rate Digital Subscriber Line
VGanyLAN	Virtual Grade any Local Area Network
VLAN	Virtual Local Area Network
VM	Virtual Machine
VoIP	Voice over IP
VPI	Virtual Path Identifier
VPN	Virtual Private Network
VxDS	Virtual Device Driver
W	Watt (unit of power)
W2K	Windows 2000
W3C	Word Wide Web Consortium
WAN	Wide Area Network
WAP	Wireless Application Protocol
WEP	Wired Equivalent Privacy (IEEE 802.11b)
WINS	Windows Internet Name Service (Microsoft)
WLAN	Wireless Local Area Network
WM97	Word Macro Virus
WMI	Windows Management Instrumentation (WMI)
WMLscript	Wireless Markup Language (WML) Scripting Language
WPA-I	WiFi Protected Access specification I
WSH	Window Scripting Host
WTC	WorldTradeCenter (New York City)
WTLS	Wireless Transport Layer Security
WWW	World Wide Web
XDM	X Display Manager
XDMCP	X Display Manager Control Protocol
XML	Extensible Markup Language
XOR	Exclusive OR
Y2K	Year 2000
YMMV	Your Mileage May Vary
YP	Yellow Pages (Network Information Service)

