



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Auditing Systems, Applications, and the Cloud (Audit 507)"
at <http://www.giac.org/registration/gsna>

Auditing a Linux FTP and DNS Server: An Administrators Perspective

GSNA Practical Version 2.1, Option 1
(Amended July 5, 2002)

Author: Sean Baumann
Date: September 20, 2003

© SANS Institute 2003, Author retains full rights.

TABLE OF CONTENTS

ABSTRACT.....	5
ASSIGNMENT 1 – RESEARCH IN AUDIT, MEASUREMENT PRACTICE, AND CONTROL.....	6
IDENTIFY THE SYSTEM TO BE AUDITED.....	6
EVALUATE THE RISK TO THE SYSTEM.....	13
CURRENT STATE OF PRACTICE	18
ASSIGNMENT 2 – CREATE AN AUDIT CHECKLIST	21
INTRODUCTION:.....	21
CONVENTIONS USED:.....	22
PHYSICAL SECURITY CHECKLISTS	23
SYSTEM CHECKLISTS	25
ASSIGNMENT 3 – CONDUCT THE AUDIT	49
SYSTEM 2 CHECKLIST EXECUTION: SOFTWARE PATCHES.....	49
SYSTEM 5 CHECKLIST EXECUTION: KERNEL VULNERABILITIES AND SETTINGS.....	58
SYSTEM 8 CHECKLIST EXECUTION: DoS	64
SYSTEM 9 CHECKLIST EXECUTION: SYSTEM RECONNAISSANCE	67
SYSTEM 10 CHECKLIST EXECUTION: SYSLOG AUDIT	71
SYSTEM 11 CHECKLIST EXECUTION: NAMED CONFIGURATION.....	74
SYSTEM 13 CHECKLIST EXECUTION: FTP CONFIGURATION	80
SYSTEM 14 CHECKLIST EXECUTION: USER QUOTAS.....	84
SYSTEM 15 CHECKLIST EXECUTION: SYSTEM FILE INTEGRITY	86
SYSTEM 17 CHECKLIST EXECUTION: IP TABLES	93
RESIDUAL RISK	96
IS THE SYSTEM AUDITABLE?.....	96
ASSIGNMENT 4 – RISK ASSESSMENT.....	98
SUMMARY	98
SYSTEM 2 CHECKLIST STEP 3: SENDMAIL VERSION	98
SYSTEM 5 CHECKLIST STEPS 3, 4 AND 5: KERNEL VERSION	100
SYSTEM 9 CHECKLIST STEP 4: OS FINGERPRINT	102
SYSTEM 10 CHECKLIST STEP 2: SYSLOG SPOOF.....	103
SYSTEM 10 CHECKLIST STEP 1: BIND VERSION	104
SYSTEM 14 CHECKLIST STEPS 2, 4, 5 AND 6: USER QUOTAS	104
SYSTEM 17 CHECKLIST STEPS 1 AND 3: IPTABLES IMPLEMENTATION	105
CONCLUSIONS	107
REFERENCES.....	108

List of Figures

Figure 1 - Network Diagram.....	7
Figure 2 – Syslog Traffic Diagram	8
Figure 3 – FTP Traffic Diagram.....	10
Figure 4 – DNS Traffic Diagram.....	12
Figure 5 –Audit Environment.....	22
Figure 6 – Up2date Command Output	50
Figure 7 – Blackbox Website.....	51

<u>Figure 8 – Blackbox Version Command</u>	52
<u>Figure 9 – OpenSSL Website</u>	53
<u>Figure 10 – OpenSSL Version Command</u>	54
<u>Figure 11 – OpenSSH Website</u>	55
<u>Figure 12 – OpenSSH Version Command</u>	56
<u>Figure 13 – Iplog Website</u>	57
<u>Figure 14 – Iplog Version Command</u>	58
<u>Figure 15 – Kernel Version</u>	59
<u>Figure 16 – Up2date Kernel Version</u>	59
<u>Figure 17 – Kernel.org Website</u>	60
<u>Figure 18 – RedHat Website</u>	61
<u>Figure 19 – Kernel Options Download</u>	63
<u>Figure 20 – Loadable Kernel Modules</u>	64
<u>Figure 21 – Nessus DoS Configuration</u>	65
<u>Figure 22 – Nessus DoS Report</u>	66
<u>Figure 23 – Xinetd Settings</u>	67
<u>Figure 24 – Unauthorized Banners</u>	68
<u>Figure 25 – Authorized Banners</u>	69
<u>Figure 26 – BIND Version Dig</u>	70
<u>Figure 27 – BIND RPM Version</u>	70
<u>Figure 28 – Nmap OS Fingerprint</u>	71
<u>Figure 29 – Syslog Poison</u>	72
<u>Figure 30 – Syslog Spoofed Messages</u>	72
<u>Figure 31 – Syslog Spoofed Kernel</u>	72
<u>Figure 32 – Named Syslog Configuration</u>	73
<u>Figure 33 – Named Syslog Messages</u>	74
<u>Figure 34 – Cisco Syslog Facility</u>	74
<u>Figure 35 – Bind Version</u>	75
<u>Figure 36 – ISC Website</u>	75
<u>Figure 37 – BIND chroot</u>	76
<u>Figure 38 – named.conf zone transfer</u>	77
<u>Figure 39 – Zone transfer attempt</u>	78
<u>Figure 40 – named.conf allow-query</u>	78
<u>Figure 41 – DNS query attempt</u>	79
<u>Figure 42 – DNS query id numbers</u>	80
<u>Figure 43 – Unauthorized FTP version</u>	81
<u>Figure 44 – ftp-test user</u>	81
<u>Figure 45 – Authorized FTP version</u>	82
<u>Figure 46 – FTP directory traversal</u>	83
<u>Figure 47 – “Other” permissions</u>	84
<u>Figure 48 – FTP large files</u>	84
<u>Figure 49 – Quota kernel option</u>	85
<u>Figure 50 – Quota Version</u>	85
<u>Figure 51 – /etc/fstab quota option</u>	86
<u>Figure 52 – /etc/fstab quota option</u>	86
<u>Figure 53 – Initialize tripwire DB</u>	87

Figure 54 – rotate logs	88
Figure 55 – Tripwire DB update	92
Figure 56 – Root's email	93
Figure 57 – IPTables file	94
Figure 58 – IPTables chkconfig	94
Figure 59 – Netcat listener	95
Figure 60 – Connecting to listener	95
Figure 61 – IPTables log entries	95
Figure 62 – Up2date update	99
Figure 63 – Up2date complete	100
Figure 64 – Kernel update	101
Figure 65 – Nmap OS detection	103
Figure 66 – Final TCP nmap scan	106
Figure 67 – Final UDP nmap scan	107

List of Tables

Table 1 – Software Packages and OS Versions	6
Table 2 – Physical Risks	14
Table 3 – System Risks	15
Table 4 – Physical Access Checklist	23
Table 5 – Redundant Power Checklist	24
Table 6 – Replacement Hardware Checklist	24
Table 7 – Installed Packages Checklist	25
Table 8 – Software Patches Checklist	26
Table 9 – System Settings Checklist	27
Table 10 – Sendmail Settings Checklist	30
Table 11 – Kernel Vulnerability Checklist	31
Table 12 – Network Settings Checklist	32
Table 13 – Daemon and Open Ports Checklist	34
Table 14 – DoS Checklist	35
Table 15 – System Reconnaissance Checklist	37
Table 16 – Syslog Checklist	37
Table 17 – Named Configuration Checklist	38
Table 18 – FTP Bounce Checklist	40
Table 19 – FTP Configuration Checklist	41
Table 20 – User Quota Checklist	43
Table 21 – File Integrity Checklist	44
Table 22 – Administrative Access Checklist	45
Table 23 – IPTables Checklist	47
Table 24 – Hosts.allow Checklist	47

Abstract

I chose to complete the practical assignment for the GSNA certification by conducting an audit of a system that I manage, and reporting the results of that audit. Since I am an administrator of the system, I carried out the audit from the perspective of an administrator. The practical assignment has been broken down into four sections. The first section describes the subject of the audit; in this case, it is a Linux FTP and DNS server that also provides rudimentary Syslog service. It also describes the risks associated with the system, as they relate to network and system security. The second section provides detailed checklists that an auditor would use to conduct a thorough audit of this specific system. In the Third section, I provided the results of ten of the most important provided checklists. Screen captures have been provided as proof of their execution. In the final section, I have explored the residual risks, costs associated with fixing noncompliant items, and the actions taken to further secure the system.

© SANS Institute 2003, Author retains full rights.

Assignment 1 – Research in Audit, Measurement Practice, and Control

Identify the System to be Audited

I am one of many people, known as the administrative team, who are responsible for the administration of key Linux systems at Company X. Company X is a moderate sized, well-known company in the life sciences industry.

The audited system is a Compaq DL360 that is running RedHat Linux 9 as the operating system. Company X uses the system as a file transfer protocol (FTP) server, a domain name service (DNS) server, and a Syslog server. The workgroup to which I belong not only designed and built the system, but we are also responsible for maintaining all aspects of the server. Company X considers this server somewhat critical to the organization; the administrative team must correct all detected failures within eight hours. The information technology (IT) organization and the business units that rely on these services have jointly developed a service level agreement (SLA) that defines this requirement.

The server is running a highly modified version of the RedHat 9 operating system. The administrative team has developed a “template system” that allows us to rapidly deploy identical Linux based systems. As RedHat releases new minor operating system revisions, the administrative team can update the template system. These changes can then be rolled out to all of the IT managed Linux systems. The template system includes all of the services that the IT organization is required to support; however, we only enable the specific services that required for a particular server. The supported software packages include Apache Web Server, PHP, MySQL, BIND9, OpenSSL, OpenSSH, Very Secure FTPd (vsftpd), and Syslog. In addition, all of the Linux servers deployed are running Tripwire and IPTables to provide additional security measures. The following table shows the software packages that will be included in the audit:

Table 1 – Software Packages and OS Versions

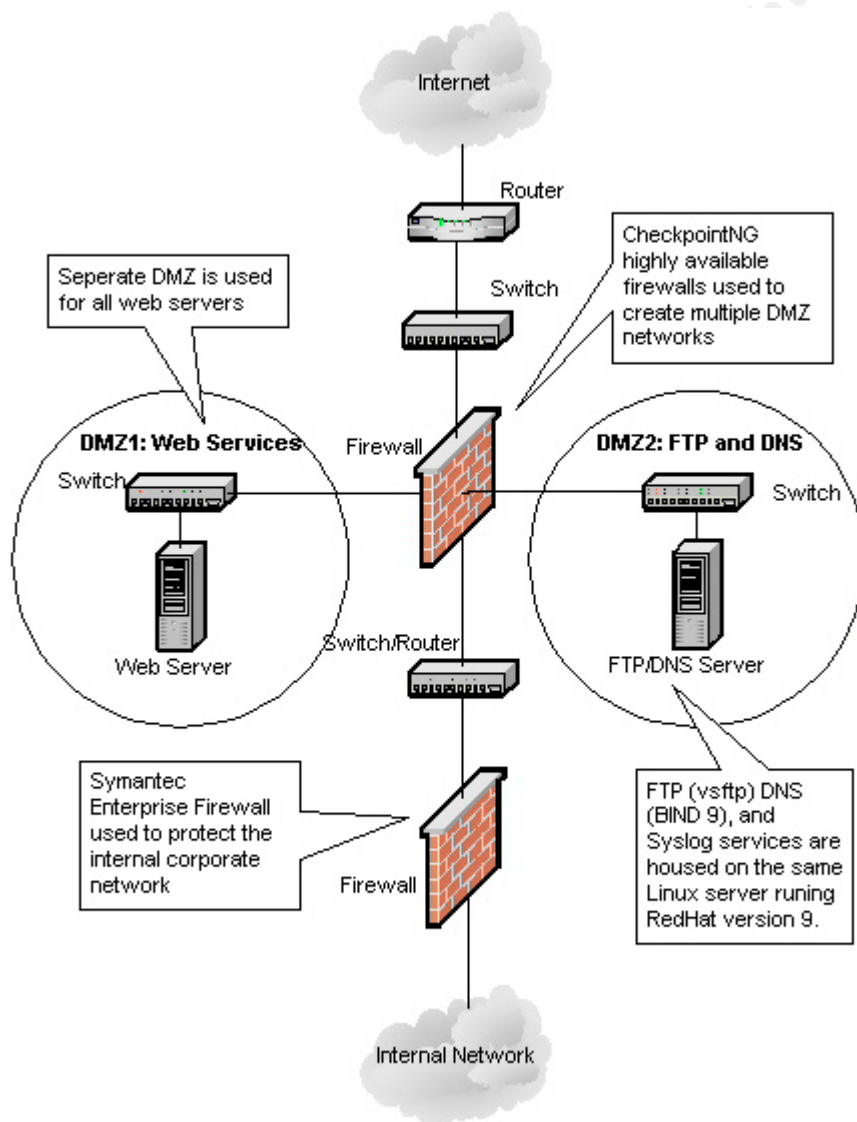
Software Package/OS	Version	Comments
RedHat Linux	9 (kernel 2.4.20-8)	Installed w/ minimum required packages
OpenSSL	0.9.7a	In this case, used with OpenSSH
OpenSSH	3.6.1p1	Used for administrative access
Tripwire	2.3.1-17	Used to monitor the file system
IPTables	1.2.7a-2	Used to allow only specific network traffic
BIND	9.2.1	chroot environment is used
Vsftpd	1.1.3-8	Fast and flexible ftp server

sendmail	8.12.5	Required to deliver local email
Syslogd	1.4.1	Daemon to log system alerts

Network Connectivity

The server is connected to one of Company X's demilitarized zone (DMZ) networks, which is a protected network that resides behind a pair of highly available (HA) Checkpoint firewalls. Figure 1 shows the server's exact position within the network infrastructure.

Figure 1 - Network Diagram



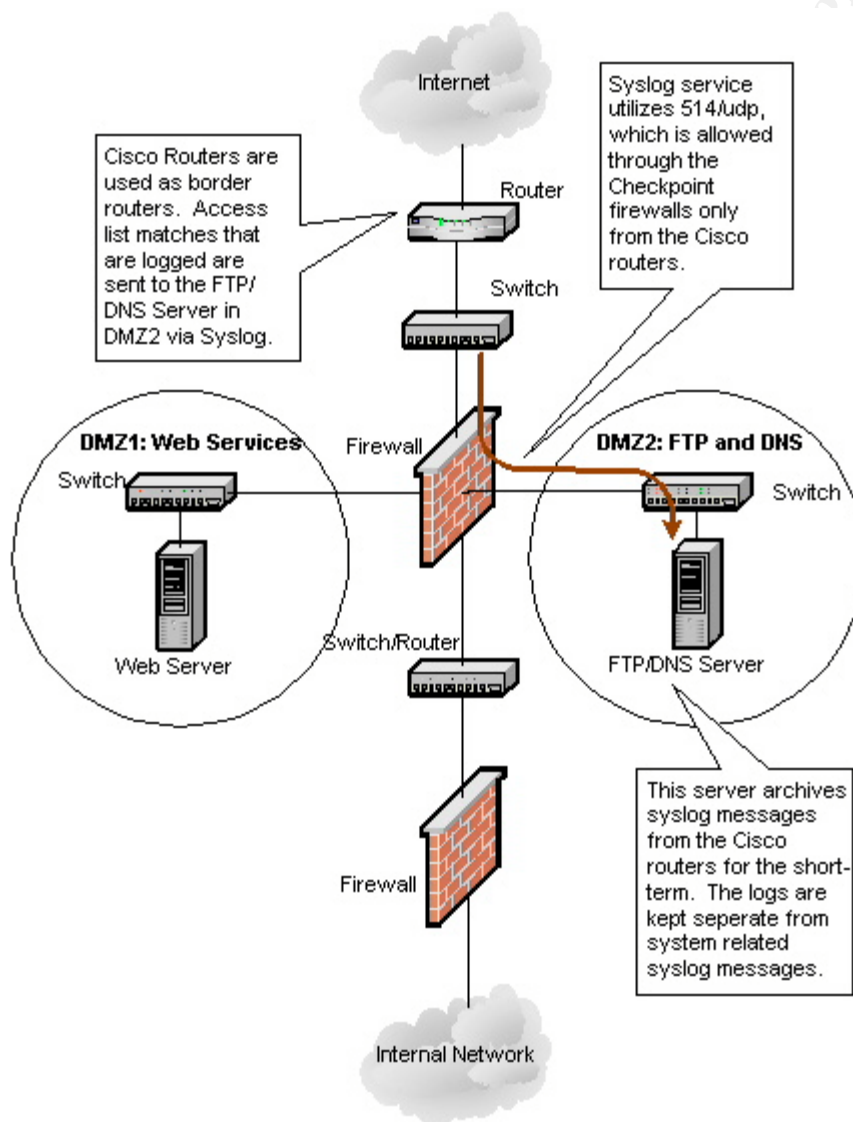
Major Services

Outlined below are the three main services provided by the server:

Syslog

The syslog daemon on this server provides a logging facility for messages generated by the internet Cisco routers. The messages are gathered and logged to a separate file for later review by the administrative team. Company X does not consider the Cisco router syslog data as mission critical, it is stored on this server for trouble-shooting convenience. This, however, will be included in the audit.

