Assessing UNIX Security

Kumar Setty – Solo Cup Company

Intended Audience

- IT and Financial Auditors
- Finance and Accounting professionals
- Anyone who wishes to gain an understanding of the risks and controls surrounding UNIX.

Unintended Audience

- UNIX security administrators – if you are here, please feel free to sleep.

Motivations

- Noticed that the ISACA audit program is incomplete and not up to date with the most recent developments and practices.
- Lack of a centralized and comprehensive set of guidelines that is flexible enough to enable an auditor to assess the risks and determine a proper course of action.
- In the process of developing a WikiBook and requesting feedback and experiences of other auditors.

Objectives

- Share my experiences and knowledge of assessing UNIX security.
- Elucidate some of the details of the UNIX operating system and its relevance to common audit projects and efforts.
- Educate auditors in the risks and controls relating to UNIX.
- To offer a conceptual understanding on how to conduct a risk-assessment for a UNIX OS within specific time and resource constraints.

Observations and Trends

- Software and OS vendors are beginning to take the lead in decoding the security design and architecture of their products.
- In some cases, vendors now offer a much more comprehensive view of the security design of their products.

Excuses, Excuses, Excuses…

- *We don’t have time to audit all of our UNIX servers.*
- *Don’t worry about UNIX. Look at the end-user applications.*
- *Vendor patches will address your security concerns.*
- *UNIX is not plagued with security issues like MS Windows.*
- *We have a firewall. Isn’t that enough?*

Common Misconception

- ***"UNIX is more secure than Windows."***
  - In fact the opposite is true.
  - Most of the ire and frustration is focused on Windows, IE, and other MS applications.
  - UNIX has more bugs due to its "biodiversity", but these bugs are addressed and fixed more quickly due to the open source design of UNIX and Linux.

Incentivizing IT to Assess UNIX Security

- Scare tactics will not work. You will only increase resistance.
- Time, money, IT budgets are at a premium.
- How do we convince IT to take the time to assess UNIX security configurations?
- Status quo is unacceptable.

Incentivizing IT to Assess UNIX Security

- Do your homework prior to the assessment.
- Clearly outline the risks of not addressing security gaps in UNIX. Keep in mind that security is a broad concept.
  - Start with the overall risk to the enterprise.
    - Risks to reputation, business discontinuity
  - Then mention specific risks.
    - Data corruption, data loss, data theft
- Cost of remediation
  - More expensive to make changes to production systems.

UNIX – A Brief History

- Originated in 1969 at Bell Labs by Dennis Ritchie and Ken Thompson.
- In 1974, UNIX became the first OS written in C.
- UNIX became the first non-proprietary, open or standard OS that could be enhanced or improved by anyone.
- Over the past four decades, the UNIX OS has been bundled by various vendors in varying workstation products – i.e. Sun Microsystems, IBM, Hewlett-Packard.

Who Uses UNIX?

- Corporations
  - Google uses Ubuntu Linux for everything.
- Universities
  - Very high adoption rate.
  - Universities are huge targets due to their “open” nature.
  - One large local university has undergone a hardening of their entire UNIX infrastructure
    - Using Ubuntu UFW (uncomplicated firewall) on all desktops.
    - Eliminated sendmail, Telnet, FTP.

UNIX – Popularization of Email

- In the 1980s, the sendmail utility was packaged within the BSD UNIX OS. Subsequently, the sendmail program on BSD UNIX has gone on the become the most commonly used SMTP server on the Internet.

Why UNIX?

Why did the UNIX OS gain so many early adopters?

- Multi-tasking capability
- Multi-user capability
- Portability
- UNIX programs and utilities
- Library of application software
- TCP/IP ready
- “Biodiversity”
- Free variants such as Linux

UNIX - Advantages

- **Full multitasking with protected memory.** Multiple users can run multiple programs each at the same time without interfering with each other or crashing the system.
- **Access controls and security.** All users must be authenticated by a valid account and password to use the system at all. All files are owned by particular accounts. The owner can decide whether others have read or write access to his files.

UNIX Advantages

- **A rich set of small commands and utilities** that do specific tasks well -- not cluttered up with lots of special options. UNIX is a well-stocked toolbox, not a giant do-it-all Swiss Army Knife.

- **A powerfully unified file system.** Everything is a file: data, programs, and all physical devices. Entire file system appears as a single large tree of nested directories, regardless of how many different physical devices (disks) are included.

UNIX Advantages

- A **lean kernel** that does the basics for you but doesn't get in the way when you try to do the unusual.
- **Portable** – works on a wide variety of machines.
- The Unix operating system offers an **efficient** **level of virtual memory**. What this means for the user is that you can use a number of programs at the same time using only a modest level of physical memory. The system can handle several programs at once without severely pulling on the system's resources.

