

Canaudit Perspective

February 2005
Volume 6, Issue 2

Topics of Interest:

- The use of Linux for critical IT business functions
- Linux Control and Security Concerns
- Cooperative audit approach
- Using existing IT Audit resources to conduct Linux Control and Security audits

Linux migration creates new challenges for IT Audit and Security: Here are some tips to help get you started



It's no secret that businesses are on the constant lookout for ways of gaining a competitive advantage over others in their industries. New methods for increasing efficiency and lowering operating costs while increasing customer satisfaction and product quality are always on the radar screens of strategic managers. With emerging technologies making the headlines almost every day, it is no wonder IT Departments are among the largest blips on the radar screen of cost reduction. Outside of outsourcing IT functions overseas, some businesses are, or already have been, exploring the possibilities of using Linux as a lower cost alternative to traditionally high-cost IT solutions. Many CIOs, if they haven't already, will likely have to answer inquiries from the corporate powers that be about the feasibility of implementing Linux as a lower-cost alternative to traditionally costly commercial IT solutions. Some of the more technology savvy CIOs have been researching possible Linux

migrations and implementations for years. Linux has become a major player in the business IT world and is steadily building momentum every day.

The increasing use of Linux for critical IT business functions creates new control and security challenges for IT Security and Audit to tackle. While the diversified nature of Linux is a leading attribute to its growing popularity among businesses, this diversification also has an inherent side effect of complicating the IT Audit and Security functions as well. There are many different distributions of Linux to choose from and just as many ways of using them. For example, Linux can be used to host a perimeter firewall, a network router, workstation functions, any type of server function, and the list goes on. So how does an auditor go about conducting a controls and security review of such a diversified operating system environment? Without a doubt, the best way to reduce the complication of controlling and securing Linux is to take the time up front to develop a sound implementation plan prior to placing Linux into production. As auditors and IT security professionals you certainly already knew this. It would be even more efficient to include an experienced IT auditor as part of the team assigned to the Linux implementation project right from the start. This way, the general control and security methodologies used every day by IT audit would be applied right from the get-go of project planning thus saving time and aggravation in after-the-fact remediation efforts. Early system implementation planning does little to help the vast majority of auditors and IT security professionals who have had the Linux bomb dropped on them by finding out Linux has recently been implemented by IT and they now are responsible for reporting on the controls and security of it. Oh, and, by the way, management needed the report yesterday. Well, not to worry, I am here to help.

Selecting the Best Internal IT Auditor for the Job

Among the first items to be addressed should be selecting the best auditor or audit team to meet the demands of a Linux based control and security review. Not all IT auditors can dive right into a control and security audit of an existing business Linux implementation and be successful. IT audit experience is a critical factor in selecting the best available auditor to conduct a Linux control and security review. Likewise, the chances of success will increase even more if the auditor has solid experience in UNIX auditing. This, of course, is assuming there are no IT auditors available with Linux audit experience to start with. If the resources are available, selecting a seasoned

internal IT auditor with experience in UNIX auditing is an ideal place to start. An experienced UNIX IT auditor should have a solid knowledge base which can easily be developed into Linux control and security knowledge as well. While, generally speaking, Linux and UNIX are similar in design and control features, there are some critical characteristics of Linux which are very different from UNIX and must be recognized and addressed accordingly. Most auditors with experience in UNIX will already be aware of the similarities and differences, or at least that there are differences, between UNIX and Linux. What if the internal audit resources are already overloaded and specific UNIX and/or Linux control and security skills are nonexistent within the audit department? Hiring an external consultant is always an option; but so is sending an internal IT auditor to training. The latter of which allows the company to keep the knowledge for use in subsequent audits and ensures an operational audit perspective.

Auditing General Linux OS Controls and Security

Determining the depth or scope of the Linux control and security audit is also critical and should obviously be conducted early and meet or exceed management requirements. If all that is needed is a general IT controls review, of which Linux is just a small portion of, then there is no point spending a lot of time trying to figure out detailed Linux control points such as kernel hardening procedures or file system mounting options for example. The detailed Linux OS control points should be assessed in a separate detailed Linux OS control and security audit. Conversely, if a detailed Linux OS control and security audit is required by management then General IT Controls and Security issues should only be assessed if they directly linked to the Linux environment password policy compliancy. For the purpose of this article I will take you down the middle of the road and detail a **General Linux OS Control and Security Review**; a hybrid audit of sorts which shares blurry scope boundaries with a **General IT Control and Security Audit** and a **Detailed Linux OS Control and Security Audit**. Now that I have confused you, let me try to clear things up:

General IT Control and Security Audit:	A high level audit focusing generic system controls such as computer operations, security, support, policy and procedure designed to ensure a stable business computing environment
General Linux OS Control and Security Review:	A hybrid review of general OS controls and security as they specifically relate to the Linux OS
Detailed Linux OS Control and Security Audit:	An in-depth “nuts and bolts” operating system audit focusing on access controls, data integrity/security, file permissions and local/remote threat reduction designed to ensure a stable and secure platform for supporting business applications and processes

Depending on your interpretation of the scope definition for a general Linux control and security audit there are still some Linux specifics that you may need to be familiar with in order to meet the audit objectives. For example, what work paper references are needed to identify the accounts that have root access? Keep in mind, the needs of management should ultimately define the scope boundaries or depth of the audit which is why I refer to the scope boundaries as blurry. Chances are that my interpretation of general Linux controls and security audit is a bit more detailed than most IT auditors, but then I am a technical auditor who specializes in penetration testing and vulnerability assessments. I have never done, nor am I qualified to conduct financial audits which from my understanding have clear lines between general, detailed and key control reviews. Some may even question how I can specifically include the Linux OS in terms of a “general” IT control and security review. As I mentioned before, it is all dependent on the expectations of management.

The control points which should be assessed as part of a General Linux OS Control and Security review are listed below. This list does not include the high level general IT controls, application controls or the detailed Linux

control and security points. It is simply a list of general control and security audit points requiring specific information or files from the Linux OS.

General Linux OS Control and Security Points:

- Account Controls
- Password Controls
- Physical Controls
- root access controls
- Remote Access Services
- Backups and Restoration
- Data Integrity
- Patch Management, Versioning and Change Control
- Logs and Monitoring

As with any IT control and security audit it is important to know where to find the source information for each control and security point being assessed. In some cases a single control or security point may have several sources files to review. Likewise, a single issue may be adequately controlled by overlapping control layers of which Linux has many.

Account & Password Controls: When auditing Linux based account and password controls there are several primary control points that must be checked.

Passwords:

-Is the password file shadowed?

All Linux password files should be shadowed. By shadowing the password file you are taking the encrypted password hash out of the **/etc/passwd** file and storing it in a secure shadow file that can only be read by the root account. To check if this control has been implemented simply look for the **/etc/shadow** file and ensure the permissions are restricted to root only. Note: Some distributions may store the shadow file in different locations. The main concern here is to ensure the encrypted password hash is not stored in the **/etc/passwd** file. So for the purpose of this control you could also just check the **/etc/passwd** file to ensure it does not contain encrypted password hashes.

-Do all local accounts have passwords?

There are several ways to determine if every account has a password. To me the best way is to launch a single command on every Linux machine being audited which will output a list of accounts with blank passwords (note: this command must be run as root):

```
awk -F: '($2 == "") { print $1 }' /etc/shadow
```

-Do all passwords meet the password policy? Are exceptions noted and authorized?

One of the best ways to determine if a Linux machine has weak passwords is to crack them for a period of time. John the Ripper distributed by the Openwall Project is a free and simple-to-use password cracker that can be used to find weak Linux and Unix passwords. John the Ripper can also be configured to only crack passwords that do not match your company password policy or from a dictionary. PAM or (Pluggable Authentication Module) is useful for enforcing password policy in Linux and is highly recommended for use with local system account authentication.

I recommend the following minimum password policy for local user accounts:

- Passwords expire every 60 days or less
- Passwords contain alpha, numeric and two special characters
- Passwords have a minimum length of 8 characters
- Password history of 12
- Minimum password age of 2 days

Most of these settings can be found in the **/etc/login.def** file on a Linux machine. However, password complexity is likely only being enforced if PAM or similar application is being used. To see what the current password requirement settings are if using PAM, view the **/etc/pam.d/system-auth** file.

Accounts:

-Are all existing accounts required?

The only way to determine if all existing accounts are required is to read the **/etc/passwd** file and determine one by one if every account is required. Be careful when making this determination with services accounts. If the service is not being used on the machine then chances are it can be safely removed, i.e., uucp and games. Be sure to exercise proper change control procedures before removing system accounts. Ensure each account can be tied to a user who requires access. Be sure to compare your list to a current employee list from HR to ensure no accounts from ex-employees remain active. Also be sure that each account is required. I have often found instances where local accounts are created for employees who didn't even know they had an account on the machine.

-Do system accounts have login shells?