Figure 2 – Syslog Traffic Diagram



FTP Drop box

Several organizations within Company X utilize the server for file sharing with outside scientific collaborators, business partners, or software vendors. Company X's security policy mandates that internal hosts cannot FTP-Put to any servers on the internet (some exceptions have been made); all FTP-Put operations must originate from the company's own FTP server. Its effectiveness aside, this policy was instituted to mitigate the risks of losing company intellectual property. The company also has a policy concerning email attachment size, which not only limits their size to 5MB, but also limits them to a small subset of file extensions (in order to mitigate email-borne virus outbreaks).

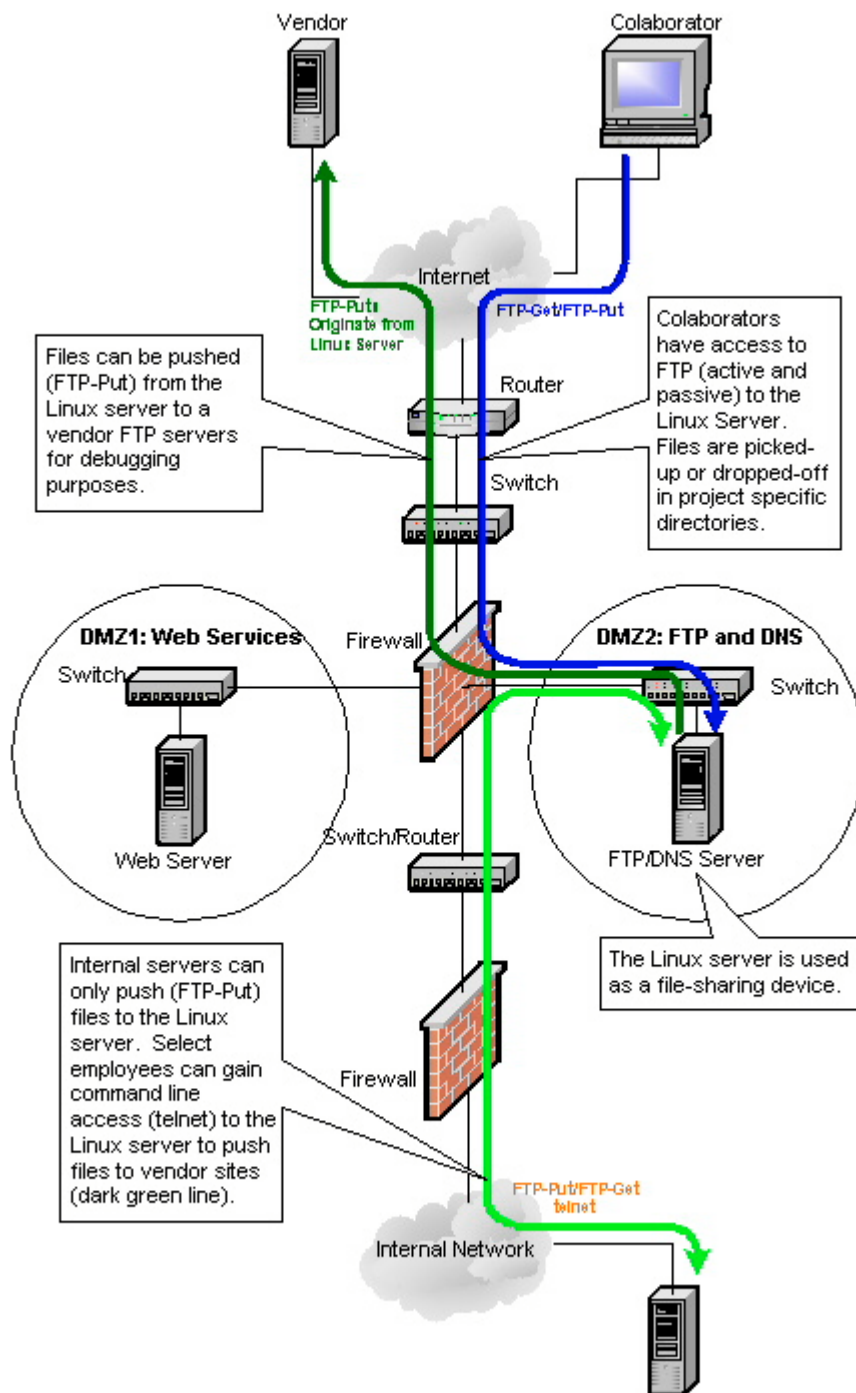
An employee (internal user) that requires the use of the FTP server for their project must submit an application to the administrative team. The team reviews the application, and, upon its approval, creates groups and accounts specific to that project. The users utilize the FTP server in two ways:

- 1.) Users push (FTP-Put) files to vendors for debugging purposes. In this case, an internal user would FTP the files to the FTP server using an FTP client. The user then logs in to the FTP server, using a telnet (legacy) or SSH client, and pushes the files to the destination system.
- 2.) Users also use the FTP server as a "drop box." Either a partner or an internal user can deposit files on the FTP server for the other involved party to retrieve later. In this case, the partner or collaborator would be required to have his or her own account for FTP access.

The following diagram depicts the traffic flow for the FTP server.

© SANS Institute 2003

Figure 3 – FTP Traffic Diagram



DNS Server

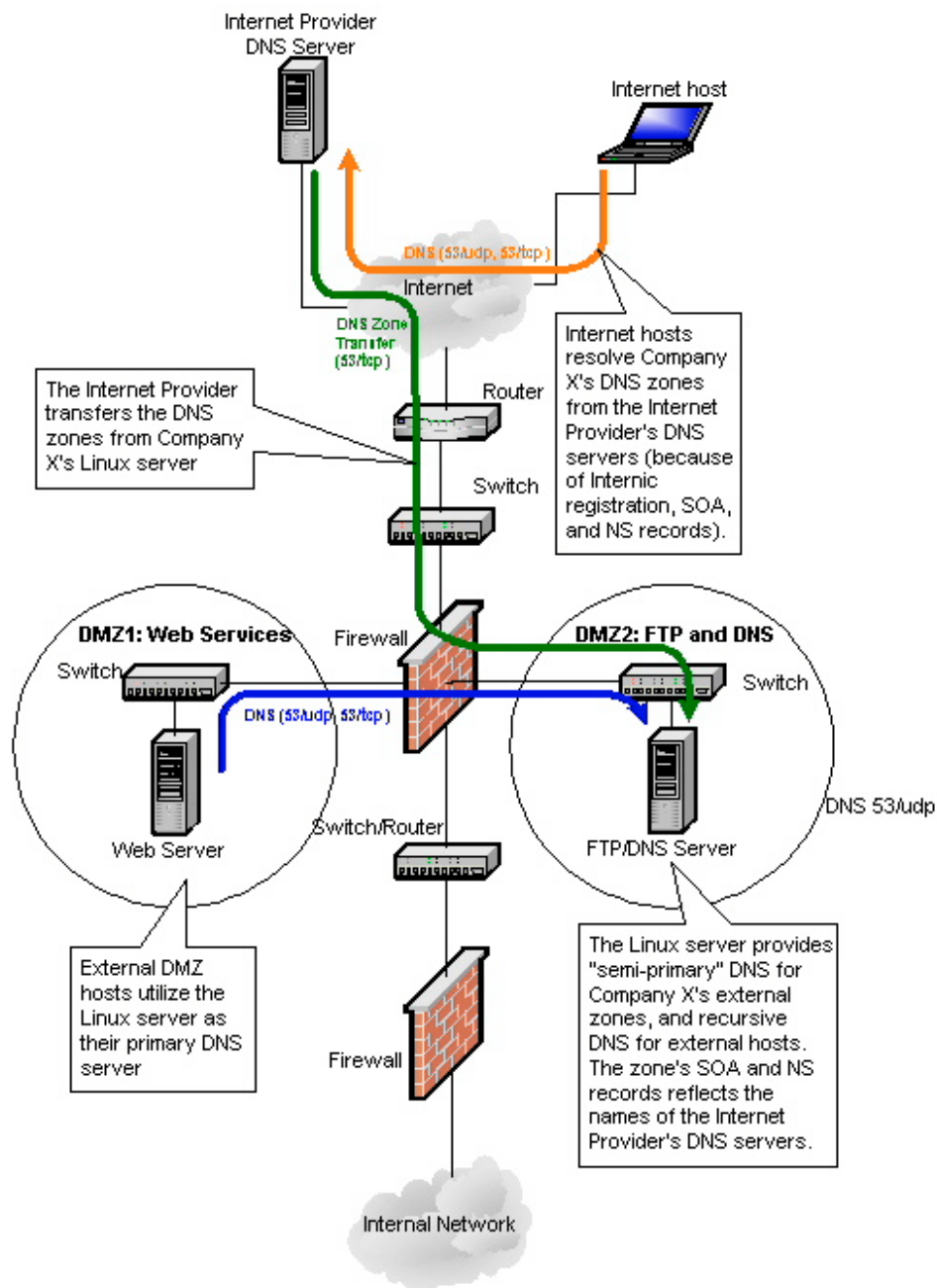
The BIND9 DNS server running on the system is a part of the RedHat distribution. The software is running in a “chroot” configuration, where the root directory for the daemon is separate from the system’s root directory. This ensures that a compromise of the DNS server does not compromise the entire system, since the daemon is “jailed” to that separate root directory. The DNS server provides two types of service:

- 1.) Machines connected to a DMZ network (either DMZ1 or DMZ2) are provided with recursive resolution capabilities. In other words, hosts in the company’s external networks utilize this Linux server as their DNS server.
- 2.) The DNS server is authoritative for all of Company X’s external DNS zones. However, the Internic has Company X’s internet provider listed as the primary for these zones. The internet provider’s DNS server, configured as a slave, transfers the zones from Company X’s DNS server. In essence, this allows Company X to control the DNS zones, but all internet resolution of those zones occurs from the internet provider’s DNS servers. No internet clients need to be able to resolve DNS directly from this Linux server.

Figure 4 shows the traffic flows for this service.

© SANS Institute 2003, Author retains full rights.

Figure 4 – DNS Traffic Diagram



Administrative Traffic

The administrative team conducts system maintenance by connecting to it with a secure shell (SSH) client. The system is running OpenSSH to provide this access. The RedHat OpenSSH RPM package is not used. Historically, the administrative team has found that RedHat lags behind on patches for OpenSSH, so we have chosen to compile it from source.

Additional Auditable Services

In addition to the services that the system provides and the operating system, two other packages will be specifically included in the audit. Tripwire and IPTables both provide additional system security.

The system utilizes IPTables as a host based firewall. The current system template is the administrative team's "demo" for the IPTables software. The software only permits access to the required network services: FTP (21/tcp), secure shell (22/tcp), telnet (23/tcp), and DNS (53/tcp, 53/udp).

Tripwire provides system file integrity checking. The software monitors key files and specific directories. The system executes the tripwire software at specific intervals to generate change reports. The system's root user receives the reports in the form of an email. The administrative team periodically reviews the reports for anomalies.

Network Security

The network security topology of the DMZ infrastructure and the security policies implemented on Company X's firewalls are beyond the scope of this audit. This is an audit of only the Linux FTP/DNS/Syslog server.

Evaluate the risk to the system

Considering that the system is running software packages that comprise three of the top ten Unix system vulnerabilities, as listed on the SANS/FBI top 20 vulnerability list, evaluation of the associated risks is essential to the success of an audit. OpenSSH, File Transfer Protocol, and BIND all made the list. All of these must be carefully evaluated to ensure that the system remains reliable and free of unacceptable vulnerabilities. These risks, however, should not discredit the risks associated with common flaws such as an incorrectly configured operating system.

Since I am an administrator of the system, I evaluated the risks with a level of knowledge of both the current state of the system and the administrative team's standard practices. While I do have root level access, and understand the general requirements of the system, I was not directly involved in the build of the Linux template or this server. The following tables describe the existing risks:

Table 2 – Physical Risks

#	Description	Threat (L/M/S/H)	Likelihood of Exploit (L/M/S/H)	Consequence	Risk Level (L/M/S/H)
P1	Someone could compromise the server from the system console.	Moderate An unauthorized user could access the system console.	Low The Company has not advertised the services to the general user population. However, the DNS name ftp.companyx.com does exist.	An unauthorized user or attacker that gains physical access to the system could fully compromise the system and disclose sensitive company intellectual property.	Moderate: It is unlikely that anyone outside the administrative team would know the location of the server in question, especially since there is a limited set of system users.
P2	A power outage could cause a prolonged system outage.	High Company X's location has suffered brown outs and power failures due to high winds and thunderstorms.	Moderate Power failures occur at frequent intervals in Company X's area.	If there is a power outage, the system could lose power, and possibly damage the hardware.	High: A system outage would cause an interruption in FTP service for critical projects. Short outages (2 to 4 hours) are acceptable, but long outages are not.
P3	A hardware failure could cause an extended system outage.	Low The Compaq server could have a hardware failure. The DL360s have had faulty power supplies. (experience)	Low Company X has had very few issues with Compaq equipment, besides the power supply.	A hardware failure would cause a service outage. The SLA indicates that the team must restore the system within eight hours.	Low: It is unlikely that the hardware would fail.

Table 3 – System Risks

#	Description	Threat (L/M/S/H)	Likelihood of Exploit (L/M/S/H)	Consequence	Risk Level (L/M/S/H)
S1	An attacker could compromise the system.	High The hacker community regularly discovers new system exploits. Company X's IDS system has detected vulnerability probes in the past.	Low The administrative team proactively monitors for new bug and vulnerability reports for the operating system and software packages. It is unlikely that an attack will be successful if the system is properly maintained	If the threat were realized, an attacker would compromise the system, possibly gain root access, and steal sensitive company intellectual property.	Moderate This system is accessible through the internet, and as such, it is highly likely that it will eventually become a target. However, it is unlikely that an attack would result in the compromise of the system.
S2	The system is subjected to a denial-of-service (DoS) attack.	Moderate DoS and DDoS attacks are trivial for attackers to launch. Virus and worm code can incorporate these attacks.	Moderate Company X is a high profile company, and as such, could become a target.	The system would be effectively unavailable to the network. This would cause a system outage; however, no information would be lost.	Moderate A DoS attack would cause a service outage, which may be acceptable for short periods.
S3	The system reveals too much information during hacker reconnaissance.	Moderate Would-be attackers can probe for information, which would allow them to target specific OS vulnerabilities or software packages.	Moderate Reconnaissance is a popular method for targeting specific vulnerabilities. There are tools freely available to the public for this purpose.	A hacker could gain enough information about the system to begin targeting specific software packages or operating system flaws. This could lead directly to a system compromise	Moderate Information gathering could lead directly to an attack against the system, and potentially a compromise. This is a moderate risk to the system.

S4	The syslog logging service is interrupted or messages are spoofed.	Low An attacker could overwhelm the syslog service with unrelated, forged alerts. Syslog traffic is not encrypted or authenticated.	Moderate An attacker can easily spoof syslog packets. This is due to a fundamental flaw in the protocol; there is no authentication or encryption of the UDP traffic.	The syslog file stored on the server would contain erroneous data, and could grow large enough to cause system resources to become exhausted.	Moderate From experience, this does not seem to be a highly useful or popular attack. Company X's IDS system has not captured any syslog attempts to the system. However, this is still a potential DoS mechanism.
S5	The DNS server is susceptible to cache poisoning.	Moderate Cache poisoning is one of the oldest and most effective ways of compromising DNS. The possible attacks are trivial due to insecurities in the protocol (Stewart).	Low The administrative team maintains the most recent revisions of BIND9; it is unlikely that the threat would be realized.	The FTP server on this system does rely on DNS reverse look-ups to allow access. If the forward resolution does not match the reverse, the system denies access. If the cache were to be poisoned, then an attacker could cause a DoS.	Moderate-Low A cache poisoning attack could cause an outage of the FTP service. However, an attempt to redirect a user to another site would not have an effect, as clients do not utilize this system for DNS (The servers that do, do not require it)
S6	An attacker compromises the DNS server software and alters zones files.	Moderate The Security Focus website lists many vulnerabilities for the BIND DNS server. BIND is also non-trivial to configure properly.	Low The administrative team did not register the server with the Internic. However, an attacker may scan for the service.	An attacker would compromise the system, and potentially be able to alter zone files. The attacker could redirect external services (email, web, ftp, etc) to a server of his or her choosing.	Moderate-Low There is a low to moderate level of risk by running a BIND DNS server in Company X's environment.