UNIX - Disadvantages

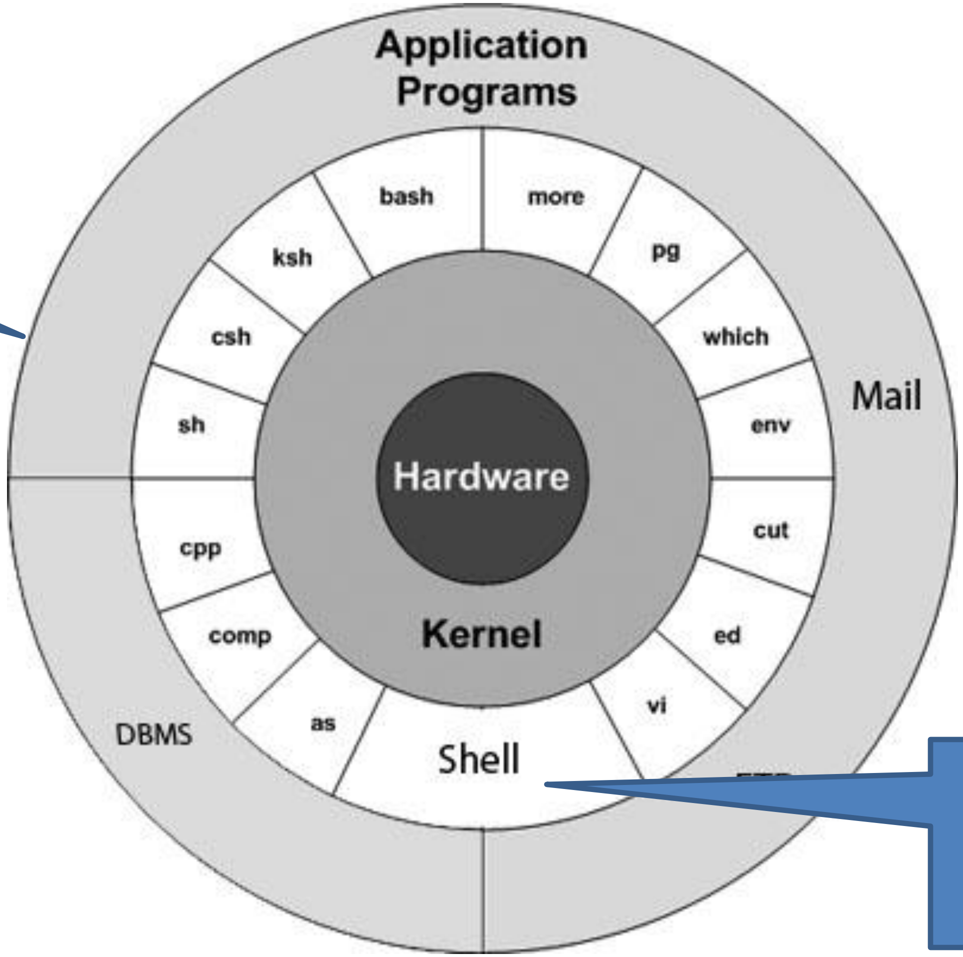
- Not too user-friendly.
- To use Unix well, you need to understand some of the main design features. Its power comes from knowing how to make commands and programs interact with each other, not just from treating each as a fixed black box.
- Steeper learning curve
  - Use UNIX “man” or ***manual*** command.
  - If you have no UNIX administration experience, you have to spend some effort studying and practicing.
- Newer versions of Linux are very user friendly.

How Does UNIX Work?

UNIX is functionally organized at three levels:

- Kernel
  - The heart of UNIX.
  - Responsible for resource allocation, security, and low-level interfaces with hardware
- Shell
  - The component of Unix that you will interact with is the shell.
- Programs
  - ***Utilities*** - software tools included with the Unix operating system that let you do work such as text editing, programming, and communications.
  - ***Process*** is an instance of a program that is being executed by the operating system.

How Does UNIX Work?



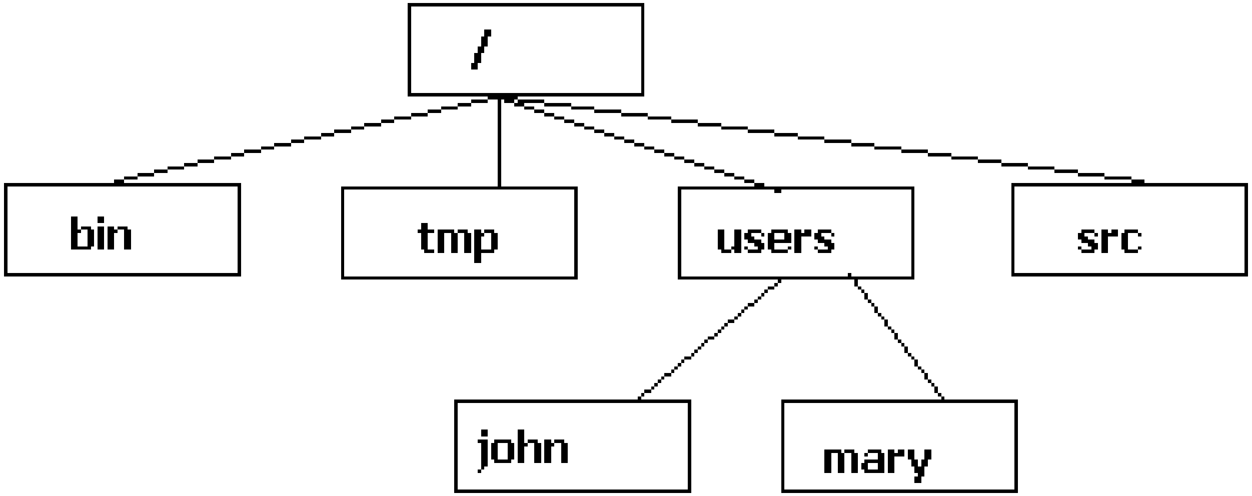
Server Software

The firewall protects the outer perimeter. The inner perimeter may be accessed using FTP, Telnet, or ssh.