Not all accounts require shell access to the operating system. To determine if an account has shell access review the **/etc/passwd** file and look at each account. Usually the last colon-delimited field in each line specifies if the account has a shell or not. System accounts do not usually need shell access to the machine and should have **/dev/null** specified in the shell field. Accounts requiring a shell should have the path to the shell specified, i.e., **/bin/bash** or **/bin/csh**. Removing system account shells reduces the risk of the OS being compromised as a result of default or simple passwords commonly associated with system accounts.

Groups:

-Are groups used to controls account access?

For this level of control and security review the **/etc/group** file should be reviewed to determine if accounts are in the proper groups. In most cases there should not be any accounts in the root group.

Physical Security: Physical security includes a broad range of control points for most organizations. For the purpose of this guide we are focused on physical security in direct relation to the Linux Operating System.

-What is the Bios Boot Order for the machine?

Typically it is best to have the BIOS boot order set to the hard drive first then either the floppy (if there is one) or CDROM. This prevents an attacker with physical access to the machine from using a bootdisk to load another operating system such as those that run in memory.

-Has a Bios password been set?

A Bios password would prevent an unauthorized user with physical access to the machine from rebooting it.

-Is the boot loader password protected from having runtime commands passed to it?

Boot loaders, depending on the one being used, typically allow users at the console to pass runtime commands to it during the boot process. The concern here is that an unauthorized attacker with physical access to machine could reboot the machine into

single-user mode with no authentication. Single-user mode allows the console user to have root access to the OS with no password. This is useful from a trouble shooting perspective but dangerous from a security perspective. By implementing a boot loader password single-user mode will require a password.

Root Access: Access to the root account can be gained and controlled in many different ways. Only administrators who will be making authorized changes to the system should have access to the root login process.

Su and sudo allow accounts to become root or execute specific commands as root respectively. Su and sudo are very useful programs to help administer Linux and Unix based operating systems. If implemented properly they can also be controlled securely. The problem is that it is very easy to make a configuration mistake with su and sudo that could allow unauthorized access to the root account.

- Su and sudo: Who has been trying to become root?

Su logs are one of the best places to look on a Linux system for accounts actively using su to become root. The problem is that logging must be enabled in order to be able to review the logs. Eight out of ten times the level of logging is not sufficient enough to use if for an audit. There is also the possibility of the logs being edited. The best bet here is to sit down with the administrator and ask them to provide the su logs as they can be stored anywhere and under any name even on a remote syslog server. On my Bastille-hardened Red Hat machine the su logs were contained in the /var/log/loginlog.X files (x representing the number of closed log files). You must then determine if the accounts that are using su to become root are in fact authorized to become root. I also recommend following up with some of the su root account holders to find out if they had actually been using su or if perhaps their account had been compromised and someone else was suing to root. While reviewing the logs it is also a good idea to take notice of any excessive su failures as they could be indicative of an attack.

-What accounts have access to su?

Check the permissions of the su binary. On most hardened Linux systems the permissions of the su program have been restricted to a group. In such cases you will also have to view the /etc/group file to determine the accounts which are able to use su. The su program is usually located in /bin.

-Who is using sudo and what are they using it for?

To determine the accounts which are using sudo to execute commands we must once again resort to the logs. Hopefully sudo logging is enabled. In most cases execution of the sudo command will be logged in the /var/log/secure.X (X representing the number of closed log files).

-What accounts are authorized to use sudo?

Again, as with su, check the permissions of the sudo program, usually located in /usr/bin and match up with the group file if necessary. Additionally, check the /etc/sudoers file for details on the accounts which can use sudo and the restrictions they have on them.

-Logging in as root remotely?

Logging in as root from a remote machine is not a good idea from a security perspective. The best way to see if remote root logins are authorized is to actually try it. To support your finding it is also a good idea to check the /etc/securetty file. If using SSH, check the /etc/ssh/sshd_config file to ensure root is not authorized to authenticate directly. If using PAM check remote root authentication for all PAM aware services as well.

Remote Access Services

Control of remote access services is a critical component of any control and security review. In Linux there are many ways to control services starting with disabling the service, controlling host access, controlling user access and setting restrictions on each of them. This is a high-level assessment of some of the high-level access control points that should be reviewed as part of a Linux general control and security review.

-What services are currently running?

Start with a quick port scan using any port scanner you like. I prefer nmap. We are not looking for every service that is running, just the more prominent ones. Simple services or plain text services such as telnet, ftp and rlogin, rexec and rsh should not be used. They are antiquated and vulnerable to network sniffing attacks and session hijacking.