S7	The FTP server is susceptible to the FTP-Bounce attack.	Low The administrative team selected the vsftpd server for its security features; there are no known security issues with this software package.	Low If the software package is current then there is a low possibility of exploiting this service.	If the server was susceptible to this attack, an attacker could use this server for reconnaissance of Company X's network (CERT).	Low There is a low level of risk associated with this attack.
S8	FTP users can glean information from the system, or access other user's data.	Moderate Some information on this server may be proprietary. Users may try to find information about other projects by traversing directories or downloading other user's files.	Moderate Users are curious people; most of which would likely try to traverse directories on the system.	An internal user or partner/collaborator would gain information about other projects and possibly be able to download proprietary data.	Moderate System users may be successful in gaining information unrelated to their use of the service.
S9	FTP users create a DoS by exhausting drive space.	Moderate System users may inadvertently consume all available drive space. Without the use of user quotas, this is somewhat likely to happen.	Moderate Historically, this has not been an issue on the system. As the company's technology refresh and research initiatives accelerate, this issue could become more prominent (due to increased file sharing requirements).	The result of this condition would be a DoS situation. Users would no longer be able to deposit files.	Moderate There is a moderate risk of the system becoming unavailable due to a user exhausting hard drive space.

S10	An attacker compromises the FTP server software.	Low The administrative team selected the vsftpd server for its security features; there are no known security issues with this software package.	Low Access to the FTP service is controlled using hosts.allow (libwrap). A user's system must be preconfigured to use the service.	If an attacker compromises the FTP server software, he or she could gain root access to the server and steal proprietary information.	Low There is a very low risk that an attack could compromise the vsftpd daemon.
S11	An attacker (or user) alters system files without knowledge of the administrator.	Low Only specific users have command line access to the system; therefore, there is a low threat level. FTP server configuration provides an added layer of security; users are chroot jailed to their own home directories.	Moderate Users or administrators could accidentally alter important files on the system. A targeted attack by an insider could also occur.	If files are changed without the administrative teams' knowledge, the system could be compromised, back-doored, or trojaned. This could lead to a loss of proprietary information.	Moderate Since there are command line users of the system, and they could potentially make change to system files, there is exists moderate risk.

Current State of Practice

During my research for this project, I uncovered a wealth of information about how one should configure a Linux system. However, there was a severe lack of actual audit procedures. It seems as if auditing is not the priority of the open source or other online communities. Fortunately, the creation of audit checklists from detailed system build instructions is a natural progression. The only true checklist that I referenced was a checklist about physical security that provided information I used for creating the "Physical Checklist" sections of Assignment 2.

1. Knowledgeleader.com. "Physical Security Audit Checklist." 2003.
URL: <http://www.knowledgeleader.com/iafreewebsite.nsf/content/TechnologyAuditPhysicalSecurityAuditChecklist?OpenDocument> (September 19, 2003)

The rest of the references I used are articles or white papers, where the intended audience is system administrators. They describe system settings and methodologies used to secure Linux systems and the services they provide. I did not directly quote all of the following sources, but they all influenced the style and focus of this audit.

2. Forbes, Liam. "The First Ten Steps to Securing a UNIX Host."
URL:<http://www.arsc.edu/~lforbes/cug/HHPaper.html> (September 19, 2003)
3. Mourani, Gerhard. "Securing and Optimizing Linux: RedHat Edition."
OpenDocs, LLC. 2000. URL: <http://www.tldp.org/LDP/solrhe/Securing-Optimizing-Linux-RH-Edition-v1.3> (September 19, 2003)
4. Bastille Linux: <http://www.bastille-linux.org>
5. Red Hat. "RedHat Linux 8.0: The Official Red Hat Reference Guide."
Red Hat, Inc. 2002 URL:
<http://www.redhat.com/docs/manuals/linux/RHL-8.0-Manual/ref-guide/ch-tripwire.html> (September 19, 2003)
6. SANS Reading Room (<http://rr.sans.org>)
7. SANS Posted Practical Assignments for GIAC Systems and Network Auditor (GSNA) and GIAC Certified Unix Security Administrator (GCUX) – (<http://www.giac.org/cert.php>)
8. Hannett, Dan. "Other BIND Gems." April 4, 2000.
URL:<http://www.freebsdjournal.org/bind-version.php> (September 19, 2003)
9. Sax, Doug. "DNS Spoofing (Malicious Cache Poisoning)." 2000.
URL:http://www.giac.org/practical/gsec/Doug_Sax_GSEC.pdf (September 19, 2003)
10. Hobbit. "The FTP Bounce Attack."
URL:<http://www.geocities.com/SiliconValley/1947/Ftpbounc.htm> (September 19, 2003)
11. Red Hat. "Quota Mini-HOWTO." Red Hat, Inc. August 1997. URL:
<http://www.europe.redhat.com/documentation/mini-HOWTO/Quota-4.php3> (September 19, 2003)
12. Welte, Harald. "IPTables FAQ." August 16, 2002. URL:
<http://www.netfilter.org/documentation/FAQ/netfilter-faq.html> (September 19, 2003)
13. Andaesson, Oskar. "IPTables Tutorial." 2003. URL:<http://iptables-tutorial.frozentux.net/iptables-tutorial.html> (September 19, 2003)
14. Brockmeier, Joe. "Using IPTables." April 2001. URL:
<http://www.unixreview.com/documents/s=1236/urm0104I/0104I.htm> (September 19, 2003)
15. Hobbit. "Netcat 1.10 README." 2003.
URL:http://www.atstake.com/research/tools/network_utilities/nc110.txt (September 19, 2003)
16. Deraison, Renaud. "Nessus Demonstration." 2002.
URL:<http://www.nessus.org/demo/index.html> (September 19, 2003)

© SANS Institute 2003, Author retains full rights.

Assignment 2 – Create an Audit Checklist

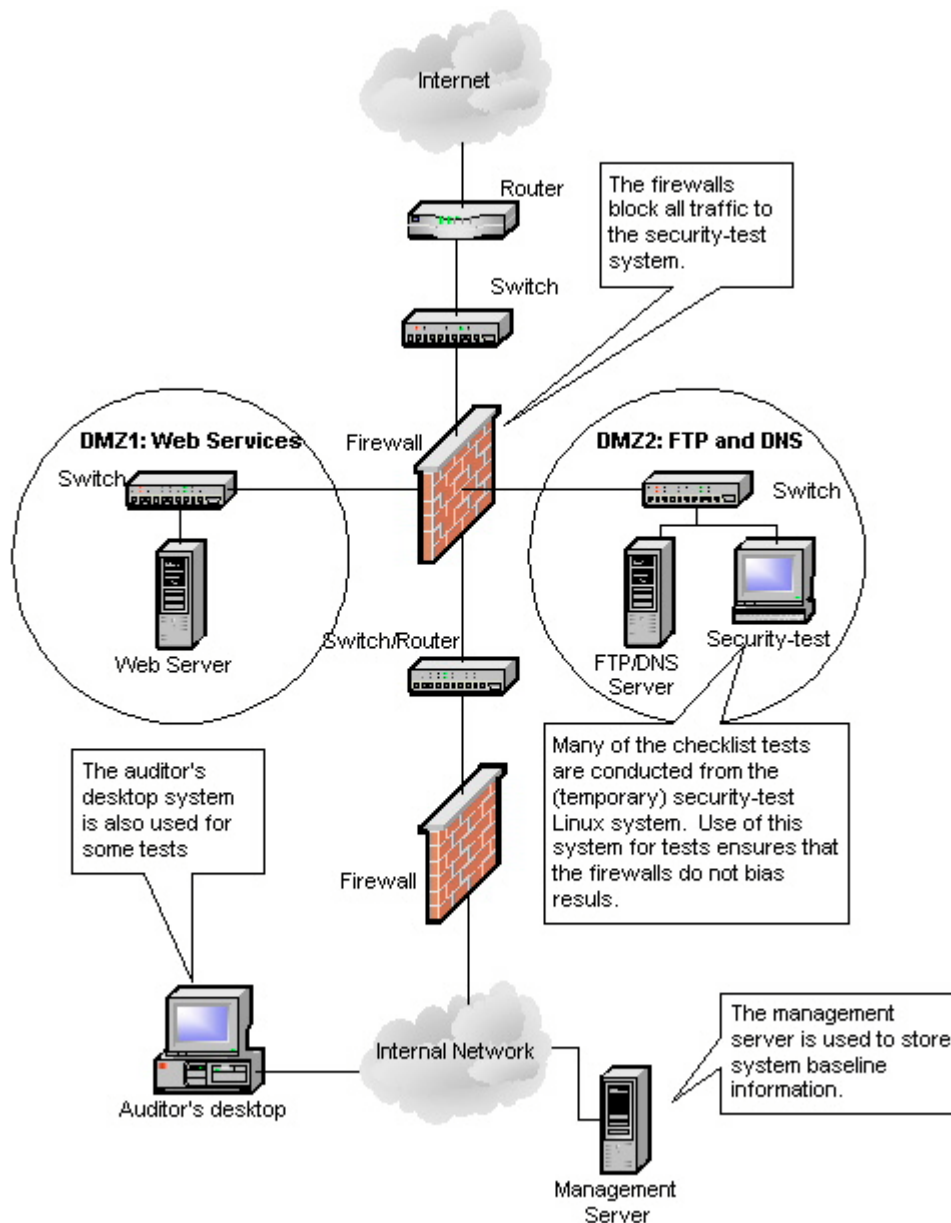
Introduction:

I narrowed the focus of the audit to just the ftp.companyx.com Linux system and the services that it provides. To add cohesion to the entire process, I created separate checklists for each audit area. Each audit checklist contains the steps necessary to determine compliance. Along with the audit steps, I have also provided a description of the audit checklist, the references used to create the checklist, a list of risks that the checklist addresses, the control objectives, the conditions for compliance, and type of judgment utilized (objective or subjective).

I further subdivided the twenty-one checklists contained in this assignment in to physical and system checklists. The physical checklists address the physical security of the system, while the system checklists cover the security of the operating system and installed software. Procedural checklists have not been included in the scope of this assignment. As one of the administrators of the system, I have the inside perspective to know that procedural risks, such as backups and change management, are not critical audit items because Company X already has well established processes to cover them. This, however, may be the topic for future audit at my organization.

Many of the checklist items require the use of a test system connected to the same network as ftp.companyx.com. The utilized system is a temporary system, which is also running RedHat version 9. The test system, referred to as security-test.companyx.com (or security-test), is homed to the DMZ2 network. Any of the tests that originate from this system will only measure the security of the ftp.companyx.com system, since they do not traverse the Company X firewalls. The security-test system is required to have an installed and running FTP server, ssh client, nmap, and Nessus server. In addition, some checklist items require a Company X internal system as the source. In this case, the auditor may use his or her desktop system. System baseline information, such as package listings and daemon configurations, is archived on the administrative team's management server when systems are moved in to production. The management system is located on the company's internal network. The following diagram depicts the auditing environment:

Figure 5 –Audit Environment



Conventions Used:

- In the checklists provided, I have used the `Courier New` font to represent commands and filenames.
- If a command is contained within a sentence, then the command is in quotes.
- If the command is on a new line, I have omitted the quotes.

- Filenames that are contained within a sentence are not enclosed in quotes; this is how the reader can determine the difference between a command and a filename.
- Information that the auditor must provide to complete a command or file entry is contained within <>.
- The references listed for each checklist correspond to the numbered reference provided in assignment one.
- The risks that are provided are directly taken from the tables three and four from assignment one.
- The compliance section of each checklist explains how the system can achieve total compliance. However, each step will list the conditions for success where applicable.
- I have numbered the checklist steps for easy reference in Assignments 3 and 4.
- Each checklist item should be assigned a (P)ass or (F)ail depending on the outcome of the test or program execution. The auditor should consider failed execution of tests or programs on the security-test Linux system as a failure of compliance until the auditor fully investigates them.

Physical Security Checklists

Table 4 – Physical Access Checklist

Physical Checklist 1: Physical Access		
Description	The following is an objective checklist focusing on the physical security and location of the system.	
Reference	1, Personal experience	
Risk	P1: A compromise could occur from the console. The system needs to be physically secure.	
Control Objective	This procedure will check to ensure that physical access to the system is limited to authorized personnel and an updated access log exists.	
Compliance	Compliance is binary. The system is compliant if it passes all of the steps provided.	
Objective/ Subjective	Objective	
Steps for Testing Compliance		
P/F	Step	Description of Step
	1	Is the system located in a datacenter? If the system is not located in the Company X datacenter, the physical security of the system cannot be guaranteed.
	2	Does an actively maintained datacenter access log exist? An access log will provide audit information in the event of a security breach.

3	Does a biometric scanner or lock and key secure the datacenter? In this situation, either method of securing access is acceptable. These will help to ensure that only authorized users can physically access the system.
4	Is there a process in place to verify new access requests? If a process does not exist, the organization will not know who truly requires physically access to the system.
5	Can visitors access the datacenter without supervision? If guest access were not limited, then the system would be at greater risk.

Table 5 – Redundant Power Checklist

Physical Checklist 2: Redundant Power		
Description	This is a checklist of observed items, which provides checks for power redundancy.	
Reference	1	
Risk	P2: Loss of power could cause an extended system outage.	
Control Objective	This procedure is used to ensure that the system is connected to an uninterruptible power supply (UPS) and a generator, which would mitigate the risk listed in P2.	
Compliance	Compliance is binary. The system is compliant if it passes all of the steps provided.	
Objective/ Subjective	Objective	
Steps for Testing Compliance		
P/F	Step	Description of Step
	1	Does the system contain redundant power supplies? Redundant power supplies would provide additional assurance in the event of a power circuit failure within the datacenter.
	2	Have the administrators connected the system power supplies to different circuits?
	3	Are the circuits on a UPS system? A UPS would ensure continuous power in the event of a power outage or brownout.
	4	Does a back up generate provide an alternate power source for the datacenter? A generator would provide power during extended power outages, allowing the organization to continue utilizing the system's services.

Table 6 – Replacement Hardware Checklist

Physical Checklist 3: Replacement Hardware	
Description	This objective checklist is used to verify the availability of replacement hardware.
Reference	Personal experience
Risk	P3: A hardware failure will cause a extended system outage
Control Objective	This procedure will ensure that the administrative team can replicate the system in the shortest possible time if a hardware failure occurs. The audited system is not a

		redundant system, and the SLA specifies that there will be redundant hardware available in the event of a failure.
Compliance		Compliance is binary. The system is compliant if sufficient spare hardware exists. System backups are not required for system restoration; user FTP data also resides on other systems (stipulated in the SLA).
Objective/ Subjective		Objective
Steps for Testing Compliance		
P/F	Step	Description of Step
	1	Do system restoration procedures exist? <ul style="list-style-type: none"> Verify that updated procedures exist. The procedures should include a revision number and a date corresponding to the template build used for the system.
	2	Is replacement equipment available? <ul style="list-style-type: none"> Check stock for a spare, complete, Compaq DL360. Check stock for two spare 36GB hard drives.
	3	Verify that the system is covered by a service contract.

System Checklists

Table 7 – Installed Packages Checklist

System Checklist 1: Installed Packages		
Description		The following objective checklist provides the steps necessary to determine if any additional, unneeded packages are present on the system. This checklist assumes that the system baseline information has been stored on the administrative team's management server, which is standard procedure for the group.
Reference		3, Personal experience
Risk		S1: An attacker could compromise the system. (Operating System)
Control Objective		The administrative team must ensure that the system includes only those packages that are a part of the RedHat 9.0 Linux template by utilizing this procedure. The system could have extraneous packages installed, and therefore does not follow the standard template. This could introduce vulnerable software on the system, software that the administrative team does not monitor.
Compliance		Compliance is binary. The system will pass if it includes only the allowed packages.
Objective/ Subjective		Objective
Steps for Testing Compliance		
P/F	Step	Description of Step

	1	Gain command line access to the system (console or remote login)
	2	Gather package information by executing <code>"rpm -qa > ~/system_packages.txt"</code> .
	3	Create an MD5 checksum of the text file <code>/tmp/system_packages.txt</code> by executing: <code>cd; md5sum system_packages.txt > system_packages.md5</code>
	4	Transfer the files to the administrative team's management server by utilizing FTP from the management server. <ul style="list-style-type: none"> Gain command line access to the management server (<code>management.companyx.com</code>) by either the console or remote access (<code>ssh</code>). Change directory to <code>/home/system_backups/ftp.companyx.com</code>. Download the <code>system_packages.txt</code> and <code>system_packages.md5</code> files from the <code>ftp.companyx.com</code> server by using the FTP protocol.
	5	Compare the MD5 checksum of the <code>system_packages.txt</code> file to the contents of the <code>system_packages.md5</code> file by executing <code>"md5sum system_packages.txt; cat system_packages.md5"</code> . Examine the output to ensure that both checksums are the same. This will ensure that the file was not altered or corrupted during the transfer.
	6	If the file passes the above test, compare the contents of the <code>system_packages.txt</code> file to the contents of the <code>system_packages.ORIG</code> file by executing <code>"diff system_packages.txt system_packages.ORIG"</code> .