Access to the applications

UNIX File System

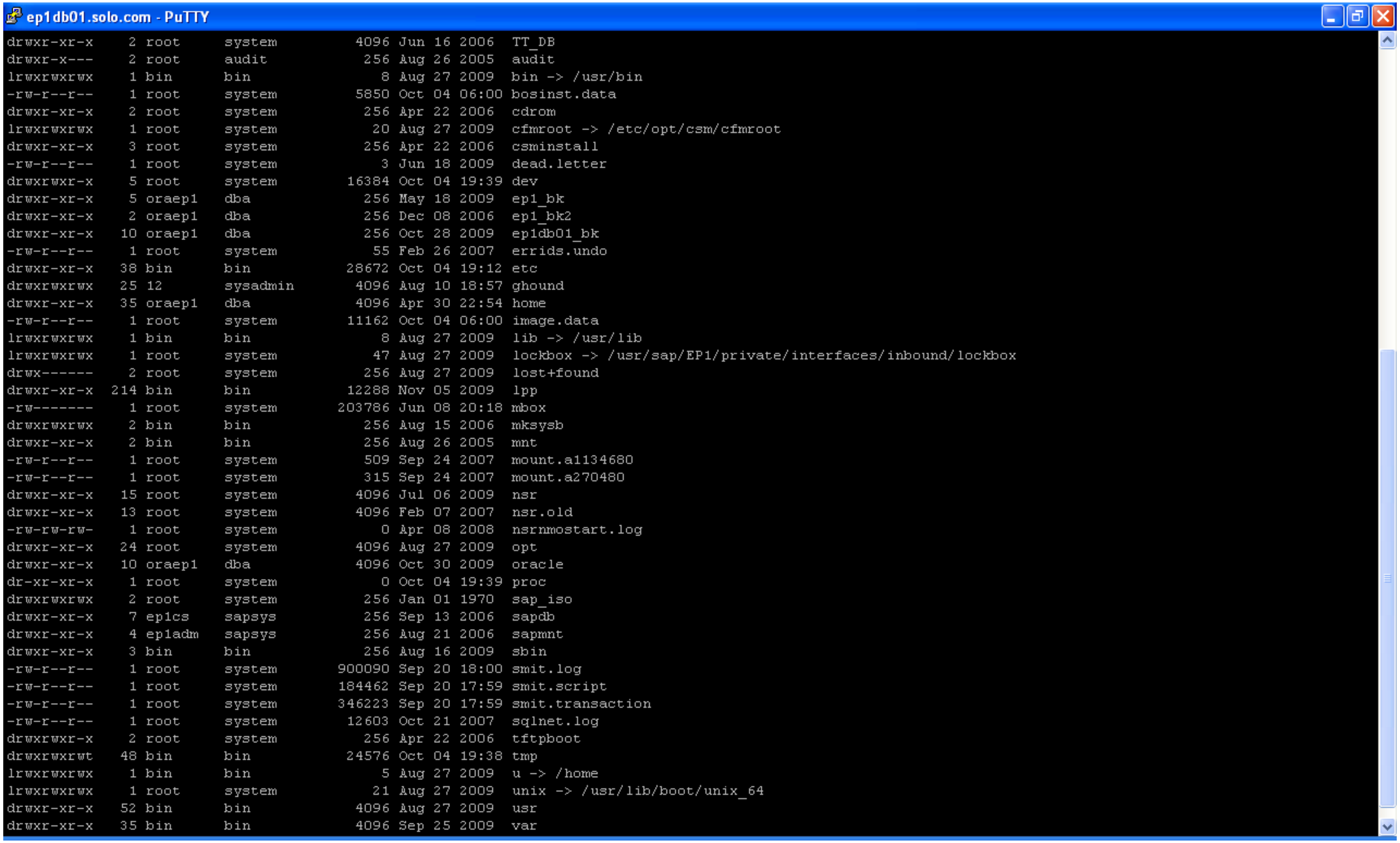
- Think of the UNIX file system as an upside down tree:



Navigating Through UNIX File System

- The UNIX administrator may set up the file system in any way desired.
- Some UNIX commands are similar to commands used in navigating through a MS-DOS environment.
  - / = root directory
  - pwd = present working directory
  - cd = change directory
  - .. = up one directory