-TCP Wrappers

TCP Wrappers is a host-based control that can be used to allow and deny specific hosts to specific services. If being used, be sure to review the /etc/hosts.deny and /etc/hosts.allow files. The man pages are very useful in understanding how TCP Wrappers works. Two important points to note are: TCP Wrappers will stop checking the connection at the first applicable rule and that the rules are read in order, starting with the hosts.allow file then the hosts.deny file. All connections that do not meet any rules in either file will be accepted.

-Are there any system trusts?

System trusts can be used in several different ways to compromise a machine and I recommend they not be used. Check for a .rhosts file in each user home directory as well as the /root directory and the / directory. If there is a .rhosts file check the contents and ensure they are empty. I recommend placing an empty .rhosts file in each of the above mentioned directories and changing the permissions so that only root can read them. This will help prevent the creation of a .rhosts file in being used to compromise the machine.

Additionally check for a /etc/hosts.equiv file and if it exists check the contents. Hosts.equiv files are also used to set up host-based trust relationships and should not be used. I recommend the permissions of the hosts.equiv file be set to root read only and the contents of the file be blank.

-What services are set to start automatically? (Run Level Services)

Depending on the distribution of Linux being tested there are several files to check. If using Red Hat try the /sbin/chkconfig -- list command. It should give you a list of both Xinetd services and other services that are scheduled to start during boot based on run levels. Usually run levels 3 – 5 are those that are executed during boot. The others are reserved for other uses. Be sure test any changes prior to making changes in production.

To see what services are in a LISTEN state, execute the command **netstat -lp** as root.

-What services are set to run in xinetd.d?

Also referred to as a super service, xinetd controls some, not all, services and should be checked. Not all Linux machines will be using xinetd depending on the distribution of Linux and the purpose or function of the implementation. Check the /etc/xinetd.d directory. You should check each file within this directory to see if the service is enabled, and determine what restrictions if any are placed on the service. Xinetd is usually run in conjunction with TCP Wrappers so the services started through xinetd should be wrapped, but be sure to verify this.

-What services are set to start via inetd?

Some distributions of Linux use inetd to start remote access services. Review the /etc/inetd.conf file and look for all the services which do not have a # at the beginning of the line. If TCPWrappers is being used to control access to inetd services be sure that the TCP Wrappers program (tcpd) is in place of the first instance of the service program path. It should look something like this for the ssh service:

```
ssh stream tcp  nowait root  /usr/sbin/tcpd  /usr/sbin/sshd -i
```

-Determine what services the host-based firewall is blocking / allowing?

Work with the system administrator when determining the host-based firewall configuration settings, assuming there is one. IPchains and IPtables are commonly used with Linux and generally come with the distribution depending on the distribution being used. There are many host-based firewalls that can be used so I will not go into detail on any one in particular. The main rule to remember when reviewing the host-based firewall rules is to deny everything that is not explicitly required. Make a list of what is needed to support the business process and make sure the firewall blocks everything else. I know it sounds easy, but it can get complicated very quickly.

Backups and Restoration

Basically the only questions that should be answered at this level of audit regarding backup and restoration are the following: Are the backups working? Can we restore the system? Are the backups stored securely? Policy and procedures for backups and restoration are dependent on the system being used to conduct the whole process and are generally left to the General Controls Audit. That said, the best way to check if the backups and restoration procedures work is to test it.

Data Integrity

Data Integrity checkers are applications that prevent and can alert the administrator upon unauthorized changes to important system files. Tripwire offers both a freeware and commercial version both of which can be used on Linux. It is generally a good idea from a security standpoint to ensure your Linux systems are using some form of data integrity checker. The baseline should be taken prior to placing the machine on the network in order to get a clean image to work from. AIDE (Advanced Intrusion Detection Environment) is another freeware integrity application which is intended to be even more robust than Tripwire. The moral of this story is to ensure there is an integrity checker being used and to check the results of the regular scans that it performs.

Change Control, Patch Management/Versioning

Change Control, Patch management and versioning are all tied together and share one common goal among them: to ensure the rapid, safe and secure maintenance of software changes and configuration settings in a production setting. These areas have been a hot topic in the information security field for some time. A system for maintaining vendor patches should be established and implemented by now. Take a look at the Linux kernel version using `uname -a` as well as some other applications and services on the system. Then go to the vendor website and determine if the version is recent or not. Check the Internet to see what the vulnerabilities are with the version you have and determine if they are major threats or minor threats. In some cases it is not always necessary to stay up to date with the most recent version as is the case with the Linux kernel. However, exceptions should be documented and approved along with a threat analysis. Also it is important to ensure all patches and configuration settings go through proper change management for obvious reasons.

