Using LinuxAuditor to automate auditing of RedHat, SuSE, and Mandrake Linux systems.

GIAC System and Network Auditor (GSNA) Practical Assignment v2.1. Assignment Option 2 Topics in Auditing

William Schroeder
October 2003

This paper and accompanying application describe LinuxAuditor and demonstrates its usage on a RedHat AS 2.1 Server. The LinuxAuditor script was written for my use at work as an Information Security Analyst to provide a method of automating as much of a Linux audit as possible. Therefore, it may not include or do everything that the general population may want, but it works well enough for me and makes management happy.
The script, accompanying database, demonstration, and documentation are written for the Option Two – Topics in Auditing portion of the SANS Network and Auditing certification.

Introduction

As part of my day-to-day work in information security I am asked to "have a look" at new servers as they are introduced from development and QA into our pre-production environment. What this means is that the responsible manager wants a baseline of the new box as it is currently configured, and recommendations for changes to the operating system to increase the level of security. I did this manually for a while which meant redirecting output from commands like netstat into a file then massaging it in VI for eventual import into a spreadsheet. As the number of request increased, this became an incredibly time consuming task.
So, I set out to find utilities for automating my audit process. I found a large number of tools exist that assist the auditor's work on Microsoft Windows platforms and older UNIX systems such as Solaris. There are far less unified tools for the auditor to examine "hardness" and create a baseline for a Linux installation. The Center for Internet Security Linux Benchmark is an excellent tool for assessing RedHat and Mandrake Linux (other flavors are currently TBA). The disadvantage of the benchmark tool is its current limitation to two distributions and that it is not setup to be run from a centrally managed location. Also, the large task of collecting, organizing, and reporting remains to be done after the script run is complete.
Bastille Linux is a set of RPM packages that when installed actually makes physical changes to a machine to increase the overall security. While this is desirable, some clients will not want an auditor to make permanent changes to production machines. Also, Bastille Linux does not provide a client with a baseline of the machine to use for future reference or assist the auditor in generating a report to management on the status of the machine.
Two new utilities that were introduced during the writing of this paper were also investigated. These are "TIGER" and "lsat". They both audit Linux and Solaris but they require installation on the machine to be audited. TIGER was fairly thorough but the output still required me to parse and create a report for management. I was not able to test lsat at it would not compile on any machine at my disposal. Also, none of the existing tools I found can be centrally managed without some effort.

Based on my experiences auditing Linux servers at work, I developed a set of requirements for a utility to audit and baseline a server based on requirements set forth by management. They are fivefold:

1. The utility must be able to record data from large numbers of machines at the same time.
2. The utility must be able to store the collected information and later allow it to be manipulated into a coherent report.
3. The utility must not require the installation of anything on the audited machine.

4. The utility must allow for the retention of the records for comparison with later audits and for incident response handling.
5. Reduce the turn around time from audit/baseline request to turnover of the final report. Not finding anything that filled all of the above requirements I wrote a utility- LinuxAuditor. This tool fills in the gap between Bastille Linux that makes changes to a system and CIS's BenchMark tool.

LinuxAuditor when run in conjunction with an external vulnerability scanner like Nessus or ISS Scanner and Nmap will yield a complete and searchable status of a Linux machine.

About the Tool

The LinuxAuditor script is written in Perl, which is standard on all UNIX/Linux distributions. There are two components to LinuxAuditor, a client and a server. The client is the component that actually performs the audit on the machine. The requirements for the client (box to be audited) are simply Perl and the ability to connect over the network to the server. The second component of LinuxAuditor is the server. The server lives on the auditor's computer and listens on a socket for a request from a client. When it receives a request, a child is forked and the stream is parsed and inserted into a database. The server needs to have Perl, a network connection, and MySQL. A display manager for report generation requires Apache and Embedded Perl. The installation of these packages is discussed later in this paper.

In a nutshell, to use LinuxAuditor, edit Connector.pm to add the IP address of the server and an appropriate network port. Then, move LinuxAuditor.pl and Connector.pm to the box that will be audited via SCP. On the server, make sure that the peer address is set and that AuditServer.pl is started. While logged in as root on the client, start LinuxAuditor.pl. Root access is required because the application needs to run utilities like *lsof*. On machines where I do not have root access, I pre-arranged time with one of the system administrators to work with me. The script works by making a socket connection on the port determined in Connector.pm back to the server and writes all audit results over the socket to the server which then parses the data and inserts the records into a MySQL database. I additionally log a text file for each host audited into /tmp of the audit server. This report is useful as a permanent record of audit activity and augments the information in the database. I then use the display manager to display individual captured findings as screen shots.

The display tool includes a report summary that contains a table of the audit findings, comments on each item, and spaces for comments on audit findings. The final report is then printed out on 8.5 x 14 paper and any comments for the clients are filled in.

Auditing Linux- What to audit? "Defining The Checklist"

Linux is a recent addition to our organization and so there are no corporate policies regarding Linux installations. I was asked to perform a number of audits on Linux servers to determine a baseline for future Kickstart installations and to provide guidelines to define what the new policy should be. I began by meeting with the manager responsible for Linux and discussing what the risks to the system are. I also began independently researching what are the current best practice standards for auditing Linux.

The risks to Linux systems in our environment are minimized by placement in the network architecture. All systems are located on an internal network accessible only by production networks and one administrative vlan. Multiple firewalls and strict ACLs on routers prevent external illegal access. There is some risk related to Oracle configuration and vulnerabilities, but that does not fall into the scope of this audit. Based on my meetings with management, the risks to Linux in systems in our organization will arise primarily from:

ID.	***Description***	***Threat***	***Vulnerability***	***Risk Level / Metric***
1	Incorrect user permissions on important files.	Moderate	Weak file permissions may allow access to sensitive files such as application configuration scripts that contain passwords.	Moderate. Only a limited number of user ID's have access to production machines.
2	Extraneous services such as web servers or SNMP.	High	For each unnecessary service running, the level of risk increases.	High. An exploitable ftp server may allow a malicious user root access.
3	NFS mounts from users without permission.	Moderate	NFS mounted file systems may allow users unauthorized access to data.	Moderate. A certain degree of privilege is required to mount file systems.
4	Additional users with UID zero.	Moderate	Unauthorized root privileges.	High. UID zero yields total control of the machine.
5	Accounts without passwords left after setup.	High	Unauthorized access to files and services.	High. Unauthorized remote system access and possible compromise.
6	Weak access controls.	Moderate	Unauthorized users or possibly compromised machines may assess a server with weak access controls in place.	Moderate. An attack of this would have had to already compromise strict ACLs.

ID.	Description	Threat	Vulnerability	Risk Level / Metric
7	Network interface cards assigned inappropriate IP addresses.	High	Misdirected network traffic may cause application stability issues.	Moderate. A limited number of users have the access level necessary to plumb interfaces.
8	Extraneous SUID, SGID, World Writable, and files without user or group association.	Moderate	An SUID root application with a buffer overflow may allow root access or a world writable file may contain sensitive data.	Moderate. Exploits are are commonly posted on the Internet.
9	No post install updates applied or regular patch maintainance.	Moderate	Security and stability issues arise from not patching.	Moderate. Exploits are freely available on the Internet.
10	Password and user name sniffing from unsecured protocols.	High	Sniffing may reveal username and password information that allows unauthorized access.	Moderate. Network controls are in place to detect such sniffing.

As part of my research into current best practice for auditing Linux, I found numerous web sites that address the subject of auditing Linux. The three best sites that I found that discuss current practice are.

1. The SANS site at www.giac.org where papers on auditing are published and available for download.
2. LinuxSecurity.com. In addition to papers on Linux security, they have an excellent reference card from which I absorbed many points for my checklist.
3. www.cisecurity.com provided me with numerous pieces of information to include in my checklist and corresponding discussion. They also have a tool for making the changes that are suggested in the CIS checklist. I am not allowed as part of the audit to make any changes to a server. This limitation prevents me from running LinuxBenchmark on production systems although the checklist has proved to be invaluable.

Based upon my research and management requirements, I have created a checklist for the LinuxAuditor tool shown in the following table. This checklist covers the points required by management and current best practice for Linux auditing.