File and Directory Privileges

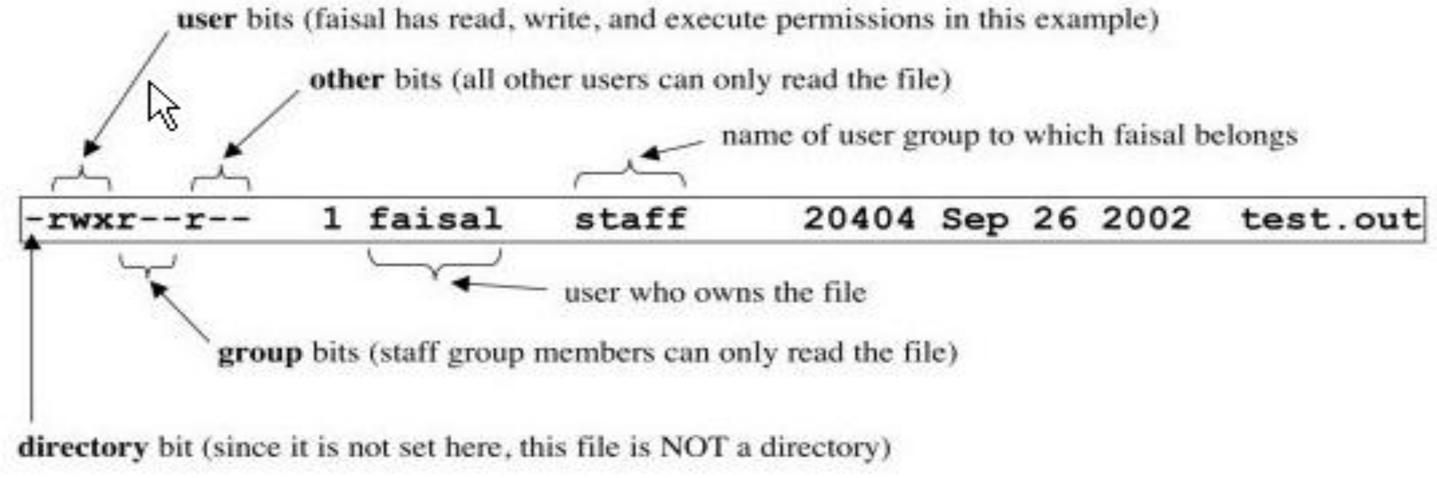


```
drwxr-xr-x    2 root     system        4096 Jun 16 2006  TT_DB
drwxr-x---    2 root     audit          256 Aug 26 2005  audit
lrwxrwxrwx    1 bin      bin              8 Aug 27 2009  bin -> /usr/bin
-rw-r--r--    1 root     system        5850 Oct 04 06:00 bosinst.data
drwxr-xr-x    2 root     system         256 Apr 22 2006  cdrom
lrwxrwxrwx    1 root     system          20 Aug 27 2009  cfmroot -> /etc/opt/csm/cfmroot
drwxr-xr-x    3 root     system         256 Apr 22 2006  csminstall
-rw-r--r--    1 root     system           3 Jun 18 2009  dead.letter
drwxrwxr-x    5 root     system       16384 Oct 04 19:39 dev
drwxr-xr-x    5 oraep1   dba            256 May 18 2009  ep1_bk
drwxr-xr-x    2 oraep1   dba            256 Dec 08 2006  ep1_bk2
drwxr-xr-x   10 oraep1   dba            256 Oct 28 2009  ep1db01_bk
-rw-r--r--    1 root     system          55 Feb 26 2007  errids.undo
drwxr-xr-x   38 bin      bin          28672 Oct 04 19:12 etc
drwxrwxrwx   25 12       sysadmin      4096 Aug 10 18:57 ghound
drwxr-xr-x   35 oraep1   dba           4096 Apr 30 22:54 home
-rw-r--r--    1 root     system       11162 Oct 04 06:00 image.data
lrwxrwxrwx    1 bin      bin              8 Aug 27 2009  lib -> /usr/lib
lrwxrwxrwx    1 root     system          47 Aug 27 2009  lockbox -> /usr/sap/EP1/private/interfaces/inbound/lockbox
drwx------    2 root     system         256 Aug 27 2009  lost+found
drwxr-xr-x  214 bin      bin          12288 Nov 05 2009  lpp
-rw-------    1 root     system      203786 Jun 08 20:18 mbox
drwxrwxrwx    2 bin      bin            256 Aug 15 2006  mksysb
drwxr-xr-x    2 bin      bin            256 Aug 26 2005  mnt
-rw-r--r--    1 root     system         509 Sep 24 2007  mount.a1134680
-rw-r--r--    1 root     system         315 Sep 24 2007  mount.a270480
drwxr-xr-x   15 root     system        4096 Jul 06 2009  nsr
drwxr-xr-x   13 root     system        4096 Feb 07 2007  nsr.old
-rw-rw-rw-    1 root     system           0 Apr 08 2008  nsrnmostart.log
drwxr-xr-x   24 root     system        4096 Aug 27 2009  opt
drwxr-xr-x   10 oraep1   dba           4096 Oct 30 2009  oracle
dr-xr-xr-x    1 root     system           0 Oct 04 19:39 proc
drwxrwxrwx    2 root     system         256 Jan 01 1970  sap_iso
drwxr-xr-x    7 ep1cs    sapsys         256 Sep 13 2006  sapdb
drwxr-xr-x    4 ep1adm   sapsys         256 Aug 21 2006  sapmnt
drwxr-xr-x    3 bin      bin            256 Aug 16 2009  sbin
-rw-r--r--    1 root     system      900090 Sep 20 18:00 smit.log
-rw-r--r--    1 root     system      184462 Sep 20 17:59 smit.script
-rw-r--r--    1 root     system      346223 Sep 20 17:59 smit.transaction
-rw-r--r--    1 root     system       12603 Oct 21 2007  sqlnet.log
drwxrwxr-x    2 root     system         256 Apr 22 2006  tftpboot
drwxrwxrwt   48 bin      bin          24576 Oct 04 19:38 tmp
lrwxrwxrwx    1 bin      bin              5 Aug 27 2009  u -> /home
lrwxrwxrwx    1 root     system          21 Aug 27 2009  unix -> /usr/lib/boot/unix_64
drwxr-xr-x   52 bin      bin           4096 Aug 27 2009  usr
drwxr-xr-x   35 bin      bin           4096 Sep 25 2009  var
```

File Access Permissions

Each file has a set of protection or mode bits. The first bit is the directory bit. The next set of three placeholders represents user, the next set represents group, and the last represents other or everyone else. These three categories represent how users can access the file or directory. This set of permissions may be represented in an octal number. This translates to Read (“r”) = 4 = 2^2 ; Write (“w”) = 2 = 2^1 ; Execute (“x”) = 1 = 2^0.