Table 8 – Software Patches Checklist

System Checklist 2: Software Patches		
Description		Ensure that all packages are patched to the highest revision level, unless a non-current level is required and the administrative team has accepted all risks. Company X utilizes both a manual process and RedHat's up2date program to manage patches.
Reference		Personal experience
Risk		S1: An attacker could compromise the system. (Operating System)
Control Objective		The system should be at the highest revision on all software packages to mitigate the risks of vulnerabilities. If a software package must remain at a lower version level, then the auditor must determine the level of associated residual risk.
Compliance		Compliance is binary. The system will pass if all packages are at the highest release level, unless the administrative team or management accepts all residual risk.
Objective/ Subjective		Objective
Steps for Testing Compliance		
P/F	Step	Description of Step
	1	Connect to the system by using <code>ssh</code> , and execute <code>"su -"</code> to become the root user.

2	Use the up2date program to check for new package releases by executing “up2date --nox --dry-run”. Results from the program should indicate that no updates are required. If updates are required, check the system documentation for patch waivers, otherwise system fails.
3	Blackbox window manager: http://blackboxwm.sourceforge.net/ . Compare the latest stable release number with that of the version installed on the system. Execute “/usr/local/bin/blackbox -v” to obtain the version of software installed on the system.
4	OpenSSL: http://www.openssl.org . Compare the latest stable release number with that of the version installed on the system. Execute “/usr/local/ssl/bin/openssl version” to obtain the version of software installed on the system.
5	OpenSSH: http://www.openssh.org . Compare the latest stable release number with that of the version installed on the system. Execute both “/usr/local/bin/ssh -V” and “/usr/local/sbin/sshd -V” to obtain the version information from the system.
6	IPlogger by Ojnk Software: http://ojnk.sourceforge.net . Compare the latest stable release number with that of the version installed on the system. Execute “/usr/local/sbin/iplog -version” to obtain the version of the software installed on the system.
7	If version numbers do not match, the system will fail this audit item. The administrative team must update the software, unless they determine that the risk is acceptable.

Table 9 – System Settings Checklist

System Checklist 3: System Settings		
Description	This checklist covers the basic operating system configuration items. These items are defined within the Bastille Linux scripts and the Securing and Optimizing Linux paper. Subsequent checklists address more specific audit areas.	
Reference	2, 3, 4, Personal experience	
Risk	S1: An attacker could compromise the system. (Operating System)	
Control Objective	The objective of this audit item is to ensure that the administrative team has configured the OS correctly.	
Compliance	Compliance is binary. The system is compliant if it passes all of the steps provided.	
Objective/ Subjective	Objective	
Steps for Testing Compliance		
P/F	Step	Description of Step
	1	Check NTP servers listed in /etc/ntp/ntpervers. Only servers present on the permitted NTP server list should be present in the file (since access to generic NTP servers is filtered at the firewall/router).
	2	Ensure that the system follows the standardized partition layout using the command “fdisk

		<pre>-l /dev/ida/c0d0": /dev/ida/c0d0p1 * 1 614 2505104 83 Linux /dev/ida/c0d0p2 615 1228 2505120 83 Linux /dev/ida/c0d0p3 1229 1474 1003680 82 Linux swap /dev/ida/c0d0p4 1475 8716 29547360 5 Extended /dev/ida/c0d0p5 1475 2455 4002464 83 Linux /dev/ida/c0d0p6 2456 8716 25544864 83 Linux</pre>
	3	<p>Ensure that the system is using the proper mounts for the above partitions by checking the <code>/etc/fstab</code> file and the output from <code>"df -k"</code>:</p> <pre>/dev/ida/c0d0p1 / ext3 defaults 1 1 /dev/ida/c0d0p5 /var ext3 defaults 1 2 /dev/ida/c0d0p6 /home ext3 defaults 1 2</pre>
	4	Verify that the <code>/etc/issue</code> and <code>/etc/issue.net</code> files contain the proper legal notice.
	5	Ensure that a reboot to multi-user mode requires a root password login. The file <code>/etc/inittab</code> should contain the following line: <code>"~:S:wait:/sbin/sulogin"</code> .
	6	Reboot the system to single user mode by executing <code>"sync;sync;init 0"</code> to verify the setting described in step 5.
	7	<p>Verify that the <code>/etc/profile</code> file contains following inactivity setting:</p> <pre>TMOUT=900 # Root logout after inactivity (seconds)</pre>
	8	Further verify the timeout value by logging in to the <code>ftp.companyx.com</code> system and becoming the root user by executing <code>"su -"</code> . Execute the <code>"date"</code> command and allow the connection to idle. Once the connection has timed-out, execute the <code>"date"</code> command and calculate the idle duration. The system is compliant if the idle time is fifteen minutes.
	9	Inspect the <code>/etc/cron.allow</code> file to ensure that only the root user can execute command through cron.
	10	<p>Test the crontab restriction by attempting to execute a program from the auditor's account using cron. As the auditor's account, execute the <code>"crontab -e"</code> command to attempt to alter the cron settings. The system is compliant is the following error is received:</p> <pre>You (<username>) are not allowed to use this program (crontab) See crontab(1) for more information</pre>
	11	Ensure that the system is configured properly for DNS resolution from hosts first, and DNS servers second. The <code>/etc/nsswitch.conf</code> file should contain the line: <code>"hosts: files dns"</code> .
	12	Verify that the hostname is configured correctly in the <code>/etc/HOSTNAME</code> file.
	13	<p>Verify that only the proper shells are listed in the <code>/etc/shells</code> file. This will ensure that users are only able to be assigned (and execute) the verified shells. The list should include:</p> <pre>/bin/sh /bin/bash /sbin/nologin</pre>

		<pre> /bin/bash2 /bin/ash /bin/bsh /bin/tcsh /bin/csh /bin/ksh /bin/null /bin/false </pre>
14		<p>Ensure that minimum requirements for user passwords are enforced. The <code>/etc/login.defs</code> file should contain the following lines, which define the password minimum length, and minimum automatic selection for UID and GID for use with the <code>useradd</code> command:</p> <pre> PASS_MIN_LEN 8 UID_MIN 4500 GID_MIN 4500 </pre>
15		<p>Ensure that the system enforces complexity and minimum length for user passwords. The auditor can verify this by changing his or her account password. Attempt to change the password to a dictionary word, and a short password. The system is compliant if the <code>passwd</code> program does not allow the password to be changed.</p>
16		<p>Verify that only members of the “wheel” group can “su” to the root account. The <code>/etc/pamd/su</code> file must contain the following configuration line:</p> <pre> auth required /lib/security/\$ISA/pam_wheel.so trust use_uid </pre>
17		<p>Further verify the requirement of wheel group membership to utilize the “su” command by becoming the root user and executing “su <username>”, where username is an arbitrary username from the <code>/etc/passwd</code> file, and attempting to “su -”. The system will be compliant if the arbitrary username (that is not part of the wheel group) cannot successfully “su” to the root account.</p>
18		<p>Verify that the root user can only log in to the system via the console or serial connection. This will ensure that a user or attacker cannot attempt to directly log in to the system as root over a network connection (su would be required). The <code>/etc/securetty</code> file should contain the following lines for virtual consoles and serial connections:</p> <pre> console vc/1 vc/2 vc/3 vc/4 vc/5 vc/6 vc/7 vc/8 </pre>

		vc/9 vc/10 vc/11 tty1 tty2 tty3 tty4 tty5 tty6 tty7 tty8 tty9 tty10 tty11
	19	Test remote root logins by attempting to telnet to the system from the auditor's desktop system. When prompted for the username, enter root. The system is compliant if root login is not permitted.
	20	Verify that spoof protection is enabled in the <code>/etc/host.conf</code> file. The following entries ensure that the local resolv+ library performs reverse and forward resolution of hosts, and that possible spoof attempts are logged using the syslog facility: nospoof on spoofalert on

Table 10 – Sendmail Settings Checklist

System Checklist 4: Sendmail Settings		
Description		This objective checklist will ensure the security of the sendmail daemon. RedHat version 9 requires the sendmail daemon to deliver email to local system users. Without it, the root user would not receive email.
Reference		3, Personal experience
Risk		S1: An attacker could compromise the system. (Operating System)
Control Objective		The objective of this step is to test the sendmail implementation on the system for vulnerabilities. In addition, the sendmail daemon should only be available over the loopback interface.
Compliance		Compliance is binary. The system is compliant if it passes all steps.
Objective/ Subjective		Objective
Steps for Testing Compliance		
P/F	Step	Description of Step

1	Log in to the ftp.companyx.com using ssh, and become root by executing “su -“.
2	New in RedHat 9 is the requirement of sendmail as a daemon for local email delivery. Ensure that the sendmail daemon is configured. The following settings should be included in the /etc/sysconfig/sendmail file: DAEMON=yes QUEUE=15m
3	Verify that sendmail will not disclose system user accounts. The following option should be included in the /etc/mail/sendmail.cf file: “PrivaryOptions=goaway”.
4	Ensure that users cannot verify usernames by using the sendmail daemon. To connect to the sendmail daemon, execute “telnet localhost 25” from the ftp.companyx.com system. Enter “helo ftp.companyx.com” to initiate the conversation. Attempt to use the VRFY command to guess user accounts on the system, enter “vrfy <username>” where <username> is a valid user from the /etc/passwd file. The system is compliant if the daemon displays an error.
5	To protect the sendmail daemon from network traffic (beyond that provided by firewalls and screening routers), libwrap is used to control access. Verify that the /etc/hosts.allow only permits access from the localhost over the loopback interface. The entry should resemble the following: sendmail: 127.0.0.1: ALLOW
6	Test access to the sendmail daemon from the security-test Linux system. The daemon should not be accessible from outside of the system. Execute a telnet to the IP address of the ftp.companyx.com system on port 25. If the sendmail daemon responds, the system has failed this step.

Table 11 – Kernel Vulnerability Checklist

System Checklist 5: Kernel Vulnerabilities and Settings		
Description		This objective checklist ensures that the kernel is not vulnerable to any known exploits.
Reference		3, Personal experience
Risk		S1: An attacker could compromise the system. (Operating System)
Control Objective		This audit item will enable the auditor to verify that the currently used Linux kernel does not contain any known vulnerabilities. The kernel is not required to be at the highest available version, but it must be free of known vulnerabilities.
Compliance		Compliance is binary. The system is compliant if the current Linux kernel is free of vulnerabilities, and the administrative team has properly configured all kernel options.
Objective/ Subjective		Objective
Steps for Testing Compliance		
P/F	Step	Description of Step

1	Connect to the ftp.companyx.com server using ssh and become the root user by executing “su -“.
2	Execute the “uname -r” command to obtain the currently used kernel version.
3	Utilize the up2date program to determine if RedHat has a new kernel source available in RPM format. The command to use is “up2date -nox -dry-run”. The administrative team does not allow up2date to upgrade the kernel, so the results reside in the “skipped” section of the output.
4	Compare the results from steps two and three with version information found on http://www.kernel.org . The system can still be ultimately compliant with the control objectives if it passes step five.
5	Verify on http://www.redhat.com , http://www.securityfocus.com and http://www.cert.org that the current compiled kernel version on ftp.company.com does not contain a security vulnerability. If a known vulnerability exists in the currently used kernel version, the system is not compliant.
6	<p>Verify all kernel options. This step will ensure that important kernel options, like loadable kernel modules, are correctly configured (in this case, disabled). Compare the /usr/src/linux-2.4/.config to the baseline version saved on the management.companyx.com system. The baseline version of the .config is stored in the /home/system-backups/ftp.companyx.com/kernel-options.ORIG file.</p> <ul style="list-style-type: none"> ○ Store a copy of the .config file, named as kernel-options.txt, in the auditor's home directory. ○ Create an MD5 checksum of the kernel-options.txt file with the command “md5sum ~/kernel-options.txt > ~/kernel-options.md5”. ○ On the management system, compare the MD5 checksum of the kernel-options.ORIG file to the contents of the kernel-options.ORIG.md5 file by executing “md5sum kernel-options.ORIG; cat kernel-options.ORIG.md5”. ○ Download the kernel-options.txt and kernel-options.md5 from the ftp.companyx.com server, and compare their MD5 checksums with the command “md5sum kernel-options.txt; cat kernel-options.md5”. ○ Compare the two configuration files using the diff command: “diff kernel-options.ORIG kernel-options.txt”. ○ The system will be compliant if the two files are identical.
7	<p>Manually inspect the system for loadable kernel modules by executing “lsmod” on the ftp.companyx.com system. The output from the command should be:</p> <pre>lsmod: QM MODULES: Function not implemented</pre>

Table 12 – Network Settings Checklist

System Checklist 6: Network Settings	
Description	This objective checklist tests the security of the system's network configuration. Each checklist step provides a network setting, and, if available, a particular test method for validation.

Reference	3, All checklist items are from Optimizing and Securing Linux.	
Risk	S1: An attacker could compromise the system. (Operating System). S2: The system is subjected to a denial-of-service (DoS) attack. S3: The system reveals too much information during hacker reconnaissance.	
Control Objective	The purpose of this checklist is to ensure that the administrative team has properly configured the network settings, and that the network settings are behaving as desired. The network settings will also help mitigate the risks associated with DoS attacks and hacker reconnaissance.	
Compliance	Compliance is binary. The system is compliant if it passes all of the provided steps; all of the relevant network settings must be configured, and the tests provided must be passed.	
Objective/ Subjective	Objective	
Steps for Testing Compliance		
P/F	Step	Description of Step
	1	All options listed in this checklist are set in the /etc/sysctl.conf file; verify their existence in this file on ftp.companyx.com system. This particular option will cause the tcp/ip stack to ignore ICMP echo messages, which will stealth the system and keep attackers from using it as a DoS amplifier. net.ipv4.icmp_echo_ignore_all = 1 Test this setting by attempting to ping ftp.companyx.com from the security-test Linux system. The system is compliant if it does not respond to the ping.
		The following option restricts responses to broadcast pings. It is used for the same purpose as the option in step one. net.ipv4.icmp_echo_ignore_broadcasts = 1 Test this setting by pinging the network broadcast address from the security-test Linux system. The system is compliant if it does not respond to the ping.
		The following option restricts the acceptance of the packets that include IP source routing options. net.ipv4.conf.all.accept_source_route = 0
		The following option prevents SYN floods from consuming network resources. net.ipv4.tcp_syncookies = 1 Test this setting by executing a DoS attack using the Nessus tool. See “System 8 Checklist” details on how to configure Nessus for the DoS attack. This system will pass this step if the Nessus tool does not detect a DoS vulnerability.
		This option restricts the acceptance of ICMP redirects. ICMP redirects could cause the system to alter its routing table, and route traffic through another system. net.ipv4.conf.all.accept_redirects = 0
		The following option enforces fragmentation protections on reassembly of packets, which would prevent attacks based on fragment overlaps or purposefully fragmented exploits. net.ipv4.ip always defrag = 1

	<p>According to Gerard Mourani, this option will configure the system ignore all bad error messages on the network:</p> <pre>net.ipv4.icmp_ignore_bogus_error_responses = 1</pre>
	<p>With the following option set, illegal packets such as spoofs, source routing, and redirected packets are logged to the syslog facility:</p> <pre>net.ipv4.conf.all.log_martians = 1</pre>
	<p>The following option sets the range of source ports used for local tcp connections.</p> <pre>net.ipv4.ip_local_port_range = 32768 61000</pre> <p>The auditor may test this setting by initiating several connections to the security-test system from the ftp.companyx.com system. Initiate several telnet sessions, once logged in to the security-test system, use the “netstat -an” command (on either system) to verify the source ports for the connections.</p>
	<p>The following list of tcp/ip options is for tuning purposes. They will enable the system to reap a greater amount of inactive connections, and handle more connections per time (Mourani). The auditor should verify their presence in the /etc/sysctl.conf file:</p> <pre># Decrease the time default value for tcp_fin_timeout connection net.ipv4.tcp_fin_timeout = 30 # Decrease the time default value for tcp_keepalive_time connection net.ipv4.tcp_keepalive_time = 1800 # Turn off the tcp_window_scaling net.ipv4.tcp_window_scaling = 0 # Turn off the tcp_sack net.ipv4.tcp_sack = 0 # Turn off the tcp_timestamps net.ipv4.tcp_timestamps = 0</pre>

Table 13 – Daemon and Open Ports Checklist

System 7 Checklist: Daemons and Open Ports		
Description		The system should not be running unknown daemons. The auditor should be able to identify all open ports on the system.
Reference		Personal experience, system man pages
Risk		S1: The system may be compromised (Operating System)
Control Objective		Verify that the system is not running unnecessary daemons or open network ports.
Compliance		Compliance is binary. The system will pass if the results of the test match the baseline configuration stored on the management server.
Objective/ Subjective		Objective
Steps for Testing Compliance		
P/F	Step	Description of Step