No.	**Checklist Item**	**Discussion**	**Type**	**Risk**	**Compliance**	**Source**
1	Install performed from verified media	To reduce the risk of installing trojaned/corrupt software only install from verified media. Can you trust the copy of the copy that was obtained from a friend?	Obj.	Due to the sensitive nature of data on the production networks a backdoor for a hacker from compromised software could have a significant financial impact.	Software should be installed from vendor/OEM source.	6
2	Post install OS update performed from verified media or source.	There are always OS, Kernel, and application updates after a new OS release so ensure that a box is patched before it is released into production.	Obj.	Patches that have been compromised may contain trojaned software or key loggers. Non-verified updates introduce the risk of remote compromise.	Software patches/updates should be verified and installed from vendor source.	4,6
3	OS updates are scheduled on a regular basis.	Patches come out all the time. What mechanism is used to ensure that patches are applied on a regular basis.	Obj.	Without updated software there is risk from instability issues that could result in less than %99.99 uptime that corporate IT policies dictate, as well as financial impact from data loss.	Updates and methods should be set and scheduled.	4,6
4	Install only necessary services per function of server.	Determine the role the box will perform to determine the services that will be installed.	Obj.	The number of installed packages and running services are tied to the amount of risk. If the number of packages and services is reduced to the essential minimum, risk is also reduced.	Server install build should be determined by role.	4,6
5	TCP Wrappers	TCP wrappers is a reliable way to restrict and log access to TCP daemons. TCP wrappers utilizes a simple text file for access control and does not require a reboot.	Obj.	Access control reduces risk via the defense in depth strategy.	TCP wrappers should be utilized.	1,4,6,7, 8
6	Telnet server	Telnet sends user login and password in clear text over the net and should not be used. SSH is a better alternative	Obj.	A network user could intercept root's password and gain unauthorized access or compromise production servers.	Telnet should be off or have minimal use.	4
7	FTP server	FTP sends user login and password in clear text over the net and should not be used. SSH is a better alternative.	Obj.	An network user could sniff the username and password for production accounts from the wire and compromise or gain unauthorized access to production machines.	FTP should be off or have minimal use.	4

No.	Checklist Item	Discussion	Type	Risk	Compliance	Source
8	RSH	The 'R' commands have had numerous historical security issues. User data is not encrypted on the net. SSH is a better alternative.	Obj.	An network user could sniff the username and password for production accounts from the wire and compromise or gain unauthorized access to production machines.	RSH should be off or have minimal use.	4
9	TFTP	TFTP is usually used on routers or diskless workstations. TFTP is very insecure and should not be used.	Obj.	No user name password required. Generally should not be on a production server.	TFTP should not be utilized.	4
10	NIS	NIS and its associated processes have had numerous security issues. Any normal user on a NIS system has access to the un-shadowed password file ripe for cracking.	Obj.	Unauthorized access to production machines via cracked password files or buffer overflows in RPC.	NIS should not be utilized.	4
11	Other RPC services	Site specific. Ask contact for information.	Obj.	Site Specific.	Site Specific.	NA
12	SMB Server/Client	Is there a need to share files/printers with Windows ? If not, turn off smbd and nmbd.	Obj.	Production systems me be remotely compromised, crashed, or DoS'd via SMB.	SMB should not be used.	4,
13	Portmap	If NFS/NIS is not used, turn off netfs and portmap.	Obj.	Production systems my be remotely compromised, crashed, or DoS'd via RPC which has a long history of security issues.	Portmap should be off if NIS or NFS are not in use.	2,4
14	LPR / CUPS	If no printing will occur on server, disable CUPSd/LPRng and/or remove the packages for these application	Obj.	Production systems may be remotely compromised, crashed, or DoS'd via CUPS or LPR.	LPR / CUPSD should be off.	4
15	X	X is generally not used on production servers, disable and or remove X packages.	Obj.	X has had a number of remote compromises. Also, if the box is compromised, X allows an attacker to shoot back a display on the remote system to their system.	X should be off and preferably not installed.	4,7
16	Email server	Will the server ""RECEIVE MAIL"" ? If not, removing the smtp daemon will have no impact on users ability to send mail	Obj.	Remote command execution and remote exploits exist for many mail daemons.	Mail server should be on localhost only or not on.	2,4,5
17	Web server	Many remote exploits have been accomplished via web servers. If this server will not function as a web server, remove this service.	Obj.	Many remote exploits have been accomplished via web servers.	Web servers should not be installed	4

No.	***Checklist Item***	***Discussion***	***Type***	***Risk***	***Compliance***	***Source***
18	SNMP	Will this server be monitored by SNMP? If not, turn off SNMP.	Obj.	May leak information to an attacker.	SNMP should not be installed.	1,4
19	Named	There have been numerous exploits for bind in the past years. If this server will not be a DNS server, remove named.	Obj.	Remote command, DoS, and remote exploit have all been accomplished via holes in Named.	Named should not be running.	4
20	DB Servers	If this server will not be providing database services remove this service.	Obj.	Remote command, DoS, and remote exploit have all been accomplished via holes in many DB servers.	All DB servers should be fully patched and secured or off.	4,6
22	Webmin.	Allows remote administration of a Linux machine.	Obj.	Remote command and remote exploit have been accomplished via holes in Webmin.	Webmin should be at the latest rev. or not installed.	6
23	NFS	Allows user to mount remote file systems.	Obj.	Users may mount something that they should not have access to. Also, may allow remote exploit via RPC.	Portmap, mountd, and nfsd should be off or not installed.	2,4,8
25	Restrict NFS to restricted ports.	The secure parameter in the exports file causes the NFS server to ignore requests that do not originate from ports less than 1024. The default is secure but it should be explicitly stated in exports.	Obj.	Reduce risk by using unprivileged ports.	Should be set in exports file.	1,4
26	net.ipv4.ip_forward = 0	Disable packet forwarding.	Obj.	If not set, it may allow network information leakage / DOS.	File value = 0	4,8
27	net.ipv4.conf.default.rp_filter = 1	Do not respond to packets that would cause us to go out. a different interface than the one to which we're responding.	Obj.	If not set, it may allow network information leakage / DOS.	File value = 1	4,8
28	net.ipv4.tcp_max_syn_backlog = 4096	Set to minimize the effects of SYN floods	Obj.	If not set, it may allow network information leakage / DOS.	File value = 4096	4,8
29	net.ipv4.conf.all.accept.source_route = 0	Disable inbound source routed packets to prevent spoofed IP addresses. 0	Obj.	If not set, it may allow network information leakage / DOS.	File value = 0	4,8
30	net.ipv4.conf.all.accept_redirects = 0	Reject inbound redirects.	Obj.	If not set, it may allow network information leakage / DOS.	File value = 0	4,8

No.	Checklist Item	Discussion	Type	Risk	Compliance	Source
31	net.ipv4.conf.all.send_redirects = 0	Don't send any redirects.	Obj.	If not set, it may allow network information leakage / DOS.	File value = 0	4,8
32	net.ipv4.conf.default.accept_redirects = 0	Reject inbound redirects.	Obj.	If not set, it may allow network information leakage / DOS.	File value = 0	4,8
33	SSH / SSHD	The ssh daemon and service provide a secure replacement for ftp and telnet.	Obj.	Reduces risk by encrypting network traffic.	SSH / SSHD should be installed and running.	4,6,8
34	File integrity tools	Are file integrity tools in use like Tripwire or AIDE?	Obj.	Reduces risk by allowing the Administrator to notice unauthorized file changes.	Check permissions.	4,6
35	Verify file permissions on VIP files.	Shadow, passwd,group,lilo.conf, and grub.conf are VIP system files and should have strict permissions	Obj.	Unauthorized access to system files and/or local compromise.	Verify all file permissions.	4,6,8
36	Verify that temp directories have the sticky bit set.	The sticky bit prevents non-owners from deleting files and directories.	Obj.	Unauthorized access to files or directories not owned by the accessing user.	Verify setting of sticky bit.	4,6,8
37	Identify /etc/hosts.equiv	Allows login without a password. The danger of this file is that some of the replacement for the 'r' commands will pay attention to this file if misconfigured.	Obj.	Remote compromise or unauthorized access.	Verify existence and contents of equiv file.	3,4,6
38	Identify users in ftpuser or ftpaccess files.	List of users NOT allowed to access the system via FTP. Remember that FTP sends the username/password in clear text over the network.	Obj.	FTP sends the username and password in clear text over the network.	Verify existence and contents of files.	4
40	Set permissions on LILO and GRUB to 600.	Lilo.conf and grub.conf are VIP system files and should have strict permissions	Obj.	Unauthorized local system access via booting into a root kit kernel.	Verify all file permissions.	4
41	Purge unused system accounts.	Remove or lock unused system accounts.	Obj.	Unauthorized remote system access and compromise via old accounts.	Examine users in the passwd file for unused accounts.	4,6
42	Verify no legacy + entries in passwd, shadow,group files.	The + is a marker for a NIS entry. If found, verify that is is actually used.	Obj.	Unauthorized remote system access and compromise.	Examine passwd,shadow,group for + . No + should be found.	4

No.	***Checklist Item***	***Discussion***	***Type***	***Risk***	***Compliance***	***Source***
43	Verify that no accounts have a null password.	Null passwords are obviously bad.	Obj.	Remote system access with authentication.	Verify the existence of passwords in shadow file.	4,6
44	Verify that root is the only zero account.	Root should be the only user with UID zero.	Obj.	Unauthorized root access.	Verify that root and only root is UID zero.	4,6
45	Check for IPTables default rules.	Iptables is an excellent method of restricting access from the network and logging network access and activity.	Obj.	Reduces risk by implementing access control.	Check for default rule sets.	4,6
46	RSYNC	The 'R' commands have had numerous historical security issues. User data is not encrypted on the net.	Obj.	Remote system compromise via stolen credentials.	The rsync service should not be running.	4
47	Verify default run level.	Note the default run level.	Obj.	Additional services run in each higher runlevel, this increases risk.	Note runlevel. Should be three.	4,7
48	Gather Hostname	Note the hostname.	Obj.	NA	NA Informational.	6
49	Gather OS version and type.	Note the OS manufacturer and version.	Obj.	NA	NA Informational.	6
50	Cat /proc/meminfo	The amount of physical system memory is good information to have as one would not expect this number to suddenly change.	Obj.	Malicious users may steal hardware.	Record command output.	6
51	Cat /proc/cpuinfo	The number and frequency of cpus is good information to have as one would not expect this to suddenly change.	Obj.	Malicious users may steal hardware.	Record command output.	6
52	Note Login Banners	Banners assist in the prosecution of users who illegally access servers or utilize corporate servers for non-business functions	Obj.	The absence of a banner may be argued in court as tacitly allowing any type of access.	Note banner string.	4