Examples:
-rw------- is mode 600
-rwxrwxrwx is mode 777
-rw-rw-rw- is mode 666



user bits (faisal has read, write, and execute permissions in this example)

other bits (all other users can only read the file)

name of user group to which faisal belongs

```
-rwxr--r--   1 faisal   staff     20404 Sep 26 2002  test.out
```

user who owns the file

group bits (staff group members can only read the file)

directory bit (since it is not set here, this file is NOT a directory)

File Access Permissions

- Examples:
  - drwxr-xr-x 2 root system 4096 Jun 16 2006 TT_DB
    - Octal number = 755
  - lrwxrwxrwx 1 root system 20 Aug 27 2009 cfmroot -> /etc/opt/csm/cfmroot
    - “L” means “link”
    - Octal number = 777
  - -rw-r--r-- 1 ksetty usr 54996 May 28 19:18 kumarnetstatoutput
    - Octal number = 611

Umask command

- ***umask*** is used to reveal or change the current default file or directory permissions. Default permissions are assigned by the system whenever you create a new file or directory, and these are governed by the umask setting.
- The digits in the umask number are 'subtracted' from 777 for directories or 666 for files when you are creating their initial permissions
- Examples:
  - ***000*** means that new files will have permission of 666 and new directories 777.
  - ***022*** means that new files will have permission of 644 (-rw-r—r--) and new directories 755 (drwxr-xr-x).

Root Account

- ***Root*** is represented by “/”.
- Superuser privileges – root can do **anything**.
- Can access any files or programs and may run any command. If you want to handle a file or a process you have to be the owner.
- The primary goal of every intruder is to get access to the root account.

Variety Is the Spice of Life

- Many versions of UNIX exist but three vendors remain predominant in the marketplace:
  - Sun Solaris
  - HP-UX
  - IBM AIX
- Linux variants:
  - Red Hat
  - Ubuntu
    - Rapidly increasing adoption rate
    - Google uses Ubuntu for many of its servers.

Vendor Preference

- As with any UNIX variant, many times the choice is based on a personal preference.
- Each vendor will claim to offer the best solutions for the best price.
- All vendors will claim to have the most robust security but in most cases practice “security by obscurity.”

A Sample Audit – Case Study

- Law Firm of Dewey Cheatham & Howe LLP has 10 production UNIX servers supporting their critical applications.
- Client Benjamin Dover & Co. has asked Dewey Cheatham & Howe LLP, to perform an audit of their UNIX OS implementations within three days.
- Contingent on the results of the audit, Benjamin Dover & Co. will retain the services of Dewey Cheatham & Howe LLP.

Sample Audit Walkthrough - Preparation

- Determine risks and scope of audit.
- Develop an audit program.
- Connect to servers using **PuTTY**.
- Verify privileges and ownership on critical accounts.
- Verify privileges and ownership on critical files, utilities, programs.
- Assumptions:
  - No tools or software used in assessment.
  - All commands run from command line in shell.

Risks

- Simplified Risk Quantification:
  - – Low Risk Analogy
    - Laughing while drinking milk

  - – Medium Risk Analogy
    - Skateboarding down a flight of steps

  - – High Risk Analogy
    - Playing tag with a grizzly bear

High Risk

May result in a remote or local attacker obtaining root-level access, with or without a shell, to the system. Also includes concerns relating to irrecoverable loss of critical data.

Medium Risk

May result in a remote attacker obtaining access to system’s resources as a non-privileged user. Also includes denial of service concerns relating to services and resources.

Low Risk

May result in a local attacker obtaining non-privileged user access to data and resources that should not be accessible to an external user. Also includes concerns relating to highly improbable events.

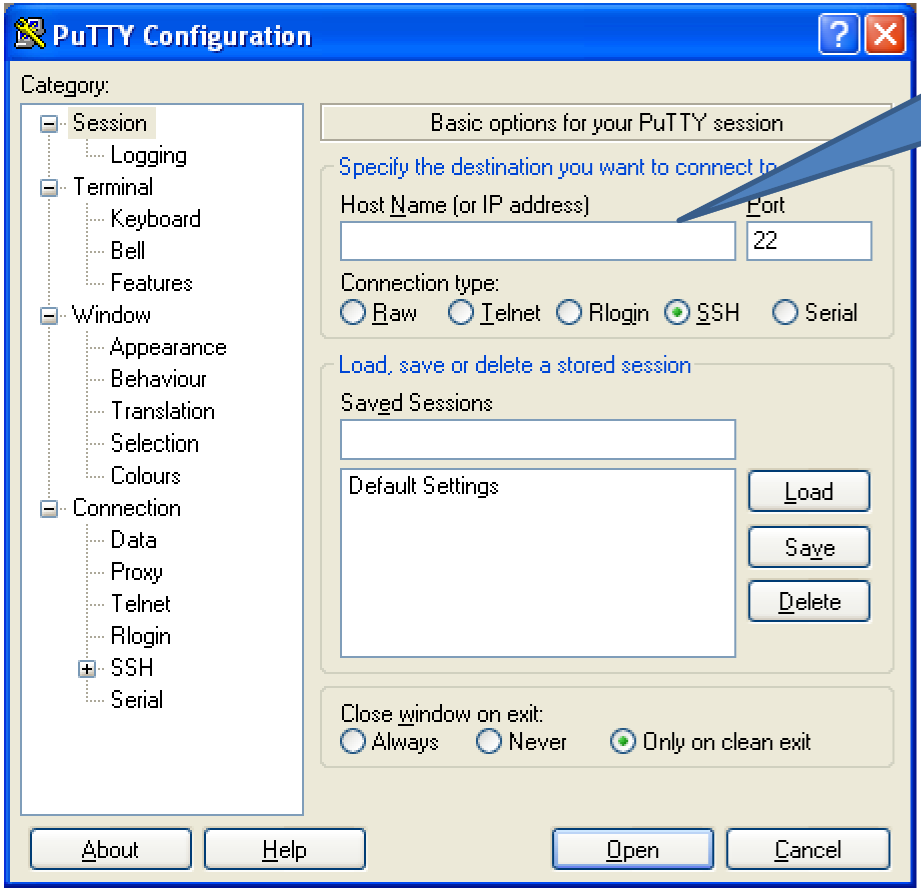
Risks - Brass Tacks

- The barbarians are inside the gates.
  - Have we appropriately assigned privileges and accounts to the right people?
- The flexible nature of UNIX introduces most of the risks.
  - TCP/IP
  - Multi-user capability
  - Utilities relying on plaintext authentication.
    - Because anyone monitoring (eavesdropping) on the network transmission can intercept and use those passwords to gain access to your systems. Many common intrusions include the use of a sniffer (eavesdropping software): a intruder will install a sniffer just to see if they can pick up some good passwords while they are in the neighborhood.

