Lecture 9: Linux System Security

Lecturer: Prof. Zichen Xu

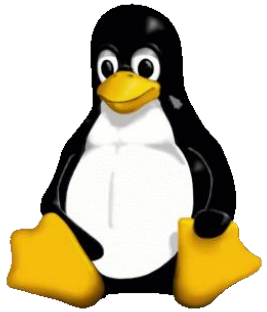
What is a Secure System?

- Secure system is an abstract concept

- Defined as “Robust”, it depends on what you need, how much time you are willing to put in, and what resources are at your disposal

P.C. vs. Server

- Close all services
- Don’t open accounts to everyone. Only to good and trusted people

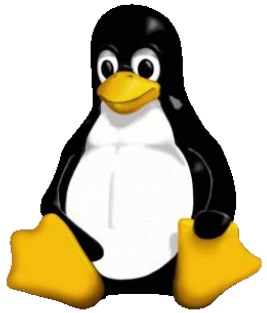


- Close as much services as possible
- Make sure users have good passwords - use crack-lib. Demand periodical password changes

P.C. vs. Server (cont.)

- Don’t install what you don’t know its origin
- Download only from known places (www.linux.org, etc.)
- Remove Suid’s if you are not the only user

- Don’t install what you don’t know its origin
- Download only from known places (www.linux.org, etc.)
- Remove as more Suid’s as possible

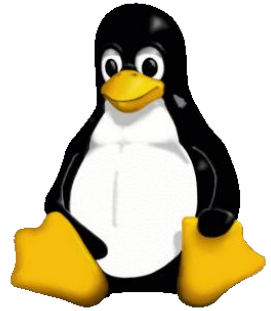


Securing Passwords

- Crack-lib them. Ensure passwords are not too short, and not too easy to crack
- Shadow them. Don't put them in */etc/passwd* but in */etc/shadow* (today's default in RH 6.1 installation)
- Connect to remote system using SSH and SCP (FTP over SSH channel) to prevent passwords from being sent as cleartext

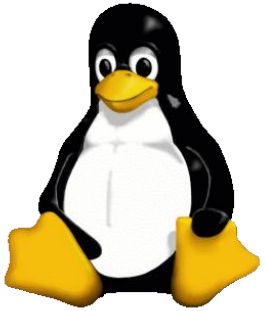
User and Group Security

- User accounts are created in */etc/passwd*
- Hashed passwords, password and account lockout policies are in */etc/shadow*
- Password and account lockout policies can be set during account creation, or with the *chage* command:
  - Minimum password age
  - Maximum password age
  - Expiry warning time
  - Inactive time after which account is locked out
  - Some future data when account will be locked out



Checks for these files

- No dormant or generic accounts present
- Accounts of separated users not present
- All system (non-user) accounts have /bin/false for the shell
- All system accounts have *NP* or *LK* in their password fields in /etc/shadow
- SOP exists for verifying validity of accounts in these files
- Every account in passwd has a corresponding entry in shadow
- Only one line contains 0 in the uid field in the passwd file



Password and Account Lockout

- Other stronger policies require use of PAM – Pluggable Authentication Modules
- PAM Allows the following to be set
  - Minimum password length
  - No dictionary words
  - No part of username in the password
  - Number of alphanumeric and punctuation characters to be present
- PAM is configured in the /etc/pam.d folder
- DEMO – change of password for user *auditor*

Password Strength Verification

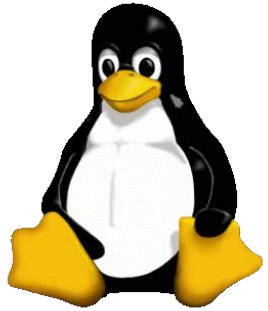
- Also known as Password Cracking
- Use ‘Crack’ from http://www.users.dircon.co.uk/~crypto/download/c50-faq.html
- Works on almost all Unix platforms, and is very fast
- Also viable password cracker is John the Ripper
- Set these tools running for a day or two and ferret out all weak passwords