Checklist Items not covered by LinuxAuditor

No.	Checklist Item	Discussion	Type	Risk	Compliance	Source
1	Nessus Scan	Nessus targets known applications and tests them to determine if any discovered applications are vulnerable to public exploits.	Obj.	Reduces risk by finding potential vulnerabilities before a malicious individual.	Perform Nessus scan and save output as HTML.	6
2	Nmap Scan	Nmap scans find lists ports via a number of TCP/IP methods.	Obj.	If a malicious user opens a listening shell with netcat, the nmap baseline from previous audits will show that a new port is in a listen state.	Perform nmap scan and capture output.	6
3	Web server scan with WebserverScanner.pl	Discovers and identifies web servers on open ports.	Obj.	Applications may have hidden web servers listening . These servers usually have XSS or remote execution vulnerabilities.	No ports should yield a server version string.	6
4	Verify that SSH is utilized	SSH encrypts all traffic over the network and is a secure replacement for RSH, RCP, and Telnet	Obj,	Unauthorized access to accounts or services via stolen user information.	More ssh login activity should be seen versus	4,6

Source Reference Key

ID Number	Source
1	Smith, Patrick. Advanced Linux Networking. Addison Wesley, June 2002.
2	Laude, Mary. Auditing RedHat Linux 7.0. SANS GSNA Paper, July 2001.
3	Nemeth,Snyder, and Hein. Linux Administration Handbook. Prentice Hall, 2002.
4	Collaborative. URL: HTTP://www.cisecurity.org/Bench_Linux.html
5	Leen,Frisch. UNIX System Hardening Checklist. URL:HTTP://www.linuxmagazine.com/2002-09/harden_List.htm
6	Personal Experience.

ID Number	***Source***
7	Spitzner,Lance. URL: HTTP://www.justlinux.com/nhf/Security/Armoring_Linux.html
8	Collaborative. URL: HTTP://www.LinuxSecurity.com

With the scope and checklist, I arrange a last meeting with the manager to finalize plans for the audit and review the security policy for the system. In this case, the policy for the system was:

- Logins from dev and qa are not allowed local accounts so I should not see UIDs with QA or DEV in the local files.
- FTP is allowed but not encouraged.
- The suite of ”r” commands is allowed for backward compatibility with existing shell scripts but not encouraged.
- SSH is enabled and remote logins should be utilizing ssh whenever possible.
- Extraneous services not related to the function of the box should be off.

The policy set forth by the manager requires that I add items to the checklist that are not included in the list from the tool. I have to add:

- Check that ssh login is being utilized.
- Verify patch levels on packages and Kernel.

How to Conduct the Audit

The method that I used to prior to LinuxAuditor would be to open three windows, one on the audited machine, and two on my local system. The first step on the audited system would be to create a directory to hold all of the audit files. I usually create this in /tmp. Next, for each item in my checklist, I would execute the appropriate command on the audited system once to verify I was getting the expected output and a second time I would redirect into a file. I would then either scp or cut and paste from the newly created file to the window on my system. I moved the file right away to minimize my presence on the audited box. It is better to massage files locally and make an error that deletes files than to accidentally remove files from a production box. I would then open the file in VI, remove extra whitespace and replace the last whitespace with a tab or a comma. A write and quit completes that step. Finally, I would move to the second window on my laptop in which I have an OpenOffice spreadsheet audit report template that has multiple workbooks for the output from each command and a report summary. I would then import the data from each command on the audit checklist into the correct sheet. Once all off the data has been written into the spreadsheet, I fill in the audit summary sheet by checking the appropriate response and writing in any comments or mitigating factors for each item in the checklist. Then, I print each sheet out and make four copies. The last step is a meeting with the manager to review the audit and schedule a follow up meeting after the requesting group has had time to review my findings. The turn around time from the start of the audit to having a complete and printed report would was on the order of three hours or more per box. A reproduction of part of my audit checklist is shown in the table that follows.

Pass	*Fail*	*CheckList Item*	*Audit Finding*	*Mitigating Factors*
X		Are web servers present?	Apache 1.3.26	Installed but not running in any level
	X	Verify permissions on /etc/shadow are 400	600	

As you can imagine, this was a very tedious process to go through, especially on days when I had more than two or three servers to examine. In all of my audits of Linux, there has always been a core set of checklist items that I would always look at. These core items were always performed in the same fashion across all systems. For example, I would always run a find in this fashion. *# find / -type f -perm +004000 -printf "%p %m %u %g\n" 2>&1 |grep -v find .* After doing this repetitively across 20 boxes , I began to think that I should script all my commands. I started by checking the common open source sites on the Internet such as sourceforge for tools to automate an audit. I found numerous utilities for Solaris but none that did what I wanted on Linux. The Center for Internet Security has a benchmark tool that was very close but due the the fact that it could make changes I was not allowed to utilize that tool. I wrote LinuxAuditor based on the checklists from CIS, LinuxSecurity.com's quick reference card, and requirements based on risk assessments from management. The LinuxAuditor scripts remove almost all of the "Grunt Work" from the auditing process that I have at work.

Setup and Configuration

This section will assist an auditor in the usage, setup, and configuration of LinuxAuditor. The LinuxAuditor utility consists of three components; LinuxAuditor.pl, AuditServer.pl, and a report tool that utilizes Apache and Embedded Perl for the interface. The LinuxAuditor.pl script runs on the remote machine that is undergoing the audit. It writes all findings over the network back to AuditServer.pl which is running locally on the auditors personal machine. In order to perform an audit using the LinuxAuditor, there are a few steps that need to be done for the scripts to run correctly. The auditors personal machine must have Perl, MySQL, Apache, and Embedded Perl installed. The requirements for each audited host is simply Perl.
The requirements for each script are summarized in the table below.

Script / Location	***Requirement***
LinuxAuditor.pl (audited box)	Perl, Network connectivity back to the auditors machine.
AuditServer.pl (auditors box)	Perl, MySQL, DBI, Msql-MySQL modules, mod_perl, Apache, Embedded Perl.

The conventions used in this section are:

- Indented 10 point font is example output from a command.
- Indented 12 point font in a numbered list is example code.
- # This is the root shell prompt.
- > This a normal users shell prompt.
- RPM is used to search for packages, which assumes that if an auditor installed a particular package from source they will know it is installed regardless of the output from rpm.

Since most of the configuration centers on the auditor's own machine, this is a good place to start. Begin by using tar to untar and unzip the tool. The command looks like this: # tar xzvf LinuxAudit.tar.gz. The contents of the LinuxAudit directory are shown below.

```
wschroed@mummer:~/Documents/Projects/LinuxAudit> ls
Connector.pm  LinuxAuditor.pl  SQL_Setup  Server Website
```

- Connector.pm: The Perl module that contains the network connectivity information for the

LinuxAuditor.pl script.

- LinuxAuditor.pl: The script that performs the audit on each remote machine.
- SQL_Setup: Directory that contains the database setup script.
- Server: Directory that contains the AuditServer.pl script which receives the audit data over the network and writes it out to the database.
- Website: Files necessary for viewing the individual findings and the final summary report.

MySQL

The first package we will look for is MySQL. From a shell run # rpm -qa |grep -i mysql. The output should contain mysql-devel-VERSION, mysql-server-VERSION, mysql-VERSION. If any of these packages aren't in the output from the rpm query, find and install the missing packages. On my laptop the output looks like this:

```
/etc/init.d> rpm -qa |grep -i mysql
MySQL-client-4.0.15-0
MySQL-devel-4.0.15-0
MySQL-shared-4.0.15-0
mysql-shared-3.23.55-22
perl-Msql-Mysql-modules-1.2219-219
mysqlcc-0.8.10-29
MySQL-server-4.0.15-0
```

If this is a new installation, create the initial access privileges by running # mysql_install_db then run # /etc/init.d/mysql start to start MySQL. After MySQL starts, login by typing # mysql -u root mysql. Next passwords need to be assigned to restrict both local and remote access to the MySQL server. To set the passwords for root, login to mysql and run:

```
mysql> set password for root@localhost=PASSWORD('some_password');
mysql> set password for root@machine_hostname=PASSWORD('some_password');
```

Once the passwords have been set, logout of MySQL.

Apache

Now that MySQL has been set up, the next package to find is Apache. Run rpm -qa|grep -i apache. The output should contain both apache-VERSION and apache-devel-VERSION. The development package contains header files that will be necessary for the installation of Embedded Perl. I have installed Apache 1.3.28 from source so there isn't any sample output.