PuTTY

- A free and open source terminal emulator application which can act as a client for the SSH, Telnet, rlogin, and raw TCP computing protocols and as a serial console client.
- http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
- Check on Google.com for the latest download site.
- Supports PKI (public key infrastructure)

PuTTY Setup



Host Name (or IP address):

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Saved Sessions:

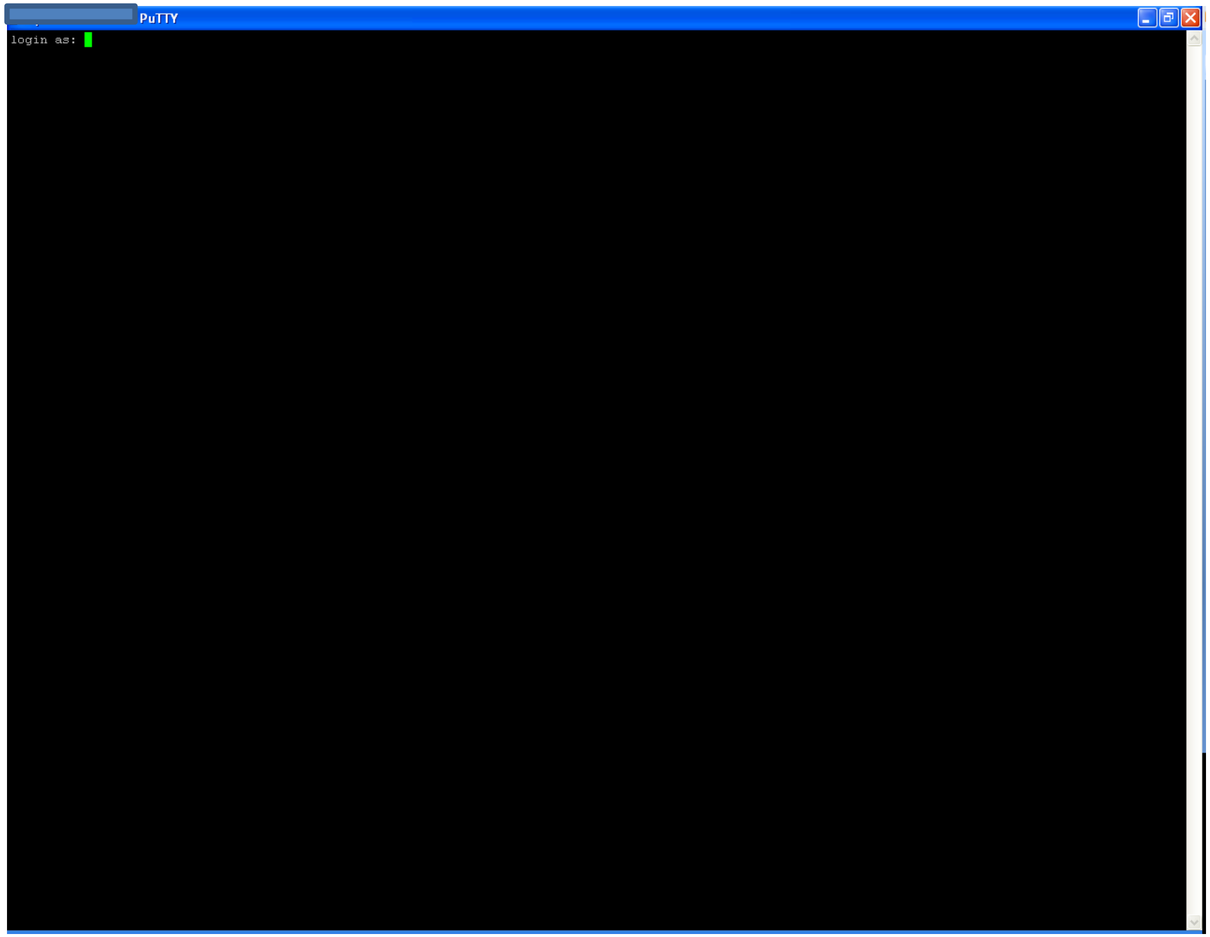
Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

1. Host name
2. IP address
3. Other URL

Consult with your UNIX administration team.

Now What?



```
login as: 
```

Considerations

- Identify the privileges on high-risk directories and files.
- Have communications been secured between the server and the outside world?
- Are users within the enterprise being monitored without adding too much overhead?

Step 1 – Where Am I?

- Login to server using PuTTY.
- Type ***pwd***
  - More than likely you will be in your home directory.
- Navigate to root, “/” by typing ***cd /***
- Type ***ls –l*** to view a list of files and directories.