Root Security

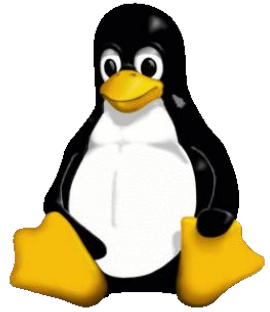
- No user must login directly as ‘root’
- Administrators must login with their own accounts, and then use ‘su’ to become root.
- This ensures accountability
- Viable alternative is the ‘sudo’ utility, which allows:
  - Listing of privileged accounts
  - Actions that can be taken by these accounts
  - Download from http://www.courtesan.com/sudo/intro.html
  - Time out of logged in user, so he has to re-authenticate in order to use ‘sudo’

Telnet and FTP vs. SSH

- Telnet and FTP are plain-text protocols
- Should be replaced by SSH
- Any inside user can sniff the traffic, even on switched networks with relative ease
- SSH uses encryption to provide services equivalent to Telnet and FTP
- Configuration is in */etc/sshd/sshd_config*
- SSH clients are available for free – putty for Windows



S vs. R

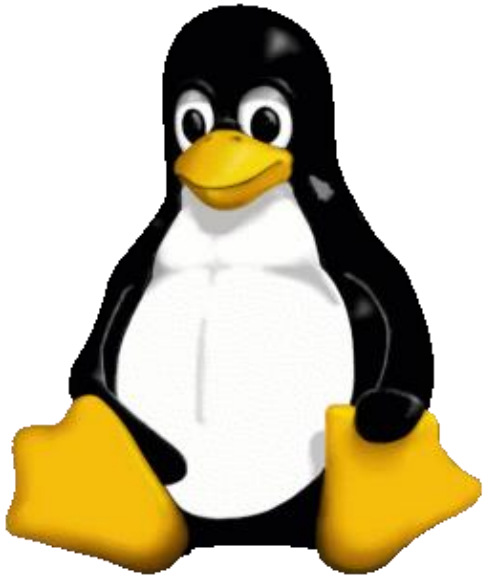


- SSH require password or a RSA phrase (SSH agent)
- SCP require password (no one will sent files without authorization)
- Several Authentication method are available

- RSH doesn’t require any password
- RCP - no passwords needed
- Work with Kreberos solely

S vs. R

- Use Compression

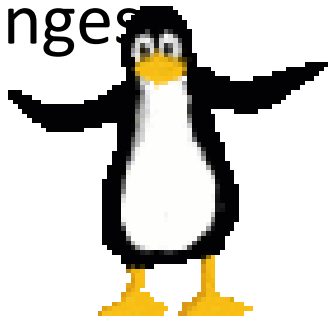


- Plain Connection

- Don’t require password at all - no password is moved, if one of the encryption functions has been broken - no one get the password!

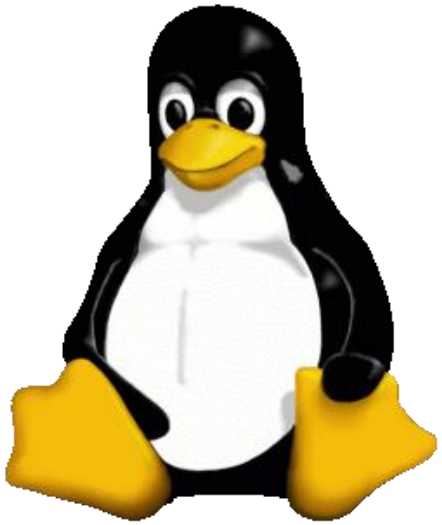
Authentication

- Prevents IP spoofing (claiming to be other IP then you are)
- Sometimes the algorithm allows also setting up a key for the rest of the session (Kreberos for example)
- Slow a little bit the connection (in the beginning)
- Known (and used) algorithms - Kreberos, RSA Challenges

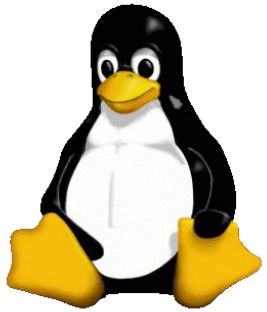


Dangerous Permissions

- Suid/Sgid - Check very carefully. Especially when the file is owned by root/wheel
- Write to all (xx2)
- Nouser/Nogroup
- .rhosts file (open R-services)
- Use "*find*" to find the files



Example - How to remove Suid's?