Mod_Perl

Mod_Perl should be installed to speed up the execution of the perl scripts for the report utility. To check if mod_perl is installed, run # rpm -qa |grep -i mod_perl. If it is missing and Apache has been installed via RPM, install the rpm version of mod_perl from the Linux CDs. This is the easiest way to install mod_perl. On the other hand, if Apache was installed from source ,download the source for mod_perl from http://perl.apache.org. Pay careful attention to the versions as version 2.x will only work with Apache 2.x. Installation details for mod_perl are found on the web site. I will not cover the full installation here as it is complicated and the syntax is completely different from version 1.x to version 2.x. The nutshell instructions for 1.x look like this:

```
Apache
# ./configure --prefix=/var/www --enable-shared=digest --enable-shared=auth_digest –enable-
shared=auth_db
                    followed by ........ make then make install
mod_perl
 # perl MakeFile.PL MP_AP_PREFIX=/var/www/ EVERYTHING=1
                    followed by ........... make, make test,make install
```

Perl-DBI and Msql-Mysql Modules

In order for the Perl scripts to be able to write to the database, the Perl DBI and Msql-MySQL modules need to be installed. On my laptop, the rpm query returns this:

```
/etc/init.d> rpm -qa Igrep -i Msql
perl-Msql-Mysql-modules-1.2219-219
/etc/init.d> rpm -qa Igrep -i dbi
perl-DBI-1.32-28
```

Most distributions include these Perl modules on the CDs or they can be installed from CPAN.

Embedded Perl

The last item that is necessary is Embedded Perl. Download the tar ball from http://perl.apache.org/embperl. Notice that are two versions of Embedded Perl. The 2.x version is for Apache 2.x and the 1.x versions are for Apache 1.x, so ensure that the correct package is installed or the report utility will not function. I use Apache 1.3.28 so I use Embedded Perl 1.3.6. To build Embedded Perl run # tar xzvf HTML-Embperl-1.3.6.tar.gz. Next, cd into the directory that was just created and run perl Makefile.PL. The installer will ask a series of questions about the configuration of EmbPerl. An example from an earlier install on RedHat is shown below.

```
# perl Makefile.PL
Build with support for Apache mod_perl?(y/n) [y]
Use /usr/include/apache as Apache source(y/n) [y]
Will use /usr/include/apache for Apache Headers
Enter path and file to start as httpd [/usr/include/apache/httpd]/usr/sbin/httpdApache Version Server
version: Apache/1.3.20 (Unix)  (Red-Hat/Linux)
Library for mod_env.c not found, please enter path to mod_env.so  []/usr/lib/apache/mod_env.so
```

Run # make test and then run # make install .
Once all all of the packages are installed, open httpd.conf in you preferred editor and add the following additions to the section that refer to the default document root:

```
1.  # This should be changed to whatever you set DocumentRoot to.
2.
3.       PerlModule HTML::EmbperlObject
4.  <Directory "/usr/local/apache/htdocs/">
5.       PerlSetEnv EMBPERL_OBJECT_APPNAME audit
6.       PerlSetEnv EMBPERL_ESCMODE 0
```

```
7.      PerlSetEnv EMBPERL_OBJECT_OPTIONS 16
8.      PerlSetEnv EMBPERL_OBJECT_BASE base.epl
9.      PerlSetEnv EMBPERL_OBJECT_DEBUG 0
10.     PerlSetEnv EMBPERL_OBJECT_SESSION_HANDLER_CLASS no
11.     <Files "*.html">
12.     SetHandler  perl-script
13.     PerlHandler HTML::EmbperlObject
14.     Options     +ExecCGI
15.     </Files>
16.
17.     <FilesMatch "*.epl">
18.     Order allow,deny
19.     Deny from all
20.     Satisfy All
21.     </FilesMatch>
```

Lines three through ten set up environment variables for Embperl and tell Apache how to handle Embedded Perl files. Line eight is noteworthy as the file base.epl is used to set certain variables like PATH and is used to export global variables to all Embperl documents. This makes coding easier as code that would be duplicated in multiple documents can be written only once and still be accessible in any document. The base.epl file that I use is displayed below:

```
[- $ENV{'PATH'} = "/usr/local/apache/htdocs/:/usr/local/apache/htdocs/LinuxAudit/:/usr/bin/:/bin" -]
[- delete @ENV{'IFS', 'ENV','CDPATH','BASH_ENV'} -]
[- Execute ('ActCountComp.epl') -]
[- Execute ('urlchk.epl') -]
[- Execute ('time.epl') -]
[- Execute ('ActCount.epl') -]
[- Execute ('*') -]
```

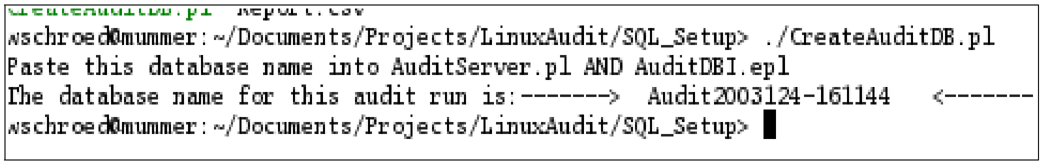
Lines eleven through fifteen set a document handler for Embperl files while lines seventeen through twenty one disallow access to raw files that may contain database passwords.

Website Files

To set up the web browser reporting tool, the files from the LinuxAudit/Website directory must be copied to the web server. To do this, edit MakeWeb.pl and add the path to the web servers document root. The script will create a LinuxAudit directory and copy the HTML files into it. That was a lot of information but now the web server is up and running and ready to generate the audit report after the audit is run.

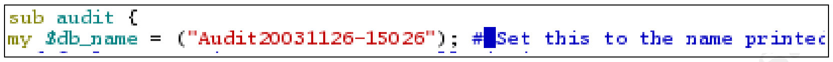
Database

The last administrative task that needs to be done is to create the database. Run the script called /LinuxAudit/SQL_Setup/CreateAuditDB.pl. This script creates an empty database and prints the name of the database it has just created to standard out as shown in the figure below.



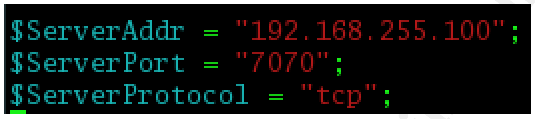
```
wschroed@mummer:~/Documents/Projects/LinuxAudit/SQL_Setup> ./CreateAuditDB.pl
Paste this database name into AuditServer.pl AND AuditDBI.epl
The database name for this audit run is:-------> Audit2003124-161144   <-------
wschroed@mummer:~/Documents/Projects/LinuxAudit/SQL_Setup>
```

Now cut / paste the new database name into AuditServer.pl as shown below.



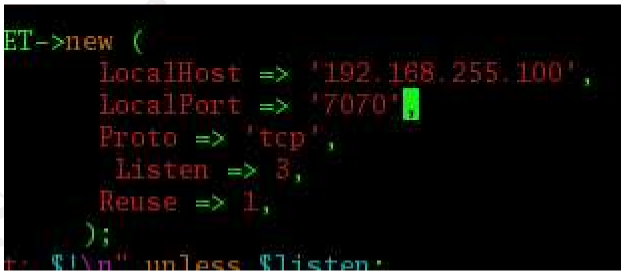
```
sub audit {
my $db_name = ("Audit20031126-15026"); #Set this to the name printec
```

Before the audit tool is run some network information needs to be verified. The file Connector.pm must be edited to have the correct IP of the auditors machine. The file that must be changed is shown in the figure below.



```
$ServerAddr = "192.168.255.100";
$ServerPort = "7070";
$ServerProtocol = "tcp";
```

I have set the default TCP port to 7070 as this assignment does not conflict with any applications on any of the boxes that I work on. This port may be changed if port 7070 is already in use on the either the server or the client machine. It does not matter what port number you pick but it must be the same in both the LinuxAuditor.pl and the AuditServer.pl scripts . To verify the state of port 7070 run netstat from the root shell like this # *netstat --na lgrep 7070.* If netstat does not return anything, the port is not in use and the only item that will have to be changed is the variable for the server IP address: $ServerAddr. Next ,move the files Connector.pm and LinuxAuditor.pl to the server to be audited. I usually just scp the files to the machine.
On the auditors box, edit the AuditServer.pl file so the script has the IP of the auditors machine. In the example I show below, I have set the IP of the server to the IP (192.168.255.100) assigned to my laptop by the administrator.



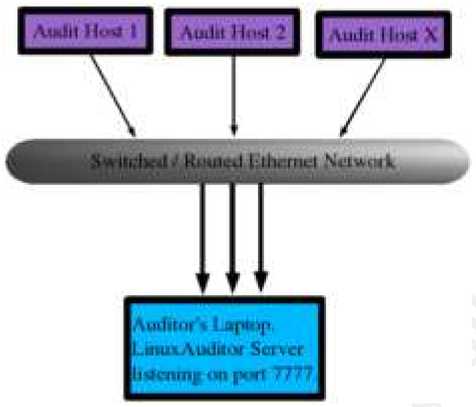
```
ET->new (
      LocalHost => '192.168.255.100',
      LocalPort => '7070',
      Proto => 'tcp',
       Listen => 3,
      Reuse => 1,
     );
t: $!\n" unless $listen;
```

Now we are ready to run audit. As root, start the server # *./AuditServer.pl.* The server will start and listen on the default port 7070. SSH to each box to be audited, then run LinuxAuditor.pl like this.