Step 2 – High Risk Items

- Root Access
- Set user ID (SUID) and Set group ID (SGID)
- Verification of restoration tests from disk or tape
- Outdated virus definitions on internal workstations
- Default accounts
- Password controls
- Sensitive files and directories
- Privileged programs
- Sendmail
- Network security – commands
- Network security – files
- Patch management

Root Access

- Determine if **only one account** is defined as root (uid=0) by issuing the command awk -F":" '$3==0 {print $0}' /etc/passwd
- Determine if there is monitoring of the root account. Confirm that users have to "su" to root.
  - Check the **/etc/ssh/sshd_config** file by typing in the command string ***egrep -v '^#|^[ ]*$' /etc/ssh/sshd_config | uniq***
  - The resulting text should show: **PermitRootLogin = "No"**

Root Access

- Other parameters to verify in the sshd_config file:
  - PasswordAuthentication no
  - KbdInteractiveAuthentication no
  - ChallengeResponseAuthentication no
  - RSAAuthentication yes
  - PubkeyAuthentication yes
- The above settings prevent interactive logins with user and password.
- Also consider using the ***sudo*** package
  - "substitute user do"
  - Verify if sudo is being used by checking **/etc/sudoers**

Root Access

- Root’s logon files (e.g., the .profile, .cshrc, .kshrc) should not run any other files not owned by root or which are group or world writable.
  - Check each of root account logon scripts to confirm if other files are being launched from root’s logon files – type ***more .profile*** and confirm that there is no “..” at the end of the PATH parameter within the file.
  - This ensures that the wrong program does not get executed.

Root – Best Practices

- Should be protected by a strong password.
- Direct root logins should be disabled and all users having access to root should be forced to login using their named user account and then should “su” or “switch user” to root.
- Root logins for all servers may be tracked using tools such as ***Splunk***.
- These root logons should be examined on a periodic basis and archived for audit purposes.

SUID and SGID

- Think of SUID and SGID as enabling impersonation.
- **SUID** is “set user id upon execution”
- **SGID** is “set group id upon execution”
- Both SUID and SGID allow users to run an executable with the permissions of the executable's owner or group.
- Often used to allow users on a computer system to run programs with temporarily elevated privileges in order to perform a specific task.
- Commonly used for running backups, su, ping, etc. Therefore, not feasible to turn this function off.

SUID and SGID

- SGID can be set on directories. SUID permission is ignored.
- Setting the setgid permission on a directory (chmod g+s) causes new files and subdirectories created within it to **inherit <u>its</u> groupID**, rather than the primary groupID of the user who created the file.
- Setting setgid on existing directories must be done manually.

SUID / SGID Considerations

- S-bit set means that a program runs as the user (or group) it is owned by. If it is owned by root, it runs as root.
- Must be used carefully.
- Examine all of the programs using SGID and SUID and turn off the ones not needed by using the commands ***chmod u-s*** or ***chmod g-s***.

Restores from Disk or Tape

- Ideally the data administration and/or UNIX teams should be able to restore an environment from tapes stored offsite.
- Most IT groups can restore from disk.
- Examine system backup settings and schedules from the UNIX and data administration teams.
- It is easy to restore from disk. DBAs get paid to restore from tape.

Restores from Disk or Tape

- Typically, the UNIX team will process a dump file (*.dmp) and will archive this file to disk.
- Most IT groups then use software such as Legato or Veritas to convert the dump file into a format that enables archival from disk to tape.
- The tape should then be stored offsite.

Restores from Disk or Tape

- Typically for n-tier applications (application and data layers), IT will perform backups of the data separately from the application.
  - “Application” includes the OS and the application configuration.
- Auditor pitfalls
  - Not checking for actual evidence of a restore from tape.
  - Not checking for OS-compatible backup matched with backup of data.

Default Accounts

- At the UNIX prompt, type ***cat /etc/passwd***
- Check for accounts such as: oadmin, shutdown, poweroff, guest, demo, gast, Informix, oracle, ingres, sam_exec and sap.
- Default super user accounts: root, makefsys, mountfsys, umountfsys ,checkfsys, and admin or sysadmin.
- Default user accounts: lp, daemon, trouble, nuucp, uucp, rje, adm, sysadm, sync, bin, games, student
- Unless there is a business imperative for these IDs, they should be disabled.

Password Controls

- In order to simplify administration, most UNIX teams change root password and then do not implement a process for periodic change.
- Default parameters may be checked at:
  - ***/etc/default/password***
  - ***Recommended settings:***
    - PASSLENGTH= 8 [Minimum length of password]
    - MAXWEEKS=6 [Maximum time for usage of same password]
    - MINWEEKS=2 [Minimum duration for which password is valid]

/etc/passwd

- /etc/passwd file stores essential information, which is required during login
- User Name, Password, User ID, Group ID, User Description, Home Directory, Shell. Each field is separated by the ':' character.

/etc/passwd Guidelines

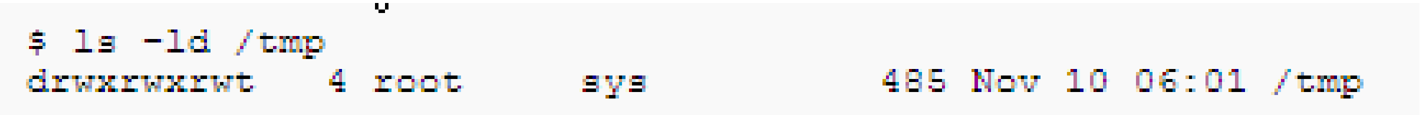
- No default user accounts or guest accounts present
- No unused accounts of users who have already left the organization are present
- No entries should contain the + sign, in either the */etc/passwd* or the */etc/shadow* files. “+” tells the login daemon to look to the NIS server for authentication if it is not in the /etc/passwd file.
- Only one line contains 0 in the third field: i.e. there is only one root user on the system
- There are no entries with :: after the username
- All system (non-user) accounts have ':*:' after the user ID and have '*/dev/false*' as their login shell.
- No '+' appears in any line in */etc/passwd*
- Password shadowing is implemented or is not available for this operating system.
- The administrator periodically audits the */etc/passwd* and */etc/shadow* files for additions, alterations, and removals.