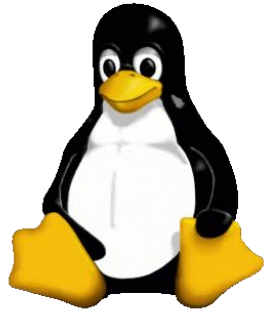


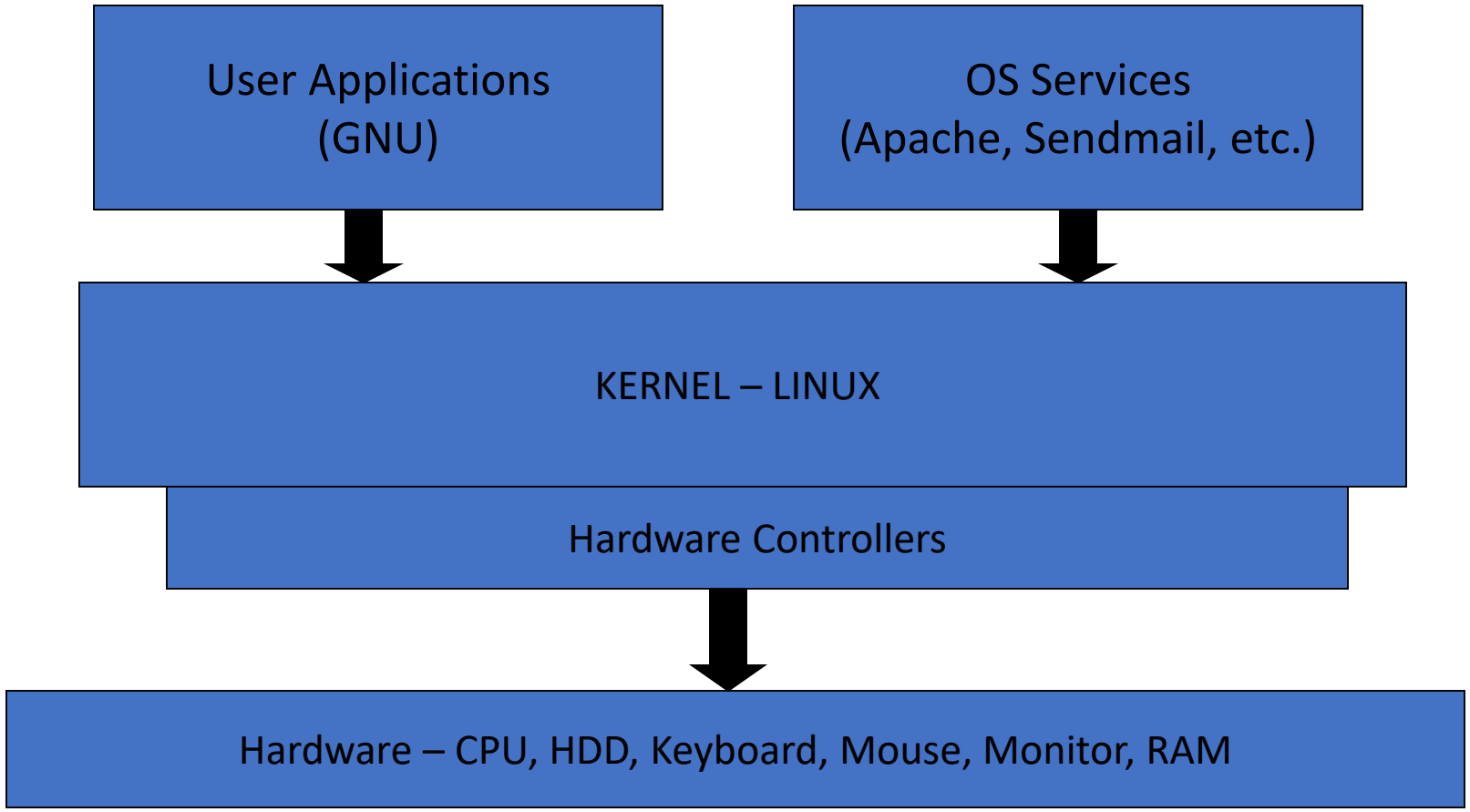
- First find them - *find -perm 4000 /*
- Then check if you need them - login, wanted deamons (Qmail, telnet, SSH, FTP)
- Close services not needed in the */etc/inetd.conf*
- Use TCP Wrappers to the rest of the ports (Those you usually get nuked - 139)