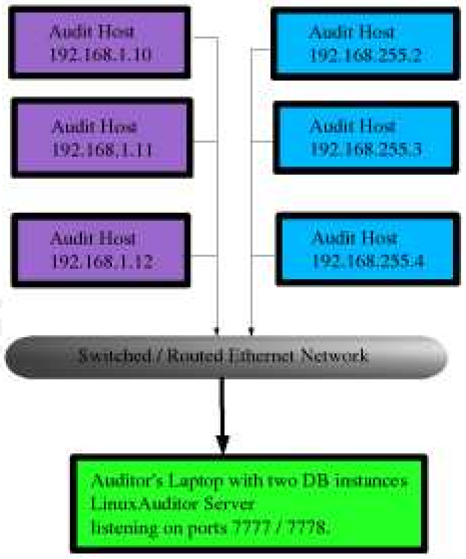
# ./LinuxAuditor.pl

You MUST be ROOT or the script will not complete every test. Root access is necessary because the script uses utilities like *lsof* that require the user to be root. The run duration takes anywhere from 40 to 300 seconds to run depending on the number of files in the file system and the disk configuration. On the SuSE box I audited which uses Pentium III and RAID 5 across twelve disks, the script runs in twenty five seconds from start to finish.
It is also possible to run multiple audits simultaneously as well. I have tested up to five clients at once connecting to my laptop. The default is a maximum of three but that can be increased as

CPU, database IO, and network bandwidth allow. A typical network host configuration is shown below.



This graphic exemplifies how three audit hosts can write back to a single auditing machine and a single database. If desired, multiple audit hosts are able to write back to a single auditing machine running multiple database instances of the audit database. This type of configuration may be useful when the auditor desires to isolate subnets to a different audit database. The graphical configuration of such a setup is shown below.



Once the audit tool is finished with a machine, grab the database name generated by the script and paste it into Auditdbi.epl in the LinuxAudit web document root. The Auditdbi.epl file contains the username,password, and active database name or the report tool. With this done, the results of the audit are viewable immediately in a web browser on the auditors machine. Open a web browser and navigate to the web page.

The script itself is written in perl and is fairly straightforward. The way it works follows this basic framework.

1. Run a command and capture the output into an array or a string.
2. Clean up the output and make it suitable to insert into a database.
3. Print the data to the network socket via IO::Socket.

To make the example clear I will discuss a short snippet of code to show how the audit tool functions.

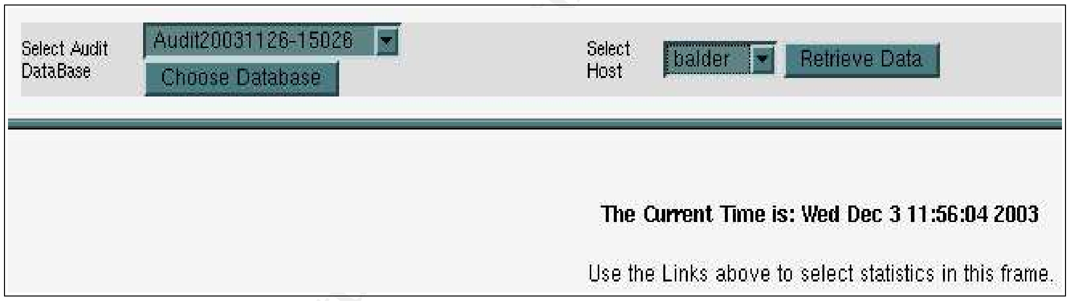
```
1.  if ( -e "/etc/sysctl.conf" ) {
2.              open (SYC, "</etc/sysctl.conf");
3.              while (<SYC>) {
4.              if ( $_ !~ m/^#/ && $_ !~ m/^\s+$/ ) {
5.              chomp $_;
6.              $_ =~ s/^\s+//g;
7.              push (@Sysctlc, $_);
8.              }
9.             }
10.if ( $#Sysctlc != "-1" ) {
11.       foreach $sysc ( @Sysctlc ) {
12.       print $sock "SYSCTLCONF: $name,$sysc\n";
13.       $sock->flush();
14.       }
15.       } else {
16.             print $sock "SYSCTLCONF: $name,Sysctl.conf is an empty file\n";
17.             $sock->flush();
18.             }
19.
20.} else {
21.             print $sock "SYSCTLCONF: $name,Sysctl.conf is not installed\n";
22.             $sock->flush();
23.                    }
24.
25.close (SYC);
```

Lines one and two simply make sure the file exists and if it does open it in read only mode. Next, in lines three through nine while the file handle is open, I match on any line that is not a comment or a blank line and push that value into an array. Line ten says that if the array has something in it we found data in sysctl.conf so we will print it to the socket. The alternative circumstance that I had to account for is that the file exists but was empty or just all comments as demonstrated on line sixteen. Finally, line twenty-one prints that the file was not found if it was not found on the system. The server listens for the connection from the audit script and when it receives a request it forks off a child to handle it. The server code is a loop that matches on key words sent by the audit script. I the example above the keyword is SYSCTLCONF. The relevant server code that handles the insertion into the database is shown below.

```
1. if ( $t =~ m/^SYSCTLCONF:\s+/ ) {
2. chomp $t;
3. $t =~ s/^SYSCTLCONF:\s+/'/g;
4. $t =~ s/,/','/g;
5. $t =~ s/\b$/'/g;
6. print "$t\n";
7. my $sth = $dbh->do (qq{ INSERT INTO SYSCTLCONF VALUES( $t ) }) || die $dbh->errstr;
8. }
```

What these lines of code accomplish is the removal of the keyword from the string and the substitution of ticks around each word. This properly formats the string for insertion in the database on line seven.
The reporting tool queries the database and generates HTML reports for each section in the audit checklist and produces a summary report that allows the auditor to verify each of the audited items against the checklist that resides in the Report table. The report can then printed and checked off or the method that I use is to wget the reports and insert each report page as a table in the report. The report can be viewed by following this URL: http://YourDomain_or_IP/LinuxAudit/mframe.html.

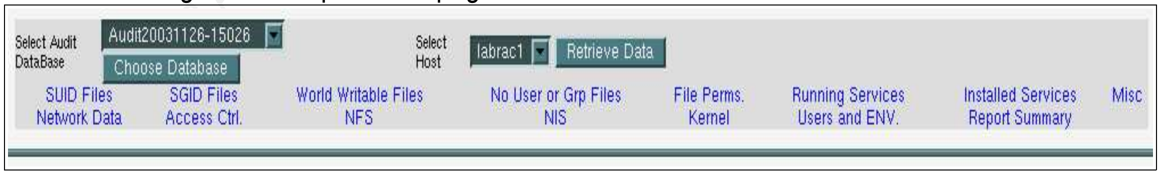
To use the reporting tool, open a web browser and direct the url to the website set up on the auditing laptop. A page will display as shown in the figure below.



The Current Time is: Wed Dec 3 11:56:04 2003

Use the Links above to select statistics in this frame.

Select the database name that was inserted into Auditdbi.epl from the drop down list. This action results in the population of a drop down box that lists each host in the current audit database. Clicking “Retrieve Data” fills the top frame that was previously empty with the listing of each audit section. Clicking on a link opens that page in the bottom frame as shown below.



Queries that possibly may display large numbers of return values display 50 items per page with links on the bottom of each page to go to each page by page number. To see the results for a different host, select it from the drop down list and click 'Get' then click on the name of the report.

The code that generates the HTML reports is perl wrapped in special tags so that it is not interpreted as HTML. A small example of the code from the report tool is displayed and discussed below.

```
1. [- $sth = $dbh -> prepare ("SELECT DenyRule from TCPWD where HostName =
   '$HostChoice' "); -]
2.       [- $sth -> execute; -]
3. <table width="65%" border="0" cellspacing"0" cellpadding="0" name="fileperms"
   align="center" class=tabs>
4. <tr>
5. <th align="center">Default TCP Wrappers Deny Rules</th>
6. </tr>
7. [$ while @row = $sth -> fetchrow_array $]
8. <tr>
9. <td align="center" class="tab" >[+ @row[0] +]</td>
10.</tr>
11.[$ endwhile $]
12.</table>
```

Line one performs the setup for the query and then line two executes it. Then lines three through twelve loop and print a simple HTML table that displays the values returned by the query.

An automated tool does make auditing easier, but it has some disadvantages. I wrote the utility so I completely understand how it works. An auditor who picks up my tool who does not know Perl may have problems and not know how to resolve them. I could have made a mistake in the code that I am not aware of that will invalidate parts of an audit. Another disadvantage may arise if asked to perform an audit on an unfamiliar network. An auditor may find that they do not have network access from the client to the audit server. In this situation, my audit tool would be useless. Despite the disadvantages, utilizing an automated tool to perform audits can be advantageous. A tool results in a reduction in the amount of time spent on a server issuing commands,redirecting output, and massaging files to make them "report presentable". The script is typographically correct every time it is used. This yields greater time efficiency for the auditor by preventing The second advantage is consistency. A script will not forget to run netstat or run a command with different switches than intended resulting in having to repeat the last item over again. The third advantage is time. I can accomplish much more in the same time frame using a script than running commands by hand. The fourth advantage is historical reproducibility. The audit results are immediately inserted in a database which I control so the results can not be edited or manipulated by a third party. This also means that I have the flexibility to compare two separate audits and discover what has changed over time. The fifth advantage is if a team of auditors is using the same script, the data gathered will be in a consistent format. The sixth and final advantage is flexibility. If some new item needs to be added to the checklist or an application service is added, it is fairly simple to add code to an existing script to look for the new checklist item. For the most part, adding new functions to the script is mostly cut and paste of existing code. This is especially true of the code in the reporting tool.

Conducting the Audit.

Audit on RedHat AS 3.0.

Now we are finally set to actually run the audit. The audit is run in the following order.