Logging and Monitoring

Logs are critical to information security and should be addressed as part of any OS-specific control and security review. Logs are useful in determining the source of problems as well as assisting in the detection and forensics of an unauthorized use incident.

-Are appropriate events being logged?

There is such a thing as logging too much information and the results are similar to those inherent in not logging enough information. Important details will be missed. Ensure syslogd is started automatically during boot by checking the runtime services discussed earlier. To identify what daemon processes are being logged review the contents of the /etc/syslog.conf file. This file is the configuration file for syslog (the Linux system logger). Under normal configurations the logs generated by syslog are stored in /var/log but other destinations can also be specified.

-Are the logs being monitored?

Even the most detailed and important logs are useless if they are not monitored. The duty of reviewing raw logs in a production environment is not fun and cannot realistically be done manually on an ongoing basis. Ensure the use of a log monitoring system or alert generating system that will provide notification of significant events such as account lockouts, failed logins, integrity checker alerts and whatever else is deemed important in your respective environments.

-Are the logs detailed enough?

When you don't need them, logs are nothing more than a resource hog. But when you need to determine the extent of a breach or conduct forensics they are invaluable. The detail of the logs is as important as the log itself. Take samples of each log and determine if the log is detailed enough to provide you with the information needed. Review the logs in the /var/log directory (or wherever your authentication based logs are stored) to make this determination.

-Are the logs being rotated and stored appropriately?

The logrotate daemon controls the rotation of log files. Review the configuration of logrotate in the /etc/logrotate.conf file. Individual service logrotate rules can be specified in the /etc/logrotate.d directory so be sure to check both. Logs should also be backed up and stored for a specified period of time on another machine typically a syslog server. If an attacker does manage to become root, one of the first things they will do is cover their tracks by removing log entries. Eyebrows should be raised if, when reviewing log files, you notice a long period of time, relative to the standard log cycle, with no entries.

Taking the Linux OS Control and Security Audit One Step Further:

Even if a detailed Linux operating system control and security audit is required by management, it still may be possible to fulfill the requirements internally by first following the guidance I provided for a General Linux OS Control and Security review and then taking it one step further. While the general Linux OS controls are being addressed, the IT auditor assigned to the project should also be developing his or her Linux control and security knowledge. A senior IT auditor is a master of multitasking so this should not be too much of a stretch in terms of realistic time management, especially if a **cooperative audit approach** is used by the auditor. Using the cooperative audit approach, the auditor creates a series of forms and checklists and sends them to the client for completion. The client completes the checklists, assembles the documentation and sends it to the auditor for review. This also provides the client with the opportunity to correct any minor lapses or weaknesses and to document their own concerns which may otherwise go unnoticed. The auditor reviews the material and then schedules an appointment with the client to perform supplemental work and follow-up interviews. In this case the checklist would contain the general information system control and security checklists only. While the general

OS control and security checklists are being completed, the auditor should be acquiring Linux-specific control and security knowledge needed to build the Linux Audit program and develop comprehensive Linux control and security checklists.

Informational resources about Linux control and security are plentiful and can be accessed on the Internet (for free), in books, whitepapers and from taking courses and seminars. While some resources are better than others, nothing beats good old hands-on Linux experience. I recommend setting up a mini-test network (completely detached from the production network) to implement and test different Linux controls and distributions and determine how they affect the supported business process. This will allow the auditor to roll up her sleeves in a test environment and learn the workings of Linux. This type of hands-on experience is critical to any technical audit. The experience will be reflected during the follow-up audit interviews with the client, in terms of auditor confidence and the ability to rationally discuss specific technical topics. In most cases the client will appreciate the fact that the auditor has taken the time to know what he or she is auditing. If the auditor doesn't have a decent understanding of the technology they are auditing the clients may feel the whole IT audit process is a joke and a waste of their time. The concern here is the client may avoid disclosing control issues which are not explicitly covered in the audit plan but are still critical issues. In severe cases the client may even attempt to corrupt or discredit the audit altogether.

All this might seem like more trouble than it is worth and fill you with a sudden urge to just outsource the whole darned thing. For some, outsourcing Linux Control and Security reviews may be the best choice. But before committing to outsourcing, take a look at what support and resources are already available within your audit team and assess what additional resources, if any, are needed to audit Linux controls and security. Chances are you'll discover that with a little additional training and research you already have what is needed.

As always, the comments in this article are mine and mine alone. I invite you to send your comments to me at (chad@canaudit.com).

Chad Parks
Manager Technical Audit & Security Services