Linux Architecture

- **Linux Kernel** – the actual code that interfaces between user applications and hardware resources
- **Hardware controllers** – used by the kernel to interact with hardware
- **Operating System Services** – software other than the kernel that are considered part of the OS: X Windows system, command shell
- **User Applications** – software other than kernel and services: text editors, browsers, etc.

Diagrammatically (GNU-LINUX)



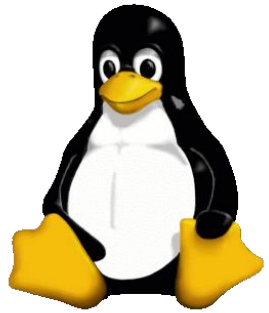


Key points about Linux Kernel

- It is separately distributed from user applications and other software
- Uses modules, which can be dynamically loaded
- For instance, support for FAT32 need not be fixed, but can be added dynamically
- Kernel can be completely recompiled and unnecessary components can be removed – unlike Windows
- Kernel has had buffer overflow vulnerabilities being discovered in it – very critical

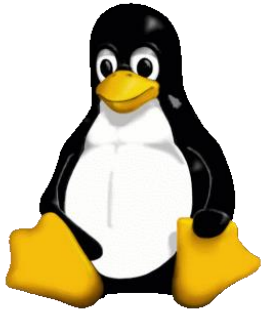
Kernel Security

- One of the most important ways to keep Linux secure is to ensure a patched kernel
- Check your kernel version
  - uname –a
- Third-party kernel patches for enhanced security:
  - Linux Intrusion Detection System – for ensuring integrity of critical files
  - Secure Linux Patch – prevent common buffer overflows, and simple security measures
  - International Kernel Patch – kernel-level strong encryption to be built-in