1. Run LinuxAuditor first. Running Nmap or Nessus first may leave changes in the network status or may even crash running processes.
2. Log on to the box to verify anything not included in the tool and delete the two files used by the tool to run the audit..
3. Run Nmap.
4. Run WebScanner.pl (Finds and ID's web servers, tool written by me)
5. Run Nessus.

After verifying the network settings and moving the audit script to labrac1, I ran the audit tool. The results from this portion of the audit are shown in following table taken from the audit report tool.

1: Run LinuxAuditor

Check "Pass"	***Check "Fail"***	***CheckList***	***Discussion***	***Audit Finding***	***Mitigating Circumstances***	
		Hostname		labrac1		
		Operating System	What version is this OS? ie... SuSe 8.2 or RedHat AS 3.0	Red Hat Linux Advanced Server release 2.1AS Pensacola		
X		Install performed from verified media	To reduce the risk of installing trojaned/corrupt software only install from verified media. Can you trust the copy of the copy that was obtained from a friend.	Install performed from RedHat OEM Cd Set.		
	X	Post install OS update performed from verified media	There are always OS, Kernel, and application updates after a new OS release so ensure that a box is patched before it is released into production.	No patching done.		
X		OS Updates are scheduled on a regular basis	Patches come out all the time. What mechanism is used to ensure that patches are applied on a regular basis.	OS updates are scheduled to be done on the approved change time on Friday.		
	X	Install only necessary services per function.	Determine the role the box will perform to determine the services that will be installed.	Service set not minimized.		
			Services			
X		Restrict NFS clients to restricted ports	The secure parameter in the exports file causes the NFS server to ignore requests that do not originate from ports less than 1024. The default is secure but it should be explicitly stated in exports.	No NFS Options		

		NFS File systems shown as mounted via the mount command.		labjs1:/export/home/oracle on /home/oracle type nfs rw addr=192.168.255.252		
		NFS File systems shown as permanently mounted via /etc/fstab		Nothing Mounted in fstab via NFS		
		NFS File systems exported from /etc/exports		Nothing exported from etc/exports		
X		NIS	The root of all evil		NIS should be replaced with LDAP. This is scheduled for 2nd Quarter.	
				nisdomainname-osglab.com		
				rpcinfo-program 100004 is not available		
				securenets file not installed		
				yp.conf-domain osglab.com broadcast		
	X	Telnet Server	Telnet sends user login and password in clear text over the net and should not be used. SSH is a better alternative	23	Telnet should be disabled as rlogin is available for compatibility.	
X		RSH	The 'R' commands have had numerous historical security issues. User data is not encrypted on the net. SSH is a better alternative.	513	Allowed for compatibility with existing scripts	
X		rsync	The 'R' commands have had numerous historical security issues. User data is not encrypted on the net.	514	Allowed for compatibility with existing scripts	
X		FTP Server	FTP sends user login and password in clear text over the net and should not be used. SSH is a better alternative.	21	Allowed for compatibility with existing scripts	

X		TFTP	TFTP is usually used on routers or diskless workstations. TFTP is very insecure and should not be used.	TFTP not running at time of audit		
X		SSH/SSHD	The ssh daemon and service provide a secure replacement for ftp and telnet.	22		
X		Email Server	Will the server RECEIVE MAIL? If not removing the smtp daemon will have no impact on users ability to send mail.	25	RedHat requires this service. It is running on localhost only.	
X		Web Server	Many remote exploits have been accomplished via web servers. If this server will not function as a web server remove this service.	A web server was not running at time of audit		
X		Named	There have been numerous exploits for bind in the past years. If this server will not be a DNS server remove named.	A named server was not running at time of audit		
X		SNMP	Will this server be monitored by SNMP ? If not turn off SNMP.	SNMP was not running at time of audit		
X		DB servers	If this server will not be providing database services remove this service.	MySQL was not running at time of audit		
X		Disable X	X is generally not used on production servers, disable and or remove X packages.	X was not running at time of audit		
X		SMB Server/Client	Is there a need to share files / printers with Windows ? If not turn off smbd and nmbd.	SMB/NMB was not running at time of audit		
		Netfs and Portmap	If NFS / NIS is not use turn off netfs and portmap.	111	Used by NFS / NIS	
X		Disable LPRng or CUPSD	If no printing will occur on server, disable CUPSd / LPRng and or remove the packages for these application.	CUPS was not running at time of audit		

Kernel						
X		net.ipv4.ip_forward = 0	Disable packet forwarding.	0		
X		net.ipv4.conf.default.rp_filter = 1	Do not respond to packets that would cause us to go out. a different interface than the one to which we're responding.	1		
	X	net.ipv4.tcp_max_syn_backlog = 4096	Set to minimize the effects of SYN floods.	1024	Should be increased to 4096.	
X		net.ipv4.conf.all.accept.source_route = 0	Disable inbound source routed packets to prevent spoofed IP addresses.	0		
	X	net.ipv4.conf.all.accept_redirects = 0	Reject inbound redirects.	1	Should be set to zero.	
	X	net.ipv4.conf.all.send_redirects = 0	Don't send any redirects.	1	Should be set to zero.	
	X	net.ipv4.conf.default.accept_redirects = 0	Reject inbound redirects.	1	Should be set to zero.	
Files and Permissions						
		Verify file permissions on VIP files. *\/etc/shadow*	This is an important file therefore the permissions need to be restrictive.	root, 400, PASS,		
		/etc/passwd		root, 644, PASS		
		/etc/group		root, 644, PASS		
	X	*/etc/grub.conf*		root, 777, FAIL	Change to 644 at a minimum.	
		/etc/lilo.conf		ND, Not Installed, ND		
X		Verify that temp directories have the sticky bit set	The sticky bit prevents non-owners from deleting files and directories.	/var/tmp, Set		
	X	Verify that temp directories have the sticky bit set	The sticky bit prevents non-owners from deleting files and directories.	/tmp, NotSet	Set sticky bit.	

	X	File integrity tools	Are file integrity tools like AIDE or Tripwire in use ?	Not in use.	Install and configure	
Access Control						
	X	Check for IPTables Rules		Iptables not in Kernel	Utilize packet filtering	
	X	TCP Wrappers	TCP wrappers is a reliable way to restrict and log access to TCP daemons. TCP wrappers utilizes a simple text file for access control and does not require a reboot.	Empty deny File	Utilize TCP Wrappers access controls	
	X	TCP Wrappers		Empty allow File		
X		Verify no legacy + entries in passwd,shadow,and group files	The + is a marker for a NIS entry. If found verify that is is actually used.	No plus in /etc/passwd No plus in /etc/group No plus in /etc/shadow		
X		Verify that there are no accounts with null password	Null passwords are obviously bad.	No Null passwords		
X		Verify root is only zero account	Root should be the only user with UID zero.	root		
X		Identify /etc/hosts.equiv	Allows login without a password. The danger of this file is that some of the replacements for the r commands will pay attention to this file if they are misconfigured. If hosts.equiv is not used, remove it.	/etc/hosts.equiv Not Installed		

X		Create appropriate warning banners	Necessary in the prosecution of crackers and has the added benefit of obfuscating information on versions and OS.	/etc/issue: Red Hat Linux Advanced Server release 2.1AS /etc/issue.net: Red Hat Linux Advanced Server release 2.1AS /etc/motd: This system is for authorized users only. Individuals using this computer system are subject to having all of their activities on this system monitored and recorded by system personnel. Anyone using this system expressly consents to such monitoring and is advised that if such monitoring reveals possible evidence of criminal activity improper usage or hacking system personnel may provide the evidence of such monitoring to proper management of the CME.		
X		Identify users /etc/ftpusers or /etc/ftpaccess	List of users NOT allowed to access the system via FTP. Remember that FTP sends the username/password in clear text over the network.	allow uid ftp allow gid ftp deny-uid 99 %65534 deny-gid 99 %65534		
X		Identify users /etc/ftpusers or /etc/ftpaccess	List of users NOT allowed to access the system via FTP. Remember that FTP sends the username/password in clear text over the network.	root bin daemon adm lp sync shutdown halt mail news uucp operator games nobody		

Misc.						
X		Verify default run level	The default run level for servers should be level 3. At this run level, X does not run which helps minimize running services.	Text Login Default		
		Data from /proc/meminfo	This information is good to know as one would not expect the amount of system memory to suddenly change	MemTotal: 4374412 kB		
		Data from /proc/cpuinfo	Baseline CPU information	processor: 0 vendor_id: GenuineIntel model name: Intel R Xeon TM CPU 2.80GHz processor: 1 vendor_id: GenuineIntel model name: Intel R Xeon TM CPU 2.80GHz processor: 2 vendor_id: GenuineIntel model name: Intel R Xeon TM CPU 2.80GHz processor: 3 vendor_id: GenuineIntel model name: Intel R Xeon TM CPU 2.80GHz		
End Of Report. Generated by LinuxAudit on Mon Dec 1 10:36:14 CST 2003						

2: Verify that SSH is utilized in preference to RSH.

Log onto the server and run :

```
Labrac1# grep ssh messages lwc -l
3
labrac1# grep rsh messages lwc -l
779
```

The commands above finds and counts each type of respective login method. The count overwhelmingly shows that ssh is not utilized in preference to rsh.