Sensitive Files and Directories

- /etc/default/passwd root sys r-- r-- r--
  /etc/passwd root security rw- r-- r--
  /etc/security/audit_user root security rw- r-- r--
  /etc/security/shadow root security r-- --- ---
- /root root bin r-x------

Privileged Programs

- SUID and SGID
- Sticky Bit
  - Used to control access to files in unprotected directories. Denoted by “t”



```
$ ls -ld /tmp
drwxrwxrwt   4 root     sys            485 Nov 10 06:01 /tmp
```

  - When the sticky bit is set, only the item's owner, the directory's owner, or the superuser can rename or delete files.
  - Normally, any user with write and execute permissions for the directory can rename or delete contained files, regardless of owner.

Sendmail

- Suggestion – disable it
- Potential for a lot of embarrassment
  - Can pretend to be any sender and can send to anyone.
  - Spammers use sendmail
  - Test it out by typing:
    - echo “Please send money!” | mailx –r warren.buffet@berkshirehathaway.com -s “Request” kumar.setty@solocup.com
- Recommend PostFix if you want a secure alternative to Sendmail.

Network Security - Commands

- Disable all “r” commands unless specifically required:
  - rlogin
  - Rsh
- If “r” commands are required:
  - Filter ports used by “r” commands
  - Use secure versions of commands
  - Use tcp wrappers so that a small number of ports are exposed.
    - Used to filter network traffic on IP servers for UNIX-like OSs.

Network Security - Files

- **Inetd.conf**
  - Manages the Internet services
  - Typically resides in ***/etc***
  - In many companies, almost all of the services are commented out. Usually keep some internal services for time management and FTP.

Network Security - Files

- ***Hosts.equiv*** and ***.rhosts***
  - This file and a local user's $HOME/.rhosts file, identifies users on remote hosts who are permitted to remotely execute commands on this host.
  - Check the contents of this file to see if the entries are appropriate.

Patch Management

- Usually wait six months after a patch has been released.
- Some patches can be applied “hot” or while the server is running.
- Kernel patches usually require bringing the server down.
- Main concern is if users are accessing applications within network from a browser on the outside of the network.

Patch Management

- Check http://www.cert.org/ for latest exploits and patches.
- Ask for evidence of patch management procedures and for patch “baseline”.
  - Is IT keeping track of their current patched level?
  - Are they recording all of their issues with applied patches?
  - Are there any unique problems that have arisen due to applied patches?

Step 3 – Medium Risk Items

- Absence of host and network-based intrusion detection.
- System accounts using FTP
- Disabled accounts should have null shell
- Default umask setting
- Absence of documented policies and procedures
- Servers not using RAID

FTP

- Disable network services that use plaintext authentication:
  - FTP
  - Telnet
- Use certificates.

Disabled Accounts

- Check the /etc/passwd file and ensure that the right-most shell field is null (“dev/null”).
- ftpadmin:x:502:502::/home/ftp/./ftpadmin/:/dev/null
  - The /dev/null part disables their login as a regular user. With this modification, the user ftpadmin now has a fake shell instead of a real shell resulting in properly limited access on the system.

umask settings

- Only restricts permissions. Cannot grant them.
- Umask can be used to display the current umask setting for files and directories.
- Boolean format = AND NOT
- Represents a “subtraction” from 666 (files) and 777 (directories).
- If umask has a value of 022:
  - For files, 666 AND NOT 022 = <u>644</u> (rw-r--r--)
  - For directories, 777 AND NOT 022 = <u>755</u> (rwxr-xr-x)

RAID

- Redundant array of independent disks.
  - Is a cost, risk/reward, and/or architectural decision.
  - Can divide and replicate data among an array of disks.
  - Array is addressed by the OS and one disk.
  - Ask for evidence of any type of data redundancy.

Step 4 – Low Risk Items

- Backup tapes not being stored offsite
- Review members of /etc/group for appropriateness
- Banners
  - /etc/motd = “message of the day”

Lessons Learned

- In the absence of software, assessing UNIX security is time-consuming and very tedious.
- Leverage the audit steps to create a specification for a script that may be run on a periodic basis. This is a great incentive for the UNIX administration team.
- Security decisions should be made **before** putting a system into production.
- Make an effort to understand the overhead involved with monitoring servers, archiving audit data, and reviewing this data.

Role of IT Audit Community At Large

- IT auditors should bridge their knowledge of controls with the security design and architecture that has been disclosed by the software vendors.
- Currently there exists a huge gap in knowledge between software vendors and IT auditors. Software vendors will eventually bridge this gap.
- Does this mean we capitulate? NO. We need the IT audit community at large to continue their work so that they keep these vendors honest.
- Decode the risks and offer solutions to decision-makers – advisory role
- Have we made a lot of progress? NO
- Can we do better? ABSOLUTELY.

Collaborative Solution

- http://en.wikibooks.org/wiki/UNIX_Computing_Security/Auditing_Guidelines
- UNIX security is a vast topic. One simply cannot expect to cover everything within a couple hours. People have spent decades and have built entire careers on the topic of UNIX security.
- Collaborative solutions:
  - Share knowledge
  - Collaboratively build a “mental model” for evaluating UNIX security
  - Iterative and collaborative approach that shares nuances among vendors, configurations, and industries.

Why Collaborate?

- Adoption of a more ***standardized*** and ***uniform*** approach.
- Productivity
- Efficiency
- Consistency

Other Resources

- http://www.deter.com/unix/#unixcode
  - Good resource on open source security tools.
  - Latest UNIX hacks
  - Password crackers

Questions and Discussion