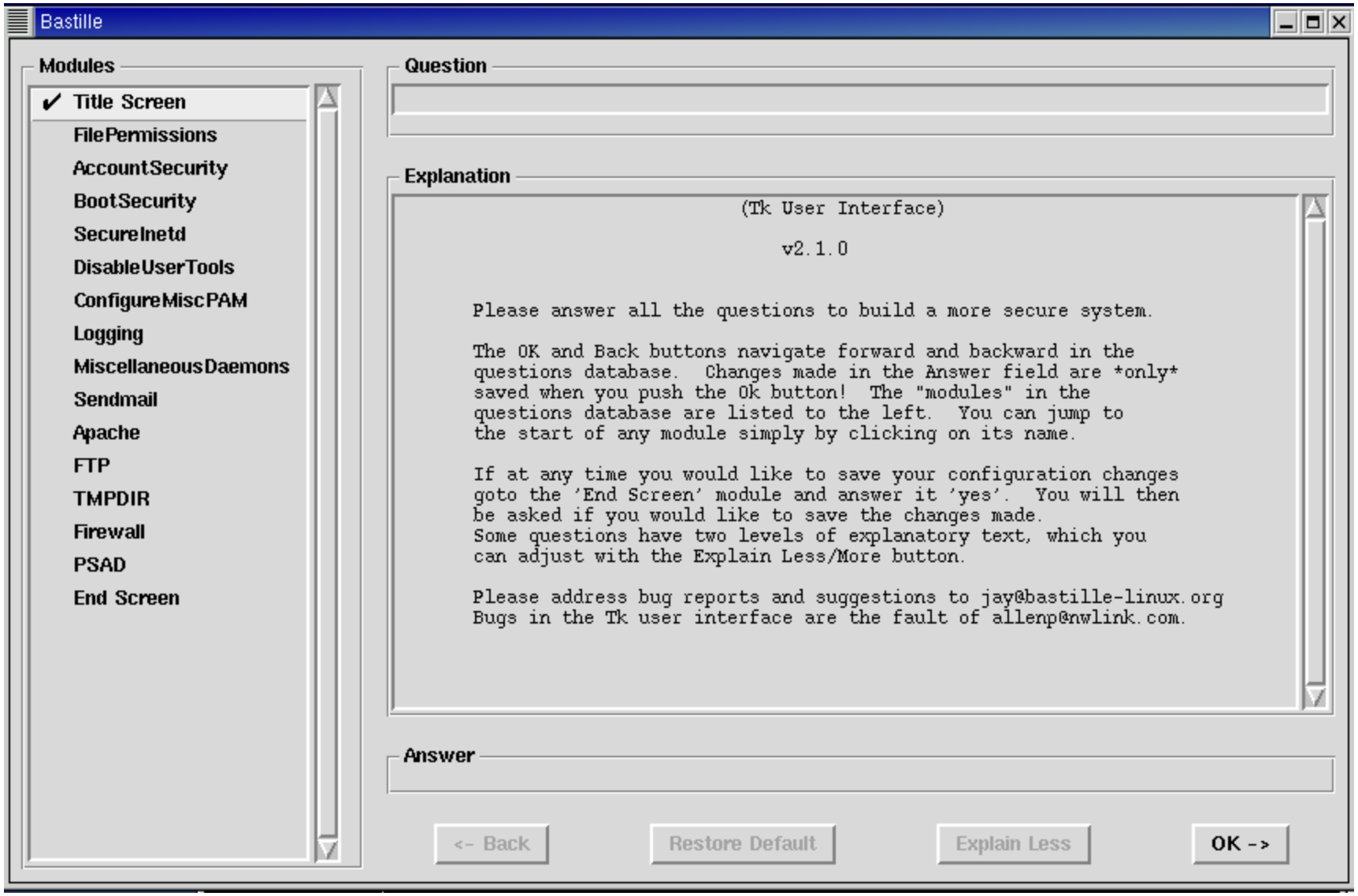


Click and run Security

- Bastille Linux
  - Available for popular Linux flavors
  - www.Bastille-linux.org
  - You'll also need Perl-Tk
  - Creates a set of security measures through a GUI
  - Most of the implemented changes can be undone
  - Must be first run on 'test' systems

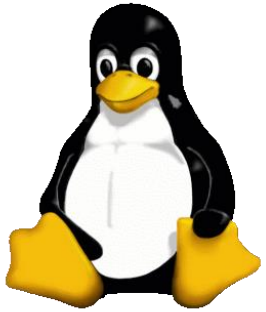


Bastille-Linux snapshot



Boot Security

- Boot configuration is decided by LILO (Linux Loader) or GRUB (Grand Unified Boot Loader)
- Check that only one OS is configured to load
- If required ensure there is an entry for password= in lilo.conf
- Also, ensure permissions are 600
- Demo

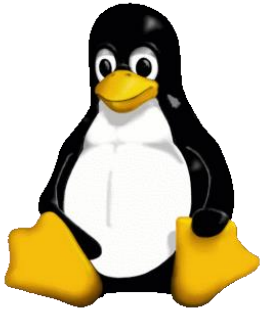


Operating System Security

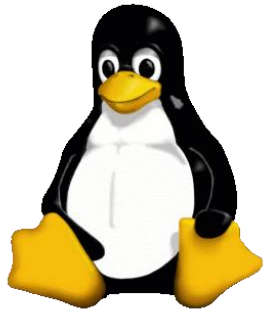
- Check processes
  - top –n 1 –b
  - ps- aux
- Check installed software
  - rpm –q –a
  - RPM = Red Hat Package Manager = installer packages for software on RH systems
  - Look out for unnecessary packages
  - Also ensure latest versions of packages are installed – especially those that are used by lower-privileged users: httpd, openssh, kernel, sendmail, etc.
  - rpm –q –a | grep kernel