3: Run Nmap

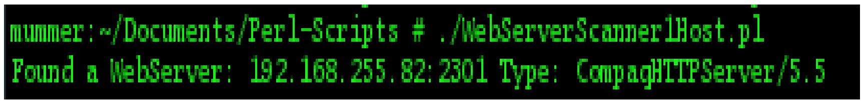
As root, I ran # *nmap -sX -sU -P0 -O -oN labrac1 192.168.255.83*. Each switch has a different effect on the output of the scan the first -sX scans for TCP ports that are closed by sending a packet with the FIN,PSH, and URG flags all set to true. This scan works because closed ports will respond with a RST while open ports will ignore these packets. The -sU scans for UDP, -P0 indicates that I should scan the host even if it is not pingable, -O tries to identify the operating system -oN saves the output to the file labrc3.

The scan returned thirteen open TCP and UDP ports and the operating system guess was correct.

```
# nmap 3.30 scan initiated Tue Oct 28 16:40:00 2003 as: nmap -sX -sU -P0 -O -oN labrqac1
                        192.168.255.82
Interesting ports on 192.168.255.82:
(The 3102 ports scanned but not shown below are in state: closed)
Port       State       Service
21/tcp     open        ftp
22/tcp     open        ssh
23/tcp     open        telnet
111/tcp    open        sunrpc
111/udp    open        sunrpc
513/tcp    open        login
514/tcp    open        shell
861/udp    open        unknown
863/tcp    open        unknown
875/udp    open        unknown
878/tcp    open        unknown
2301/tcp   open        compaqdiag
7000/udp   open        afs3-fileserver
Device type: general purpose
Running: Linux 2.4.Xl2.5.X
OS details: Linux Kernel 2.4.0 - 2.5.20
Uptime 0.297 days (since Tue Oct 28 09:56:19 2003)
```

4: Run WebServerScanner.pl

After the portscan is complete, I use a tool of my own creation that tries to identify http/https servers on any open port. The tool works by finding open ports with nmap, connecting to any ports that it finds and then capturing the server string that is returned. The output is shown in the figure below.



```
mummer:~/Documents/Perl-Scripts # ./WebServerScanner1Host.pl
Found a WebServer: 192.168.255.82:2301 Type: CompaqHTTPServer/5.5
```

5: Run Nessus

Last, I run Nessus with all of the dangerous options enabled. I also specifically turn off the optimization options to ensure that the scan does not miss anything. To start the scan initialize the Nessus server by running:

```
# /usr/local/sbin/nessusd -D
```

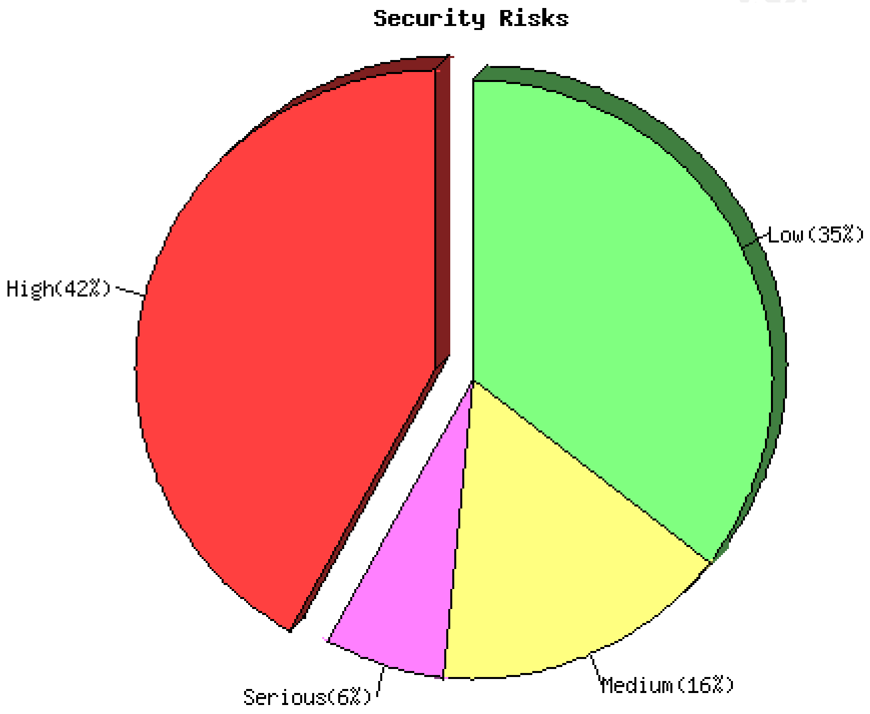
followed by:

```
$ /usr/local/bin/nessus
```

and logging into the Nessus server.

The relevant output from the Nessus generated report follows:

192.168.255.82



List of open ports :

- ftp (21/tcp) *(Security hole found)*
- ssh (22/tcp) *(Security hole found)*
- telnet (23/tcp) *(Security warnings found)*
- sunrpc (111/tcp) *(Security warnings found)*
- login (513/tcp) *(Security warnings found)*
- shell (514/tcp) *(Security warnings found)*
- cpq-wbem (2301/tcp) *(Security hole found)*
- compaq-https (2381/tcp) *(Security hole found)*

- filenet-tms (32768/tcp) *(Security warnings found)*
- general/tcp *(Security warnings found)*
- unknown (863/tcp) *(Security notes found)*
- unknown (878/tcp) *(Security notes found)*
- sunrpc (111/udp) *(Security notes found)*
- unknown (861/udp) *(Security notes found)*
- unknown (875/udp) *(Security warnings found)*
- filenet-tms (32768/udp) *(Security warnings found)*
- general/udp *(Security notes found)*
- general/icmp *(Security warnings found)*

[back to the list of ports]

Vulnerability found on port ftp (21/tcp)

The remote Wu-FTPd server seems to be vulnerable to an off-by-one
overflow when dealing with huge directory structures.
An attacker may exploit this flaw to obtain a shell on this host.
*** Nessus solely relied on the banner of the remote server
*** to issue this warning, so it may be a false positive
Solution : Upgrade to Wu-FTPd 2.6.3
Risk Factor : High
CVE : CAN-2003-0466
BID : 8315
Nessus ID : 11811

[back to the list of ports]

Vulnerability found on port ftp (21/tcp)

It was possible to disable the remote FTP server
by connecting to it about 3000 times, with
one connection at a time.
If the remote server is running from within [x]inetd, this
is a feature and the FTP server should automatically be back
in a couple of minutes.
An attacker may use this flaw to prevent this
service from working properly.
Solution : If the remote server is GoodTech ftpd server,
download the newest version from http://www.goodtechsys.com.
BID : 2270
Risk factor : Serious
CVE : CAN-2001-0188
BID : 2270
Nessus ID : 10690

[back to the list of ports]

Vulnerability found on port ftp (21/tcp)

It was possible to kill your FTP server
by reading a MS/DOS device, using
a file name like CON\CON, AUX.htm or AUX.
A cracker may use this flaw to make your
server crash continuously, preventing
you from working properly.
Solution : upgrade your system or use a
FTP server that filters those names out.
Risk factor : High
Nessus ID : 10929

back to the list of ports]

Vulnerability found on port ssh (22/tcp)

You are running a version of OpenSSH which is older than 3.7.1
Versions older than 3.7.1 are vulnerable to a flaw in the buffer management
functions which might allow an attacker to execute arbitrary commands on this
host.
An exploit for this issue is rumored to exist.
Solution : Upgrade to OpenSSH 3.7.1
See also : http://marc.theaimsgroup.com/?l=openbsd-misc&m=106375452423794&w=2
http://marc.theaimsgroup.com/?l=openbsd-misc&m=106375456923804&w=2
Risk factor : High
CVE : CAN-2003-0693, CAN-2003-0695
BID : 8628
Nessus ID : 11837

[back to the list of ports]

Vulnerability found on port ssh (22/tcp)

You are running a version of OpenSSH which is older than 3.4
There is a flaw in this version that can be exploited remotely to
give an attacker a shell on this host.
Note that several distribution patched this hole without changing
the version number of OpenSSH. Since Nessus solely relied on the
banner of the remote SSH server to perform this check, this might
be a false positive.
If you are running a RedHat host, make sure that the command :
rpm -q openssh-server
Returns :
openssh-server-3.1p1-6
Solution : Upgrade to OpenSSH 3.4 or contact your vendor for a patch
Risk factor : High
CVE : CVE-2002-0639, CVE-2002-0640

BID : 5093
Nessus ID : 11031

[back to the list of ports]

Vulnerability found on port ssh (22/tcp)

You are running a version of OpenSSH older than OpenSSH 3.2.1
A buffer overflow exists in the daemon if AFS is enabled on
your system, or if the options KerberosTgtPassing or
AFSTokenPassing are enabled. Even in this scenario, the
vulnerability may be avoided by enabling UsePrivilegeSeparation.
Versions prior to 2.9.9 are vulnerable to a remote root
exploit. Versions prior to 3.2.1 are vulnerable to a local
root exploit.
Solution :
Upgrade to the latest version of OpenSSH
Risk factor : High
CVE : CVE-2002-0575
BID : 4560
Nessus ID : 10954

[back to the list of ports]

Warning found on port ssh (22/tcp)