1	Connect to the system by using ssh, and execute “su -“ to become the root user.
2	Change director to the auditor’s home directory, from which you can download the results,
3	Collect the open ports using the following command: “netstat -tap > system_ports.txt; echo >> system_ports.txt; netstat -uap >> system_ports.txt; echo >> system_ports.txt; netstat -an >>system_ports.txt”. Create a MD5 checksum of the file by executing “md5sum system_ports.txt > system_ports.md5”.
4	Download the results from step 3 to the management station. A checksum of the system_ports.txt file should be compared to the contents of the system_ports.md5 file to ensure that the file was not changed during transmission. If the file has not been altered, compare its contents with that of the system_ports.ORIG baseline file previously stored on the system (at production deployment). The comparison must be visual as information about open connections (like the auditor’s ssh session) and random bindings, like the sig_fam daemon, may be captured in the system_ports.txt file. The Xinetd daemon starts the sig_fam daemon, but it is bound to a random port. The sig_fam daemon, used by the system to monitor file changes, it is enabled by default.
5	Collect the system startup information using the following command: “chkconfig --list > system_chkconfig.txt”. Create an md5checksum of the file, and download both files to the management system. Verify that the system_chkconfig.txt file is unaltered by comparing its checksum with the contents of the downloaded checksum file. Also, compare the checksum of the baseline file system_chkconfig.ORIG with the previously stored MD5 checksum system_chkconfig.ORIG.md5 in the same manor. Utilize the command “diff system_chkconfig.txt system_chkconfig.ORIG” to compare the contents of both files. This will ensure that the system is only running pre-approved daemons.
6	Visually check the contents of the “rc” directories to ensure that only the standard daemons are enabled. Inspect the following directories, which are located under /etc/rc.d/ to make sure that the files correspond to the contents of the system_chkconfig.txt file generated in step five: init.d, rc0.d, rc1.d, rc2.d, rc3.d, rc4.d, rc5.d, rc6.d

Table 14 – DoS Checklist

System 8 Checklist: DoS	
Description	The following objective checklist will enable the auditor to test the system for susceptibility to DoS attacks using Nessus.
Reference	16, Personal experience
Risk	S2: The system could be subjected to a denial-of-service (DoS) attack.
Control Objective	This step will ensure that system is not susceptibility to DoS attacks. The Nessus tool includes a wide array of plug-ins for testing DoS attacks.
Compliance	Compliance is binary. The system will be compliant upon passing a Nessus scan for DoS vulnerabilities.
Objective/	Objective

Subjective		
Steps for Testing Compliance		
P/F	Step	Description of Step
	1	Log in to the security-test Linux system as root. Start the Nessus daemon by executing <code>"/usr/local/sbin/nessusd -D"</code> . Start the Nessus client by executing <code>"/usr/local/bin/nessus"</code> . Log in to the client using a previously created Nessus user account that has privileges to test.
	2	Configure Nessus to scan for DoS vulnerabilities by navigating to the Plugins -> Denial Of Service plugin selection. Ensure that all options are checked. Click on Target Selection and specify the IP address of the ftp.companyx.com server, and select "Start the scan" to begin the test. The system is compliant if the Nessus reports no vulnerabilities.
	3	The Xinetd service is a replacement for the inetd service. It is used to provide access control, along with libwrap (hosts.allow, hosts.deny), for telnet, FTP, and OpenSSH. Verify that Xinetd is configured to provide fifty simultaneous connections, where twenty-five can originate from the same host. These settings are useful for preventing a DoS attack. The following lines should be present in the ftp.companyx.com /etc/xinted.conf file: <pre>instances = 50 log_on_success = HOST PID DURATION log_on_failure = HOST USERID per_source = 25</pre>

Table 15 – System Reconnaissance Checklist

System 9 Checklist: System Reconnaissance		
Description		The auditor will utilize this subjective checklist to verify that there are no information leaks such as version numbers or OS type during hacker reconnaissance.
Reference		8, Personal experience
Risk		S3: The system may reveal too much information during hacker reconnaissance.
Control Objective		These checklist steps will determine if the system is providing any useful information about the operating system or versions of software.
Compliance		This checklist is subjective. It is at the discretion of the auditor to determine if the information gathered is useful enough for targeted attacks. If the information gathered is not useful, then the system is compliant.
Objective/ Subjective		Subjective
Steps for Testing Compliance		
P/F	Step	Description of Step
	1	From the security-test Linux system, retrieve banner information for both telnet and FTP. Banner information should not reveal any information about the OS of the system or version of the software.
	2	From a system within Company X, retrieve banner information for both telnet and FTP. The banners shown for hosts present in the /etc/hosts.allow may be different from unknown hosts. Banner information should not reveal any information about the OS of the system or the version of the software.
	3	From the security-test Linux system, query the DNS server for version information by issuing the command “dig @ftp.companyx.com version.bin chaos txt”. Obtain the version information for the package installed by querying the RPM database with the command “rpm -qa grep bind”. The two version numbers should not match.
	4	Utilize nmap to attempt to guess the OS type of the system. From the security-test system, execute the command “nmap -O ftp.companyx.com”. Check fingerprint information for information listed for the system. If the system type is obfuscated, than this step is passed.

Table 16 – Syslog Checklist

System 10 Checklist: Syslog Audit	
Description	The syslog service gathers access control list (ACL) logging from Company X's internet router. The system stores the messages for later use during trouble shooting activities.
Reference	3, Personal experience

Risk		S4: The syslog logging service could be spoofed, or used as a DoS mechanism
Control Objective		This test will ensure that the system's syslog service is not susceptible to a DoS attack, and cannot receive spoofed messages. In addition, the auditor will verify that the syslog service is receiving messages from the internet router and from the named daemon (which is in a chroot environment).
Compliance		Compliance is binary. The system will be compliant if it passes all the steps provided.
Objective/ Subjective		Objective
Steps for Testing Compliance		
P/F	Step	Description of Step
	1	Obtain the syslog-poison.c code (Gamma '98) and compile on the security-test Linux system. Download the file from http://content.443.ch/pub/linfiles/Gnusoftware/spoofcode/syslog-poison.c . Compile the code using "gcc syslog-poison.c -o syslog-poison".
	2	Generate a spoofed syslog message that uses the Company X internet router's IP address as the source. To do this, execute "syslog-poison ir.companyx.com ftp.companyx.com "SPOOFED"". Examine the /var/log/messages file for evidence of the spoofed syslog alert: "grep SPOOFED /var/log/messages". This system passes this step if the alert is not logged.
	3	Verify that the BIND chroot environment can log to the syslog. The following option must be found in the /etc/sysconfig/syslog file: SYSLOGD_OPTIONS="-m 0 -r -a /var/named/dev/log" Ensure that named is logging properly to the /var/log/messages file by examining messages after a daemon restart. Restart the daemon as the root user by executing "/etc/rc.d/init.d/named restart". Immediately examine the /var/log/messages file for new entries from the named daemon. This step is passed if the daemon is logging properly.
	4	Examine the /etc/syslog.conf file on ftp.companyx.com. Ensure that logging from the Cisco router is directed to the /var/log/ciscolog file. The option should be: local3.debug /var/log/ciscolog Verify that the syslog daemon is capturing new alert entries in the ciscolog file by examining new input to the file. Execute the following to verify: tail -f /var/log/ciscolog The system will pass this step if new entries are written to the log.

Table 17 – Named Configuration Checklist

System 11 Checklist: Named Configuration	
Description	DNS is an important function of this system. The following objective checklist will verify its proper configuration.
Reference	9, dig man page, personal experience

Risk		S5: DNS server is susceptible to cache poisoning. S6: An attacker compromises the DNS server software and alters zones files.
Control Objective		The objective of this checklist is to ensure that the system is not susceptible to DNS cache poisoning, and verify that the administrative team has properly configured the named daemon.
Compliance		Compliance is binary. The system is compliant if it passes all steps provided in the checklist. The terms for compliance for each step are listed in the body of the checklist.
Objective/ Subjective		Objective
Steps for Testing Compliance		
P/F	Step	Description of Step
	1	Log in to the system using ssh, become the root user by executing “su -“, and obtain the BIND version number with the command “named -v”. Compare the version number to that of the latest release at http://www.isc.org . This system will pass this step if the version numbers are the same.
	2	Ensure that the named daemon is running within the chroot environment. The /etc/sysconfig/named file should have the following variable definition: “ROOTDIR=/home/named” The directory /var/named should be a symbolic link of the /home/named directory. This can be checked by using the “ls -al /var/named” command. The actual files reside in the /home partition because it is, by far, the largest partition on the system. The auditor can verify which daemon is currently being executed by running the command “ps -aux grep named”. The system will pass this step if the results are “/home/named/usr/sbin/named -u named -t /home/named”.
	3	Verify that zone transfers are set to “restricted” in the /var/named/var/etc/named.conf file. Each DNS zone should have an “allow-transfer” section with the addresses of Company X’s internet provider DNS servers listed.
	4	Ensure that restricted zone transfers are enabled by attempting to transfer zones using the security-test Linux system as the source. Log in to the security-test system, and utilize dig to transfer several DNS zones. The following command should be used twice: “dig -t AXFR @ftp.companyx.com <zone>”, where <zone> is first the forward zone, and second is the reverse zone. The results should be “Transfer failed.”
	5	Ensure that restricted DNS queries are enabled by examining the “allow-query” section in the /var/named/var/etc/named.conf file. This block should include Company X’s external hosts (that require recursive lookups) and Company X’s internet provider DNS servers.
	6	Verify that restricted DNS queries are functioning properly by attempting to resolve DNS queries from the security-test Linux system. Log in to the security-test system, and execute the following command: “dig @ftp.companyx.com www.companyx.com”. If DNS queries are restricted, the result should be a “status: REFUSED” message.

7	<p>Ensure that query identification numbers (ids) are arbitrary by executing 35 queries in succession. Log in to the ftp.companyx.com system and execute the following script:</p> <pre>#!/usr/bin/perl \$iterations = 35; for (\$i = 0; \$i < \$iterations; \$i++) { (\$id) = (`dig \@ftp\.companyx\.com www\.netscape\.com` =~ /id: \d+/g); print "\$id\n"; }</pre> <p>Inspect the output from the above script to ensure that the id numbers are non-sequential.</p>
---	--

Table 18 – FTP Bounce Checklist

System 12 Checklist: FTP Bounce Attack		
Description	This checklist will test the system for the FTP-Bounce attack vulnerability.	
Reference	10, steps one through five are derived from Hobbit's paper on the FTP bounce attack.	
Risk	S8: The FTP server may be susceptible to the FTP-Bounce attack.	
Control Objective	The following steps will test for the FTP-Bounce vulnerability, which will curb an attacker's ability to perform network reconnaissance.	
Compliance	Compliance is binary. The system will be compliant if it is not vulnerable to this attack.	
Objective/ Subjective	Objective	
Steps for Testing Compliance		
P/F	Step	Description of Step
	1	<p>Connect to the ftp.companyx.com FTP server from a machine that is permitted access (in the libwrap's /etc/hosts.allow file, like the auditor's workstation), and issue the following commands:</p> <pre>quote "pasv" quote "stor foobar"</pre> <p>The first command will display port information about the current connection in the form of F,F,F,X,X (where F,F,F,F is the IP address of the server, and X,X is the port information). The second command will hang the ftp session long enough to complete the following steps. Continue through step five to complete the manual test for this vulnerability.</p>
	2	<p>On the source system, create a file named test.txt using the information gathered in step one and the below lines. The user and <password> information should be the information used to log in to the security-test Linux system. A file named data (which can contain anything) must exist in the auditor's home directory on the security-test system.</p> <pre>user ftp-test</pre>

		<pre> pass <password> type i port F,F,F,F,X,X retr data quit </pre>
3		<p>On the source system, execute the following ksh script to generate a file with 60kb of null data.</p> <pre> #!/bin/ksh numb=0 while [\$numb -lt "250"] do numa=0 while [\$numa -lt "250"] do `echo -n '\000' >> null.txt` numa=`expr 1 + \$numa` done `echo -n '\n' >> null.txt` numb=`expr 1 + \$numb` done </pre>
4		Concatenate the two files from steps two and three: <code>cat test.txt null.txt > instrs</code> .
5		<p>In the FTP session from step one, enter the following commands:</p> <pre> put instrs quote "port C,C,C,C,0,21" quote "retr instrs" </pre> <p>The "C,C,C,C,0,21" is the IP address and port numbers of the test-security system. The expected result would be a failure message. The system is vulnerable if the file transfer begins.</p>
6		<p>Compliment the above tests with a Nessus scan from the security-test Linux system. The scan should focus on this particular vulnerability. Log in to the security-test system as root and start the Nessus daemon by executing <code>/usr/local/sbin/nessusd -D</code>. Start the Nessus client by executing <code>/usr/local/bin/nessus</code>. Log in to the client using a previously created Nessus user account that has privileges to test. Configure a scan that only includes the FTP vulnerabilities by navigating to the "Plugins" tab, selecting "Disable All", and checking the box for the FTP plugins. Specify the target under the "Target selection" tab, and select "Start the scan" to begin the Nessus scan. The system is compliant if the scan detects no vulnerabilities.</p>

Table 19 – FTP Configuration Checklist

System 13 Checklist: FTP Configuration	
Description	This checklist will ensure that an FTP user cannot access information that is not a part of their individual project. This, however, is not an audit of the current system users.

Reference	Personal experience	
Risk	S9: Ftp users may glean information from the system, or access other user's data.	
Control Objective	The following steps will check if users can obtain information about other users or other projects by using FTP.	
Compliance	Compliance is binary. The system will be compliant is no sensitive information is gathered.	
Objective/ Subjective	Subjective	
Steps for Testing Compliance		
P/F	Step	Description of Step
	1	Connect to the ftp.companyx.com FTP service from a machine that is not permitted access. Examine the error message displayed for information regarding versions or software packages. The system will pass this step if the information returned is non-specific.
	2	Log in to the ftp.companyx.com system using ssh and execute “su -” to become the root user. Create a test user account on the system using the following procedures: <ul style="list-style-type: none">o mkdir /home/ftp/users/testo mkdir /home/ftp/users/test/ftp-test1o mkdir /home/ftp/users/test/ftp-testo Edit the /etc/group and add a group entry for security-test with gid 3050o useradd -u 5050 -g 3050 -c “Security Test” -s /bin/false -d /home/ftp/users/test/./ftp-test ftp-testo chown root:test /home/ftp/users/test/ftp-test1o passwd ftp-test Proceed through step four to complete this part of the test.
	3	Log in to the ftp.companyx.com FTP server using the ftp-test account created in step two. Examine the log in process for information that may reveal the FTP server version.
	4	Attempt to traverse directories and glean information from the system. Utilize the following tests: <ul style="list-style-type: none">o Examine file and directory listings for user ids and group ids. The vsftpd daemon should rewrite all uids and gids as the user and group ftp.o Execute a “pwd” to find out what the root directory is for the ftp-test user.o Change the working directory to the top most directory, and try to change directory beyond the top directory. To do this, execute three “cd ..” commands. Execute a “pwd” and an “ls” to determine the current working directory. Attempt to change directory to /test and /home/ftp/users.

5	Manually examine the <code>/etc/passwd</code> file on the <code>ftp.companyx.com</code> system to ensure that all user accounts are created with the <code>chroot</code> jail option listed in the home directory. (e.g. <code>baumansc:x:4002:3000:Sean Baumann</code> <code>x3342:/home/ftp/users/./security/baumansc:/bin/ksh</code>)
6	Manually examine the permissions for subdirectories of the <code>/home/ftp/users</code> directory. Subdirectories should not contain any “other” permission. Execute the command “ <code>find /home/ftp/users/ -perm +o=rwx -type d -print</code> ”. The auditor can ignore the <code>/home/ftp/users</code> and the <code>/home/ftp/users/lost+found</code> directories for this test. These directories require “other” permissions. The system is compliant if all subdirectories have no permissions for the “others”.
7	Once the audit process is complete, remove the directories and user created in step two (clean-up).

Table 20 – User Quota Checklist

System 14 Checklist: User Quotas		
Description	This checklist will ensure that any one user cannot exhaust the system's hard drive space.	
Reference	11, personal experience	
Risk	S10: A user may create a DoS by exhausting drive space.	
Control Objective	This purpose of this check is to ensure that the system will be available if a user exhausts free drive space. In addition, one user should not be able to consume the entire (more than 75%) /home partition.	
Compliance	Compliance is binary. If a user account cannot exhaust all drive space, then the system passes.	
Objective/ Subjective	Objective	
Steps for Testing Compliance		
P/F	Step	Description of Step
	1	Log in to the ftp.companyx.com system from a system with the proper hosts.allow permissions; use the auditor's own ftp account. Transfer large files to the system, while monitoring the disk usage.
	2	Attempt to exhaust disk space by transferring large files to the system. If the /home file system becomes exhausted by one user, then the system fails this checklist item.
	3	If the /home file system can be completely filled, ensure that the system remains functional. <ul style="list-style-type: none">○ Attempt to log in to the system using ssh.○ Attempt to remove files.