Cron and At

- **Cron** is used to schedule regular jobs.
- **At** is used to schedule one time job in the future
- Both can be misused to install time-bombs on the system, which may suddenly cause the system to malfunction
- Can be restricted using files */etc/cron.allow, cron.deny*, *at.allow* and *at.deny*
  - DEMO
    - cron.allow contains root
    - cron.deny contains ALL



Linux Auditing



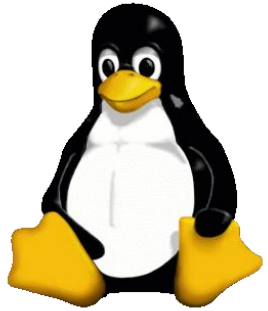
- Linux auditing is done using syslogd
- Configuration file is /etc/syslog.conf
- Format is:

Facility.Priority *Action to be taken*

- Facility – the application/program that is generating the logs
- Priority – Emerg, alert, crit, err, warning, notice, info, debug, none
- Action – send it to a file, send it to console, send it via email, send it to another system (loghost)
- Segregation of responsibilities – send logs to another system, where the security administrator has control

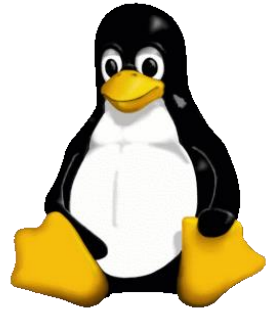
Linux Auditing – important commands

- Recent logins
  - last
- Last login time for all users (dormant users)
  - lastlog
- Last failed logins (requires to create /var/log/btmp file)
  - lastb
- Security related events
  - /var/log/secure
- Tools for Log Analysis
  - Swatch – real-time monitoring of logs
  - Logsentry
  - Logwatch



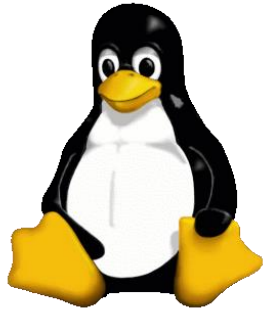
Tools for testing

- COPS
  - Computer Oracle and Password System
  - Outdated
  - Checks for common mis-configurations, weak passwords, insecure permissions, etc.
- TIGER
  - Similar to COPS, but more comprehensive
  - Also not recently updated
- TARA
  - Most updated and recent version of TIGER
  - Runs using shell scripts or preferably Perl



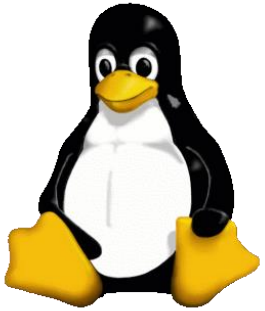
Network Security

- Services are started by */etc/rc.d* scripts and xinetd
  - chkconfig --list
  - chkconfig levels {numbers} {service} on|off
- Xinetd services are configured by individual files in */etc/xinetd.d/*
- Open network connections
  - netstat –antp
  - Use the –p option to see which processes are responsible for which open ports
  - Also lsof can be used



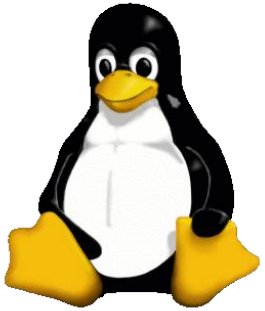
Network Services

- Possibly not required:
  - NFS and related services: autofs, nfs, nfsserver, nfslock
  - Unused networking services: routed, gated, ratvf, snmpd, named, dhcpd, dhclient, dhrelay, nscd, smb
  - Mail Services: Sendmail, postfix
  - Optional network and local services: atd, ldap, kudzu, rhnsd, ypbind, apache, quota, quotad, myself, etc.
  - Printing services: lpr, cups, lprng