You are running OpenSSH-portable 3.6.1 or older.
There is a flaw in this version which may allow an attacker to
bypass the access controls set by the administrator of this server.
OpenSSH features a mechanism which can restrict the list of
hosts a given user can log from by specifying a pattern
in the user key file (ie: *.mynetwork.com would let a user
connect only from the local network).
However there is a flaw in the way OpenSSH does reverse DNS lookups.
If an attacker configures his DNS server to send a numeric IP address
when a reverse lookup is performed, he may be able to circumvent
this mechanism.
Solution : Upgrade to OpenSSH 3.6.2 when it comes out
Risk Factor : Low
CVE : CAN-2003-0386
BID : 7831
Nessus ID : 11712

[back to the list of ports]

Warning found on port ssh (22/tcp)

You are running OpenSSH-portable 3.6.1p1 or older.
If PAM support is enabled, an attacker may use a flaw in this version

to determine the existence or a given login name by comparing the times
the remote sshd daemon takes to refuse a bad password for a non-existant
login compared to the time it takes to refuse a bad password for a
valid login.
An attacker may use this flaw to set up a brute force attack against
the remote host.
*** Nessus did not check whether the remote SSH daemon is actually
*** using PAM or not, so this might be a false positive
Solution : Upgrade to OpenSSH-portable 3.6.1p2 or newer
Risk Factor : Low
CVE : CAN-2003-0190
BID : 7482, 7467, 7342
Nessus ID : 11574

[back to the list of ports]

Information found on port ssh (22/tcp)

Remote SSH version : SSH-1.99-OpenSSH_3.1p1
Nessus ID : 10267

Warning found on port telnet (23/tcp)

The Telnet service is running.
This service is dangerous in the sense that
it is not ciphered - that is, everyone can sniff
the data that passes between the telnet client
and the telnet server. This includes logins
and passwords.
You should disable this service and use OpenSSH instead.
(www.openssh.com)
Solution : Comment out the 'telnet' line in /etc/inetd.conf.
Risk factor : Low
CVE : CAN-1999-0619
Nessus ID : 10280

[back to the list of ports]

Information found on port telnet (23/tcp)

Remote telnet banner :
Red Hat Linux Advanced Server release 2.1AS/i686 (Pensacola)
login:
Nessus ID : 10281

Warning found on port cpq-wbem (2301/tcp)

Remote Compaq HTTP server version is: 5.5
Nessus ID : 10746

Information found on port general/tcp

Remote OS guess : Linux Kernel 2.4.0 - 2.5.20
CVE : CAN-1999-0454
Nessus ID : 11268

This file was generated by Nessus, the open-sourced security scanner.

Report on the Audit

Executive Summary

The objective of this audit was to evaluate the current state of the operating system of the host labrac1, record a baseline of this host with respect to network and services, and to make recommendations on items that need to be addressed prior to the installation of Oracle by the DBA's and final production racking.
The summary of notable items is shown in the list below:

- Operating system patching has not been completed.
- The set of services has not been minimized.
- Network Kernel parameters have not been correctly set.
- File permissions on the configuration file for the Linux boot loader are to permissive.
- The sticky bit is not set on /tmp thereby allowing users to delete other users files contained in /tmp.
- File integrity tools are not in use.
- Iptables is not utilized to restrict access to labrac1.
- TCP Wrappers is not utilized to restrict access to labrac1.
- SSH is not utilized in favor of rlogin and rcp.

Audit Findings

Observation:
Operating System and Kernel patches have not been applied.
Risk:
Numerous local and remote exploits exist in the outdated packages. Stability issues were addressed as well. The organization faces significant risks, mostly from stability without package upgrades. Issues regarding the use of memory as well as numerous security issues have been resolved in the latest Kernel. Updating the OS and packages would eliminate the vulnerabilities uncovered for FTPd and SSH in the Nessus scan.
Recommendation:
The organization has opted to use YUM for updates. This box is scheduled to have package and Kernel updates installed during the Friday change window.
Cost:
The cost for this is minimal has the developer time to code the necessary changes to YUM have already been budgeted in the previous fiscal year. But, the cost to test kernel and package

updates in development and QA can be significant as patches come out frequently resulting in the need for full time staff to evaluate changes that may negatively impact our systems.
Report Link:
NA

Observation:
The set of running services has not been minimized
Risk:
Additional services increase the risk to the system by increasing the exposure to vulnerabilities.
Recommendation:
The organization is very close with respect to closing extraneous services. Telnet is the last service that should be turned off.
Cost:
The cost for this is minimal all authorized users have access to, and should be using ssh. SSH is installed by default on all Linux servers by default.
Report Link:
/home/wschroed/GSNA/WebOutPut/labrac1/services.html?HostChoice=labrac1
and
/home/wschroed/GSNA/WebOutPut/labrac1/installedservs.html?HostChoice=labrac1

Observation:
Network kernel parameters have not been set.
Risk:
Malicious users or accidental network misconfiguration may result this server becoming a packet router.
Recommendation:
Increase the following kernel parameter to 4096.
/proc/sys/net/ipv4/tcp_max_syn_backlog = 1024
Set each of the kernel parameters to zero (Off) instead of one (On).
/proc/sys/net/ipv4/conf/all/Accept_Redirects
/proc/sys/net/ipv4/conf/all/Send_Redirects
/proc/sys/net/ipv4/conf/default/Accept_Redirects
Cost:
The cost for this is minimal as this can be accomplished by a single administrator with a simple shell script.
Report Link:
/home/wschroed/GSNA/WebOutPut/labrac1/kernel.html?HostChoice=labrac1

Observation:
File permissions on /etc/grub.conf are to permissive at 777
Risk:
Malicious users or accidental misconfiguration may result this server becoming non bootable.

Recommendation:
Set the permissions on grub.conf to at least 644.
Cost:
The cost for this is minimal as this can be accomplished by a single administrator with a simple shell script.
Report Link:
/home/wschroed/GSNA/WebOutPut/labrac1/fileperms.html?HostChoice=labrac1

Observation:
The sticky bit is not set on /tmp.
Risk:
Users delete files not owned by the user.
Recommendation:
Set the sticky bit on /tmp
Cost:
The cost for this is minimal as this can be accomplished by a single administrator with a simple shell script.
Report Link:
/home/wschroed/GSNA/WebOutPut/labrac1/fileperms.html?HostChoice=labrac1

Observation:
File integrity tools are not in use.
Risk:
Malicious users may replace important system files with trojan copies.
Recommendation:
Install and configure a utility such as AIDE.
Cost:
The costs involved may be significant for enterprise licenses and administration time to install, configure, and manage the software.
Report Link:
NA

Observation:
Access controls are not utilized. Linux installs by default Iptables packet filtering software and TCP wrappers software to control network access to services and systems.
Risk:
Malicious users may gain unauthorized access to systems and compromise or steal data. Misconfigured application may send SYN floods resulting in an accidental DoS.
Recommendation:
Configure Iptables and TCP wrappers to limit access to the system and services as appropriate for the function of the server. Utilize the packet filtering capabilities of Iptables to reduce the impact from network events like SYN floods.
Cost:

The costs involved are minor as administrators may add the configuration to each machine at kickstart and tweak from that point.

Report Link:

/home/wschroed/GSNA/WebOutPut/labrac1/accessctrl.html?HostChoice=labrac1

Observation:

SSH is not utilized in preference to rsh.

Risk:

Root passwords or other important user names and password may be sniffed off the network from unencrypted sessions.

Recommendation:

Configure SSH and implement a plan to move away from the rlogin and rcp.

Cost:

The costs and time involved may be significant to re write scripts to support SSH.

Report Link:

NA

Conclusions

The audit conducted on labrac1 revealed several issues with the deployment of RedHat Linux. Although some of the issues were serious, none of the problems are insurmountable. I am certain that with the continued support from management to increase the awareness of security in general these issues will be addressed in future build outs.

Conclusion- LinuxAuditor

The number of Linux machines placed intro commercial use is growing exponentially every year. The German Government has recently made the decision to abandon Microsoft Windows and server software in favor of SuSE Professional and SuSE Enterprise Server at a price comparable to that from Microsoft. In the United States, UBS Warburg has committed all production systems to RedHat. As auditors, we will certainly be looking at many different distributions of Linux in production environments and be tasked with assessing each distributions security characteristics and configuration. Hopefully, LinuxAuditor will grow and be able to assist in this expanding environment.

References

Printed

- Hoelzer, David. Track 7 Auditing Networks,Perimeters and Systems, Auditing Principles and Concepts. The SANS Institute, 2003.
- Collaborative. Track 7 Auditing Networks,Perimeters and Systems, Hands-On Workbook. The SANS Institute, 2003.
- Smith, Patrick. Advanced Linux Networking. Addison Wesley, June 2002.
- Laude, Mary. Auditing RedHat Linux 7.0. SANS GSNA Paper, July 2001.
- Nemeth,Snyder, and Hein. Linux Administration Handbook. Prentice Hall, 2002.

Electronic

- Naidu Krishni. Auditing Linux. URL: HTTP://www.sans.org/score/checklists/AuditingLinux.doc
- Collaborative. URL: HTTP://www.LinuxSecurity.com
- Spitzner,Lance. URL: HTTP://www.justlinux.com/nhf/Security/Armoring_Linux.html
- Collaborative. URL: HTTP://www.cisecurity.org/Bench_Linux.html
- Leen,Frisch. UNIX System Hardening Checklist. URL:HTTP://www.linuxmagazine.com/2002-09/harden_List.htm
- Wreski, David. Linux Security Quick Reference Card. URL: _ HTTP://linuxsecurity.com/docs/QuickRefCard.pdf.