4	Log in to the ftp.companyx.com system, and execute “su -“ to become the root account. Examine the kernel settings for user quota support. View the /usr/src/kernel-2.4/.config file and search for “CONFIG_QUOTA”. If this option is enabled in the kernel, the option will be set to “y.” This system passes this step if this setting is enabled.
5	Execute “rpm -qa grep quota” to determine if quota software has been installed on the system. If the software is present on the system, then the system passes this step.
6	Examine the /etc/fstab file for the “usrquota” option for the /home file system.
7	To ensure that quotas are utilized, examine the output from the “repquota” command.
8	Select a random sampling of users from the /etc/passwd file. Utilize the “edquota” command to determine if they have been set up with quotas. The system is compliant if quotas are in use for all users of the random sampling.

Table 21 – File Integrity Checklist

System 15 Checklist: System File Integrity		
Description		Company X utilizes the tripwire software to monitor file changes on the system. This objective checklist will ensure that the software is functioning properly.
Reference		5
Risk		S11: An attacker (or user) alters system files without the knowledge of the administrator.
Control Objective		The objective of this checklist is to ensure that the tripwire software is functioning properly on the system. It should detect and report changes to the monitored files. Updates to the tripwire database should also be possible.
Compliance		Compliance is binary. The system is compliant if the software reports all changes to the file system, and the administrative team reviews the tripwire reports on a regular basis.
Objective/ Subjective		Objective
Steps for Testing Compliance		
P/F	Step	Description of Step
	1	Log in to the ftp.companyx.com system and become the root user by executing “su -“. Verify that the tripwire software is installed by executing “rpm -qa grep tripwire”.
	2	Examine and visually compare the /etc/twpol.txt file with the baseline copy saved on the management system. The file is short and organized enough to inspect this manually.
	3	Reinitialize the tripwire database to keep extraneous data from interfering with results. Execute “/usr/sbin/tripwire -init”. Utilize the local passphrase to begin the initialization. If the

		initialization is completed successful, the system passes this step.
4		<p>Alter several files on the system and rerun the integrity check. These changes will ensure that multiple functions of the tripwire software are reliable.</p> <ul style="list-style-type: none"> o “touch” the /etc/hosts file o Create the directory /home/test o Rotate the log files in /var/log by executing “logrotate -f /etc/logrotate.conf”. o Rerun the integrity check by executing “/usr/sbin/tripwire --check”. <p>Identify the changes in the generated report. The system has passed this step if the report indicates that the above changes were made.</p>
5		Update the tripwire database using the report generated in step four. Execute the command “/usr/sbin/tripwire --update --twrfile /var/lib/tripwire/report/<file>.twr”, where <file> is the name of the report generated in step four. If the procedure has correctly updated the database, then the system has passed this step.
6		Remove the /home/test directory and reinitialize the database with the command “/usr/sbin/tripwire --init”
7		Examine the root user’s email box to determine if the administrative team has processed, and acted upon, the previously generated tripwire reports. If the reports are unread, then this step is failed, and processes and procedures should be altered to include tripwire monitoring.

Table 22 – Administrative Access Checklist

System 16 Checklist: Administrative Access	
Description	This checklist provides the steps necessary for verifying the security of the administrative access process. Currently, the administrative team utilizes OpenSSH for remote access to the system. Unfortunately, telnet access to the system is still required for legacy reasons.
Reference	personal experience
Risk	S1: An attacker could compromise the system.
Control Objective	The objective of this process is to ensure that the version of OpenSSH being used is not vulnerable. Using this checklist, the auditor can also determine if the administrative team is utilizing the proper method of gaining remote access to the system when becoming the root user. Typing the root user password in a clear text telnet session is undesirable.
Compliance	Compliance is binary. While this checklist does include an interview question, it is still satisfied with a yes or no answer.
Objective/ Subjective	Objective
Steps for Testing Compliance	

P/F	Step	Description of Step
	1	Ensure that OpenSSH is running as a daemon on the system. Execute “ <code>chkconfig --list</code> ”, and examine the output for “ssh.” The auditor has already tested for version compliance in the System 2 checklist.
	2	Verify that the administrative team uses ssh for remote access by examining the <code>/var/log/messages</code> file for sshd connect entries. Correlate these entries to the results of a “last” command.
	3	<p>Verify that the OpenSSH daemon options set as follows in the <code>/usr/local/etc/sshd_config</code> file:</p> <pre> Port 22 Protocol 2 ListenAddress 63.89.199.70 ServerKeyBits 1024 PermitRootLogin no StrictModes yes PasswordAuthentication yes PubKeyAuthentication yes PermitEmptyPasswords no X11Forwarding yes X11DisplayOffset 10 X11UseLocalhost yes PrintMotd yes UsePrivilegeSeparation yes Compression yes VerifyReverseMapping yes Subsystem sftp /usr/local/libexec/sftp-server </pre> <p>These settings will ensure that the root user cannot log in via ssh, strong server keys are used, users cannot have empty passwords, and that protocol 2 is forced.</p>
	4	Test ssh access from the Company X internal network; the auditor can utilize his or her desktop system for this test.
	5	Attempt to log in using the root user account over ssh. The root user should not be able to log in to the system.
	6	Attempt to utilize protocol 1 when connecting via ssh. The system should deny the client when it is tries to connect using protocol 1.
	7	Interview the members of the administrative team to determine if they only “su” when they are connected using ssh. If a team member does not follow this procedure, the administrative team should conduct user training.

Table 23 – IPTables Checklist

System 17 Checklist: IPTables Configuration		
Description	The following objective checklist will enable the auditor to determine if the IPTables firewall is effectively protecting the system.	
Reference	12, 13, 14, 15, netcat README, personal experience	
Risk	S1: An attacker could compromise the system.	
Control Objective	The objective of this checklist is to ensure that the system is adequately protected from network attacks. The software should log anomalous traffic.	
Compliance	Compliance is binary. The system must be configured to permit only the required system services; all other traffic should be blocked by the IPTables firewall.	
Objective/ Subjective	Objective	
Steps for Testing Compliance		
P/F	Step	Description of Step
	1	The purpose of the system is clearly defined within the SLA and the system documentation. The auditor should examine the IPTables security policy file, /usr/local/etc/iptables, to determine if the minimal number of services is permitted. Those services should include ssh, telnet, ftp, dns, and ntp (which were the services identified under the checklist for daemons and open ports). NAT services are not required, so they can be ignored.
	2	Verify that the system starts IPTables at boot. Execute “chkconfig --list” and identify that iptables is enabled for init levels two through five.
	3	Verify that IPTables is blocking inbound traffic by utilizing netcat. Set up a netcat listener on the ftp.companyx.com system by executing “echo “Howdy” nc -l -p 8080”. From the security-test system, connect to the ftp.companyx.com using telnet on port 8080. If the message “Howdy” is displayed, then the IPTables firewall is not functioning properly.
	4	Verify log entries for the traffic generated in step three. Log entries reside in the /var/log/iptables file.

Table 24 – Hosts.allow Checklist

System 18 Checklist: Hosts.allow Checklist	
Description	The following objective checklist will enable the auditor to determine if the libwrap utility is properly blocking non-permitted connections to the server.
Reference	Personal experience
Risk	S1 : An attacker could compromise the system.
Control	The objective of this checklist is to ensure that only known, preconfigured systems can

Objective		access the system using the ssh, ftp, and telnet services.
Compliance		Compliance is binary. The system must be configured to only allow permitted hosts to access the system using the ssh, ftp, and telnet services.
Objective/ Subjective		Objective
Steps for Testing Compliance		
P/F	Step	Description of Step
	1	Attempt to connect the system using ssh, telnet, and ftp. The access attempts should originate from a system that does not have access rights defined in the <code>/etc/hosts.allow</code> file. The security-test system may be utilized for this test; however, the auditor should ensure that it is not configured for access by examining the <code>/etc/hosts.allow</code> file. The system will pass this step if access is denied.
	2	Examine the root account's email box for alerts messages that indicate access was denied. The <code>/etc/hosts.allow</code> file is configured to email denied access attempts to root. This system passes this step if the email was received.
	3	<p>To test the syntax of the <code>/etc/hosts.allow</code> file, complete the following steps:</p> <ul style="list-style-type: none"> ○ On the ftp.companyx.com system, add an entry for the security-test system in the <code>/etc/hosts</code> file. ○ Add an entry in the <code>/etc/hosts.allow</code> file so that the security-test system can access the system using telnet. Test access to the ftp.companyx.com system using ssh, telnet, and ftp. Only telnet should be successful. Check the root user's email box for denied access messages. ○ Add an entry in the <code>/etc/hosts.allow</code> file so that the security-test system can access the system using ftp (without removing telnet access). Test access to the ftp.companyx.com system using ssh, telnet, and ftp. Only telnet and ftp should be successful. Check the root user's email box for denied access messages. ○ Add an entry in the <code>/etc/hosts.allow</code> file so that the security-test system can access the system using ssh (without removing telnet and ftp access). Test access to the ftp.companyx.com system using ssh, telnet, and ftp. All three attempts should be successful.
	4	For systems to be able to connect, the ftp.companyx.com system must be able to resolve it in DNS (forward and reverse). Remove the host entry for security-test.companyx.com system from the <code>/etc/hosts</code> file (the security-test.companyx.com DNS name does not exist in the external DNS zones). Retest access using ssh, ftp, and telnet. The system should not be able to connect.
	5	Remove the security-test system from the <code>/etc/hosts.allow</code> file (clean-up).

Assignment 3 – Conduct the Audit

When I had finished creating all of the relevant checklists, I proceeded with the execution of the audit in its entirety. I found the system to be compliant with all of the physical checklist control objects, as well as the majority of the system checklist control objectives. In this assignment, I have provided the results from the most pertinent checklist executions. The results discovered from carrying out the checklist steps have also been included in the form of commentary and screen shots. These results are directly related to the risk analysis that I have explored in Assignment 4.

System 2 Checklist Execution: Software Patches

Steps for Testing Compliance: F		
P/F	Step	Description of Step
P	1	Connect to the system by using ssh. <i>(trivial step, screen shot not included)</i>
P	2	Become the root user by executing "su -". <i>(trivial step, screen shot not included)</i>
F	3	Use the up2date program to check for new package releases by executing "up2date --nox --dry-run". Results from the program should indicate that no updates are required. If updates are required, check documentation for the system for patch waivers, otherwise system fails.

The system failed step three of this audit checklist. An execution of the up2date command revealed that RedHat had released newer packages, which may have included important bug fixes. Figure 5 shows the output from the command. Further research indicated that the sendmail, unzip, and tcpdump packages that RedHat released included important bug fixes, some of which could be used to possibly crash the system or install trojans (RedHat Network).

Figure 6 – Up2date Command Output

```
[root@ftp ~]# up2date --nox --dry-run

Fetching package list for channel: redhat-linux-i386-9...
#####

Fetching Obsoletes list for channel: redhat-linux-i386-9...
#####

Fetching rpm headers...
#####

Testing package set / solving RPM inter-dependencies...
#####

Name                                Version      Rel
-----
bash                                2.05b        20.1
glibc                               2.3.2        27.9
glibc-common                        2.3.2        27.9
glibc-devel                         2.3.2        27.9
gnupg                               1.2.1        4
krb5-libs                           1.2.7        14
redhat-config-date                  1.5.15       1
redhat-config-network               1.2.15       1
redhat-config-network-tui          1.2.15       1
rhpl                                 0.93.4       1
sendmail                            8.12.8       6.90
sendmail-cf                         8.12.8       6.90
tcpdump                             3.7.2        1.9.1
unzip                               5.50         33
xinetd                              2.3.11       1.9.0

The following Packages were marked to be skipped by your configuration:

Name                                Version      Rel  Reason
-----
--
kernel                              2.4.20       20.9 Pkg name/pattern
kernel-source                       2.4.20       20.9 Pkg name/pattern
openssl                             0.9.7a       5    Pkg name/pattern

[root@ftp ~]#
```

P	4	Blackbox window manager: http://blackboxwm.sourceforge.net/ . Compare the latest stable release number with that of the version installed on the system. Execute <code>"/usr/local/bin/blackbox -v"</code> to obtain the version of software installed on the system.
---	---	--

The Blackbox window manager was found to be at the highest revision level available, version 0.65.0. Figures 6 and 7 below show the results of the test.

Figure 7 – Blackbox Website

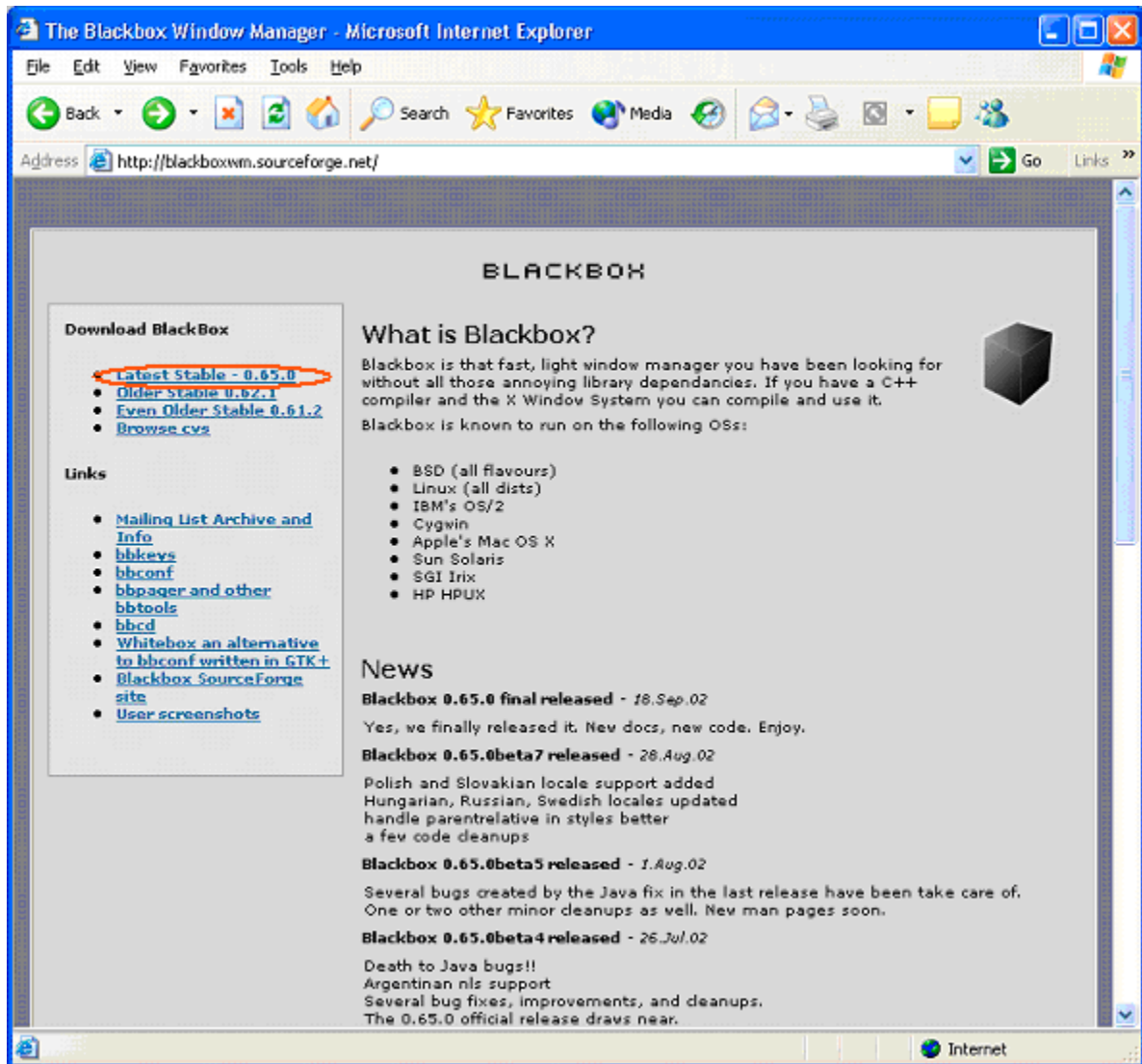
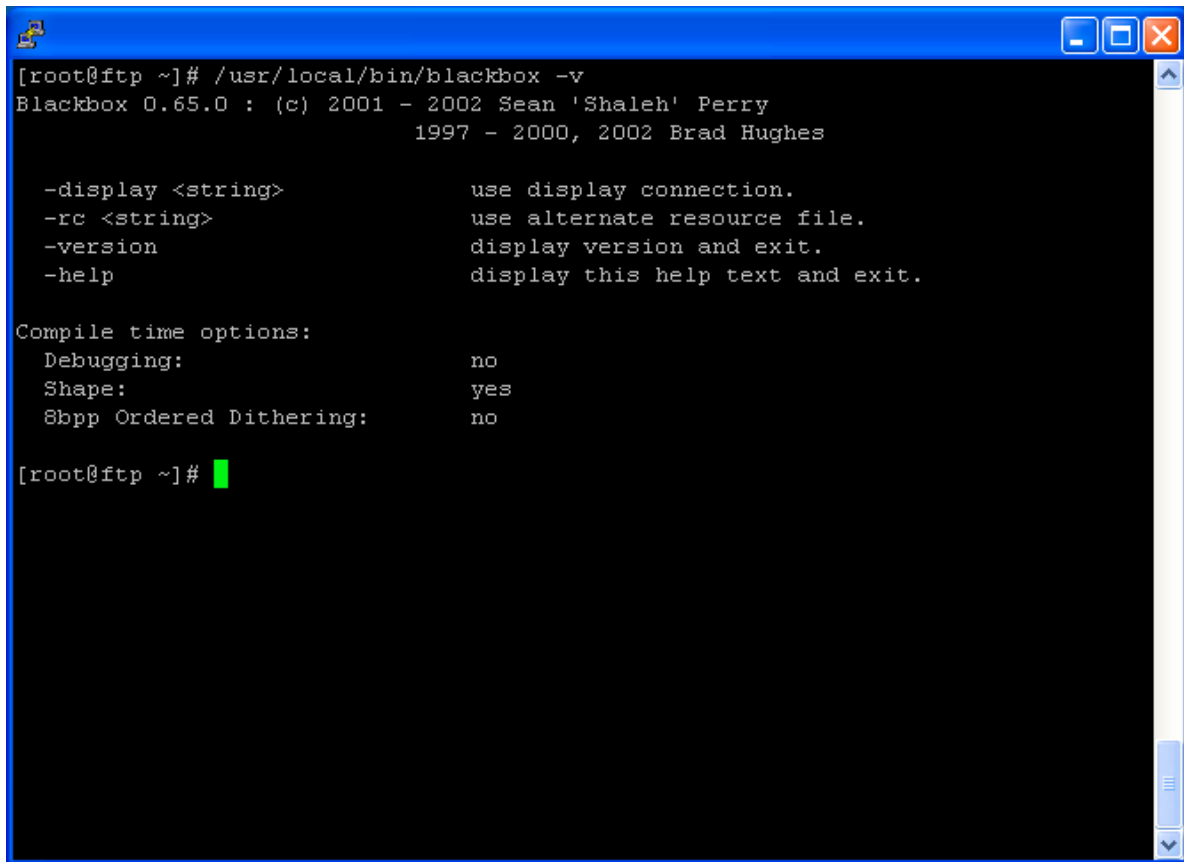