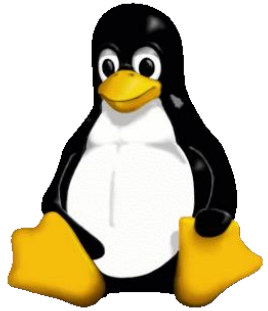
Xinetd

- Logic change from earlier inetd.conf file
- Builds in controls similar to TCPWrappers and more:
  - Access_control: which hosts are allowed to connect and at what times
  - Logging: which data gets logged
  - Resource utilization: limits on maximum connections supported, CPU usage, etc.
  - Others



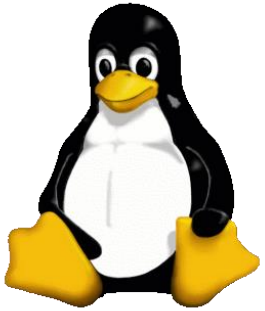
Trusted Hosts

- Entries in /etc/hosts.equiv and /etc/hosts.lpd are critical
- They allow users from those hosts to connect without supplying a password!
- Also, users can create .rhosts and .netrc files in their home directories, which function similarly. Find these as well



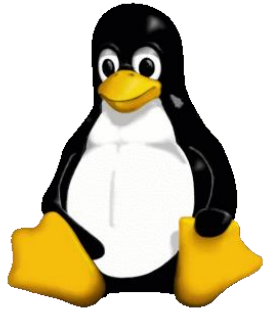
File System Security

- Unix Permissions are applicable to three entities:
  - Owner of the file (everything in Unix is a file)
  - Group owner of file
  - Everyone else
- Three main permissions apply, with numeric representations
  - Read = 4
  - Write = 2
  - Execute = 1



Unix Permissions

- Permissions are visible in the ls –l output:
  - Example
- First character identified type of file
  - D = directory
  - - = file
  - S = socket
  - L = link (shortcut)
  - P = pipe
- Next three identify **r**ead, **w**rite and e**x**ecute for owner, next three identify for group, and last three for everyone else



Unix Permissions

- These letters are added up:
- For instances:

  - rw- r-- r--

  It’s a file

  Owner can Read (4) and Write (2)

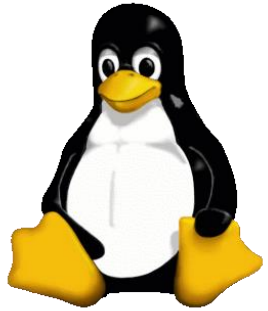
  Group can Read (4)

  Everyone else can Read (4)

  So permissions on this file are 644

  Conversely permissions, like 700 represent

  -rwx --- ---

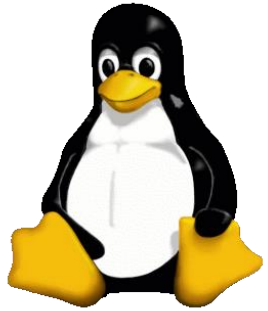


Other File Security Measures

- Permissions of a new files are determined by the value *umask*
- Advanced Windows-like Access Control Lists can also be created on Linux using the *linux-acl* package
- Disk usage can be periodically verified with the
  - df- k command
- SUID and SGID files are executables that can be executed by anyone, but they execute with privileges of owner (usually root) or group – very critical checks!

  find / -perm –4000

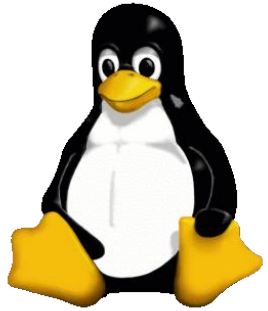
  find / -perm –2000



File Integrity

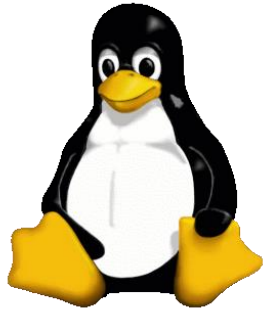
File Integrity can be verified:

- Size and timestamp – can be modified to fool the auditor
- MD5 hashes – secured method, but tedious
- File Integrity Software:
  - Must be used immediately after the installation
  - Create a database of MD5 hashes of all critical files
  - Monitor changes to these files and send alerts
  - Tripwire – commercial, scalable, central console
  - AIDE – open-source, reasonably enterprise-level



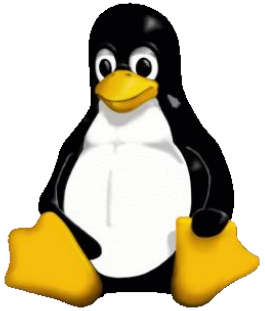
Application Security

- Linux systems can be used as
  - File Servers – Samba – Windows-compatible file server
  - Print Servers –lpd, cups, etc.
  - Mail Server – Sendmail (historically insecure), Qmail, Postfix
  - VPN Server – FreeS/WAN
  - Databases – PostgreSQL, MySQL (free), Oracle, Sybase, DB2 (commercial)
  - DNS Servers – BIND
  - LDAP Servers
  - Time Servers



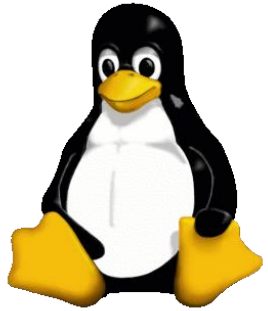
Application Security – Web Servers

- The Apache web server is an open-source, stable, robust and scalable solution with 64% market share
- Apache is usually configured to run with lower-privileged account ‘apache’ or ‘nobody’
- Installation location is referred to as $ServerRoot, and web site contents are located at $DocumentRoot
- Configuration file is at $ServerRoot/httpd.conf
- Configuration is done with the help of ‘Directives’



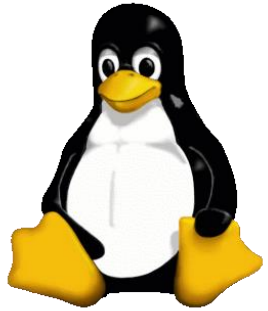
Important Directives

- Directory: access control based on source IP address or domain name for various files and folders of the website, using *Allow* and *Deny* keywords
- Also, within this directive, various options can be set. Recommended to set *Options None*
- Denial of Service and Buffer Overflow attacks can be prevented by *LimitRequest** and *Rlimit** directives
- CGI security is most important, to ensure scripts cannot be misused for compromising the server
- Apache uses various modules for added functionality. These must be reduced to a minimum
- Banner of Apache must be changed
- Apache must be run in ‘chroot’ environment



Linux Security Software

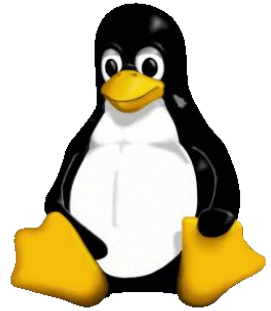
- Linux Firewall:
  - IPTables (new version of IPChains)
  - Scalable
  - Cost-effective
  - Robust
- Linux IDS
  - Snort
  - Scalable
  - Robust
  - Slight learning curve – Demo
- IPCop – Bootable CD version of firewall and IDS



Security Testing Software

- Nmap
  - Most popular security tool
  - Port scanner
  - Detects Operating System also
  - Can run in very stealth mode

- Nessus
  - Vulnerability Assessment software
  - Client-Server mode, server only in Unix
  - Uses Plugins for tests



References

- **The Unix Auditor's Practical Handbook** – K. K. Mookhey http://www.nii.co.in/research/papers.html
- **Practical Unix and Internet Security** – Simson Garfinkel and Gene Spafford
- **Linux Security Benchmark** -http://www.cisecurity.org/
- **Linux Security and Controls** – ISACA & K. K. Mookhey – to be available at ISACA bookstore in 2nd quarter

Conclusion

- We have started on System Security in Linux
- We have talked a lot on different security models
- We will start on Linux Programming on Network Programming next week, or will we?
- Reading Assignment: Chapter 9, 10 in your textbook