Figure 8 – Blackbox Version Command



```
[root@ftp ~]# /usr/local/bin/blackbox -v
Blackbox 0.65.0 : (c) 2001 - 2002 Sean 'Shaleh' Perry
                  1997 - 2000, 2002 Brad Hughes

    -display <string>      use display connection.
    -rc <string>           use alternate resource file.
    -version               display version and exit.
    -help                  display this help text and exit.

Compile time options:
  Debugging:              no
  Shape:                  yes
  8bpp Ordered Dithering: no

[root@ftp ~]#
```

P	5	OpenSSL: http://www.openssl.org . Compare the latest stable release number with that of the version installed on the system. Execute “/usr/local/ssl/bin/openssl version” to obtain the version of software installed on the system.
---	---	---

The OpenSSL software on the system was found to be at the highest revision level, version 0.9.7b. Figure 8 and 9 below show the results from the test.

© SANS Institute

Figure 9 – OpenSSL Website

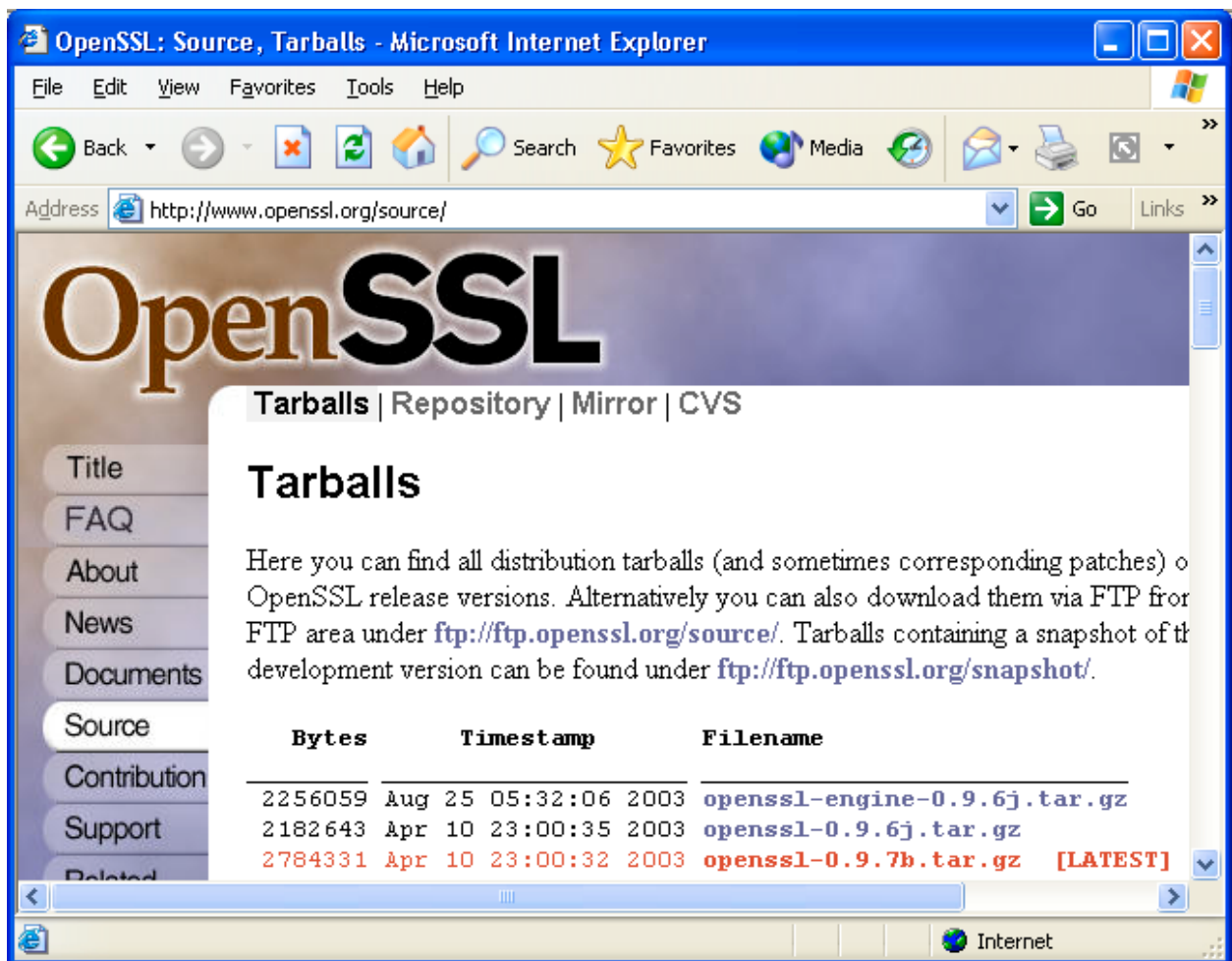
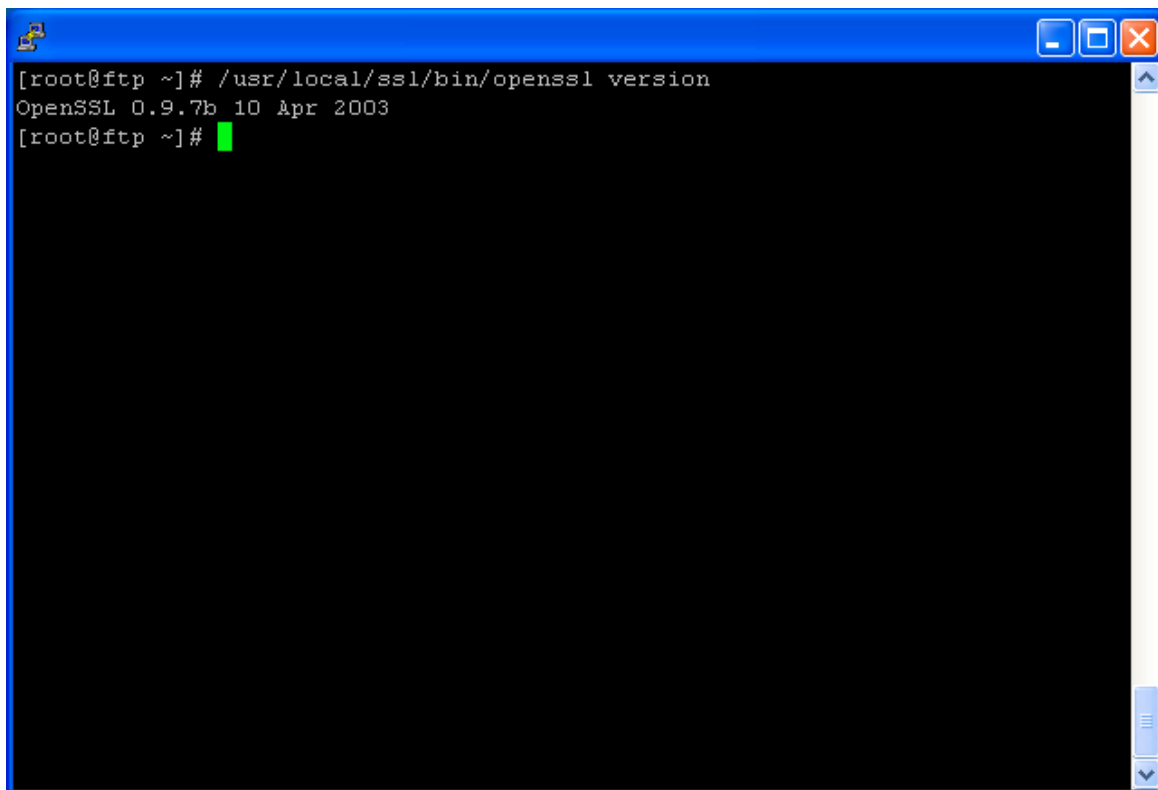


Figure 10 – OpenSSL Version Command



```
[root@ftp ~]# /usr/local/ssl/bin/openssl version
OpenSSL 0.9.7b 10 Apr 2003
[root@ftp ~]#
```

P	6	OpenSSH: http://www.openssh.org . Compare the latest stable release number with that of the version installed on the system. Execute both “/usr/local/bin/ssh -V” and “/usr/local/sbin/sshd -V” to obtain the version information from the system.
---	---	---

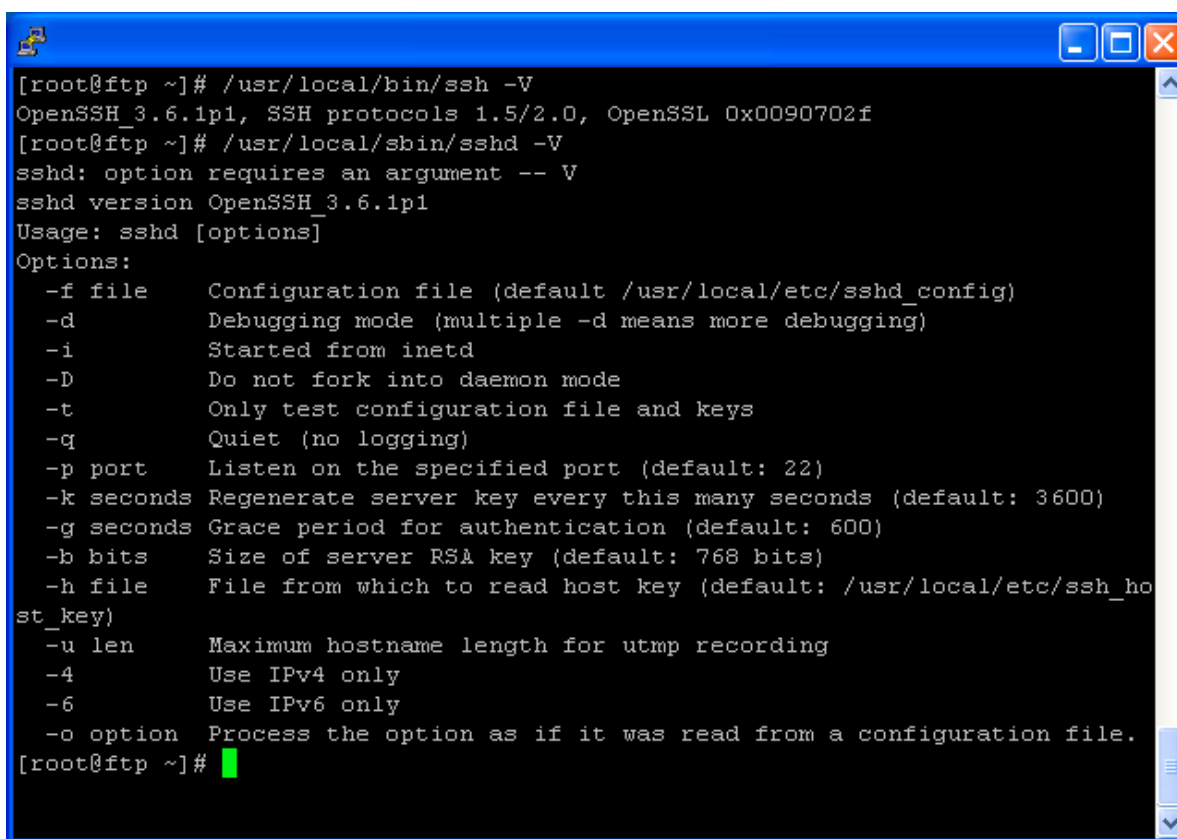
The OpenSSH software on the system was found to be at the highest revision level, version 3.6.1-p1. Figure 10 and 11 below show the results from the test.

© SANS Institute

Figure 11 – OpenSSH Website



Figure 12 – OpenSSH Version Command

A terminal window with a blue title bar and standard window controls. The terminal text shows the execution of 'ssh -V' and 'sshd -V' commands, followed by a list of options for the sshd daemon. The output indicates OpenSSH 3.6.1p1 is installed with SSH protocols 1.5/2.0 and OpenSSL 0x0090702f.

```
[root@ftp ~]# /usr/local/bin/ssh -V
OpenSSH_3.6.1p1, SSH protocols 1.5/2.0, OpenSSL 0x0090702f
[root@ftp ~]# /usr/local/sbin/sshd -V
sshd: option requires an argument -- V
sshd version OpenSSH_3.6.1p1
Usage: sshd [options]
Options:
  -f file      Configuration file (default /usr/local/etc/sshd_config)
  -d           Debugging mode (multiple -d means more debugging)
  -i           Started from inetd
  -D           Do not fork into daemon mode
  -t           Only test configuration file and keys
  -q           Quiet (no logging)
  -p port      Listen on the specified port (default: 22)
  -k seconds   Regenerate server key every this many seconds (default: 3600)
  -g seconds   Grace period for authentication (default: 600)
  -b bits      Size of server RSA key (default: 768 bits)
  -h file      File from which to read host key (default: /usr/local/etc/ssh_ho
st_key)
  -u len       Maximum hostname length for utmp recording
  -4           Use IPv4 only
  -6           Use IPv6 only
  -o option    Process the option as if it was read from a configuration file.
[root@ftp ~]#
```

P	7	IPllogger by Ojnk Software: http://ojnk.sourceforge.net . Compare the latest stable release number with that of the version installed on the system. Execute “/usr/local/sbin/iplog -version” to obtain the version of the software installed on the system.
---	---	--

The IPLog software, which is used for rudimentary packet logging and nmap avoidance, was found to be up to the highest revision available, version 2.2.3. Figures 12 and 13 depict the results from the test.

Figure 13 – Iplog Website

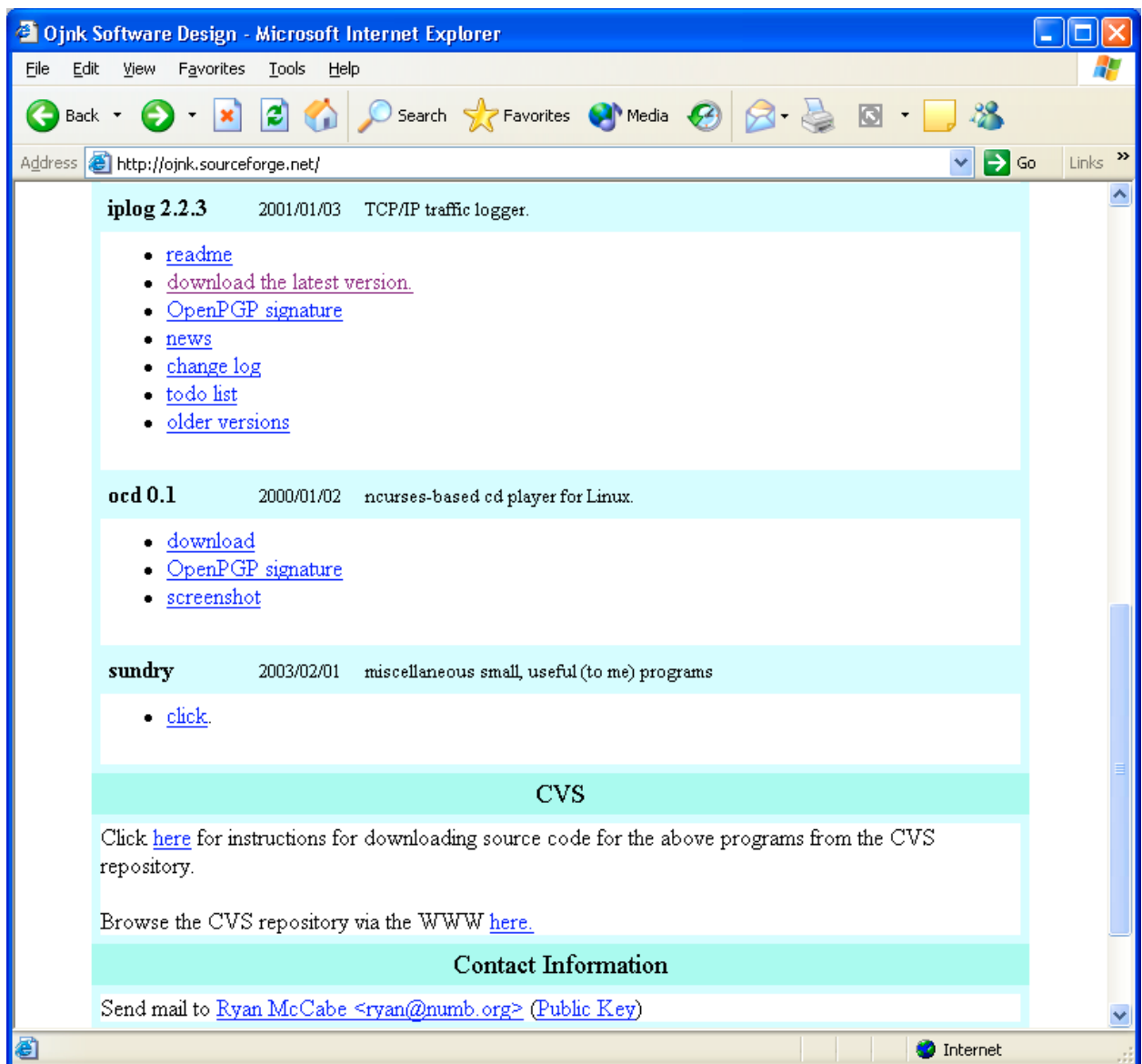


Figure 14 – Iplog Version Command

```
[root@ftp ~]# /usr/local/sbin/iplog -version
iplog version 2.2.3
by Ryan McCabe <odin@numb.org>
http://ojnk.sourceforge.net
[root@ftp ~]#
```

F	8	If version numbers do not match, the system will fail this audit item. The administrative team must update the software, unless they determine that the risk is acceptable.
----------	----------	---

Checklist Results: The system was not to be up to date with the latest release of software. In particular, the items found in figure 5 needed addressing.

System 5 Checklist Execution: Kernel Vulnerabilities and Settings

Steps for Testing Compliance: F		
P/F	Step	Description of Step
P	1	Connect to the ftp.companynx.com server using ssh and become the root user by executing “su –”. (<i>Trivial step, no screen shots provided</i>)
P	2	Execute the “uname -r” command to obtain the currently used kernel version.

The system’s kernel version is 2.4.20-8.

Figure 15 – Kernel Version

```

baumansc@ftp:/home/ftp/users/security/baumansc
[root@ftp ~]# uname -r
2.4.20-8
[root@ftp ~]#

```

P	3	Utilize the up2date program to determine if RedHat has a new kernel source available in RPM format. The command to use is “up2date -nox -dry-run”. The administrative team does not allow up2date to upgrade the kernel, so the results reside in the “skipped” section of the output.
---	---	--

The last output items from the up2date command are the items that the check ignores. The administrative team previously configured the up2date program to skip packages over which that they want a greater degree of version control.

Figure 16 – Up2date Kernel Version

```

baumansc@ftp:/home/ftp/users/security/baumansc
The following Packages were marked to be skipped by your configuration:

Name                               Version      Rel Reason
-----
kernel                             2.4.20       20.9 Pkg name/pattern
kernel-source                       2.4.20       20.9 Pkg name/pattern
openssl                             0.9.7a       5    Pkg name/pattern

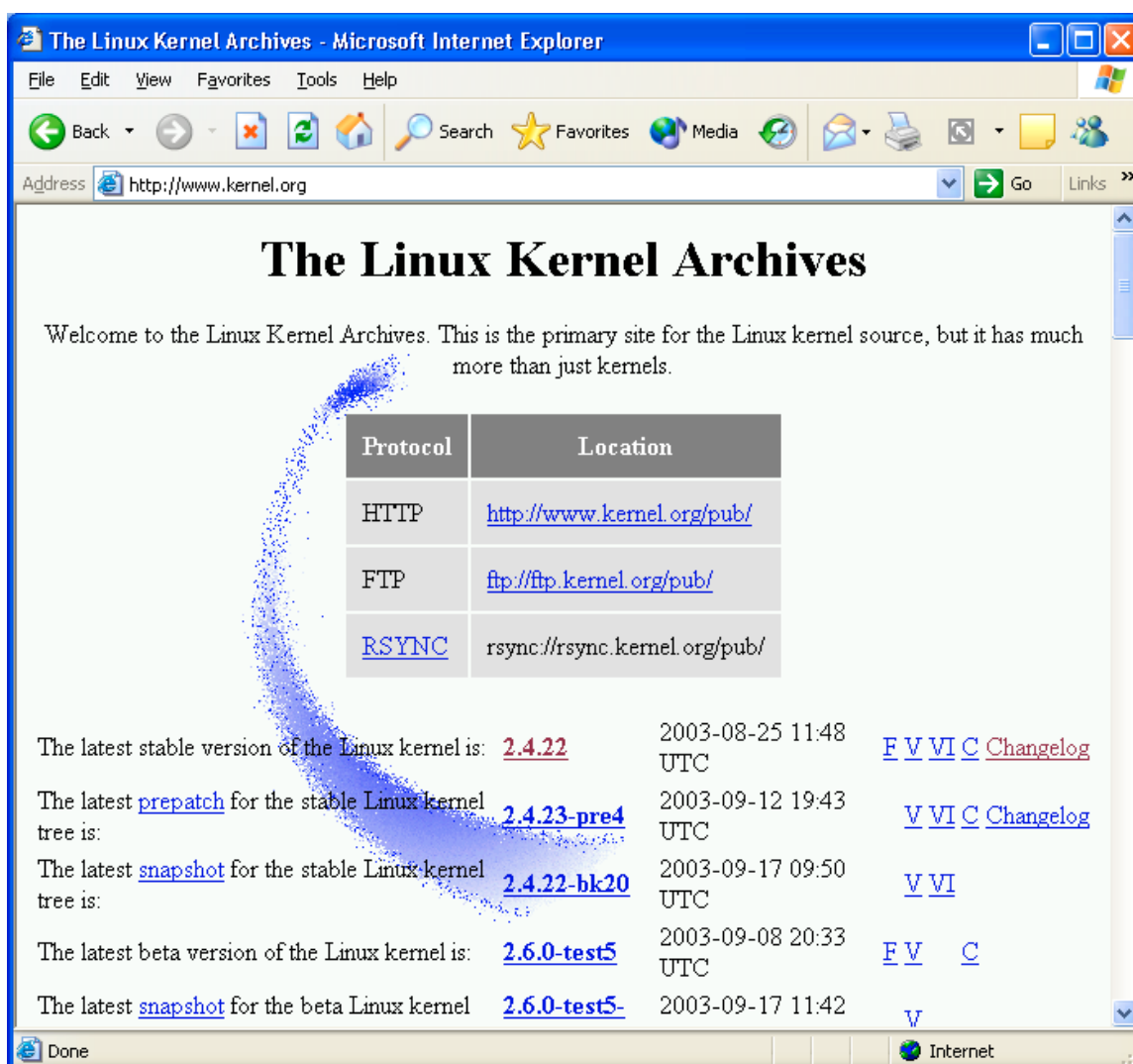
[root@ftp ~]#

```

F	4	Compare the results from steps two and three with version information found on http://www.kernel.org . The system can still be ultimately compliant with the control objectives if it passes step five.
---	---	---

The latest kernel version on the kernel.org website is 2.4.22, which is well ahead of the kernel running on the system. This could indicate that there are vulnerabilities in the system’s kernel.

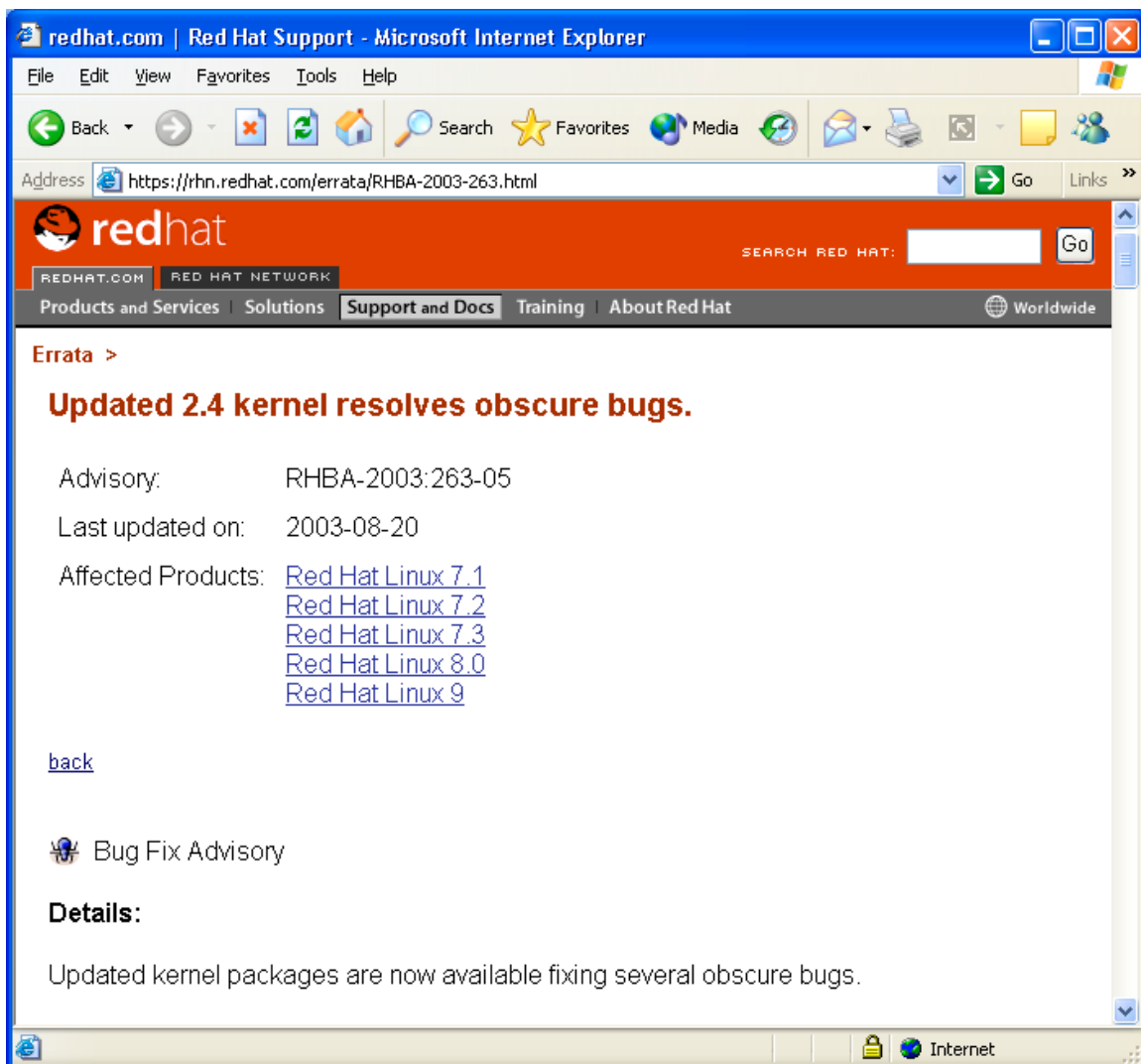
Figure 17 – Kernel.org Website



F	5	Verify on http://www.redhat.com , http://www.securityfocus.com and http://www.cert.org that the current compiled kernel version on ftp.company.com does not contain a security vulnerability. If a known vulnerability exists in the currently used kernel version, the system is not compliant.
---	---	---

A visit to the RedHat website was enough to discover that there are a number of vulnerabilities with the 2.4.20-8 kernel. Figure 18 shows the URL of one of the vulnerabilities, which I deemed as a serious risk to the system. While the bug is obscure, it can still be exploited.

Figure 18 – RedHat Website



P	6	<p>Verify all kernel options. This step will ensure that important kernel options, like loadable kernel modules, are correctly configured (in this case, disabled). Compare the <code>/usr/src/linux-2.4/.config</code> to the baseline version saved on the <code>management.companyx.com</code> system. The baseline version of the <code>.config</code> is stored in the <code>/home/system-backups/ftp.companyx.com/kernel-options.ORIG</code> file.</p> <ul style="list-style-type: none"> ○ Store a copy of the <code>.config</code> file, named as <code>kernel-options.txt</code>, in the auditor's home directory. ○ Create an MD5 checksum of the <code>kernel-options.txt</code> file with the command <code>"md5sum ~/kernel-options.txt > ~/kernel-options.md5"</code>. ○ On the management system, compare the MD5 checksum of the <code>kernel-options.ORIG</code> file to the contents of the <code>kernel-options.ORIG.md5</code> file by executing <code>"md5sum kernel-options.ORIG; cat kernel-options.ORIG.md5"</code>. ○ Download the <code>kernel-options.txt</code> and <code>kernel-options.md5</code> from the
---	---	---

		<p>ftp.companyx.com server, and compare their MD5 checksums with the command "md5sum kernel-options.txt; cat kernel-options.md5".</p> <ul style="list-style-type: none"> ○ Compare the two configuration files using the diff command: "diff kernel-options.ORIG kernel-options.txt". <p>The system will be compliant if the two files are identical.</p>
--	--	--

I compared the kernel configuration options with the baseline file store on the management server. The two files were identical. Figure 19 shows the process of downloading the file. The diff command output produced no results, meaning that the files were identical.

© SANS Institute 2003, Author retains full rights

Figure 19 – Kernel Options Download

```

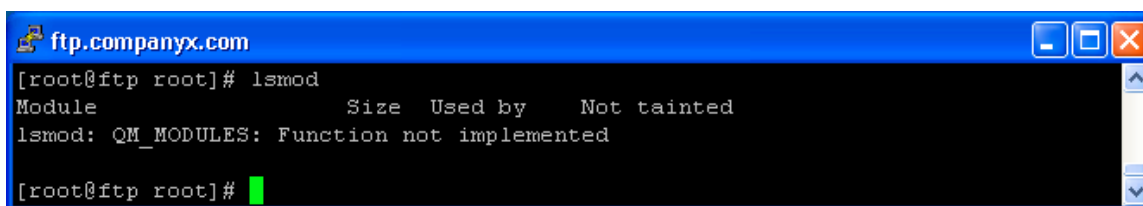
management.companyx.com
[baumansc@management ftp.companyx.com]# md5sum kernel-options.Orig ; cat kernel-
options.Orig.md5
a00eb3574fbeb4bd6fca99c97cb6c1c9  kernel-options.Orig
a00eb3574fbeb4bd6fca99c97cb6c1c9  kernel-options.Orig
[baumansc@management ftp.companyx.com]# ftp ftp
Connected to ftp ( X.X.X.X ).
220-
220-   Unauthorized access to this computer is in violation of Article 27,
220-   Sections 45A and 146 of the Annotated Code of Maryland and will be
220-   prosecuted to the full extent of the law. All usage of this system
220-   is monitored for security purposes, and by signing on to the system
220-   you are implicitly consenting to this monitoring.
220-
220
Name (ftp:baumansc):
331 Please specify the password.
Password:
230 Login successful. Have fun.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> bin
200 Switching to Binary mode.
ftp> hash
Hash mark printing on (1024 bytes/hash mark).
ftp> mget kernel-options.*
mget kernel-options.md5? yes
227 Passive mode entered ( X,X,X,X ,187,127)
150 Opening BINARY mode data connection for kernel-options.md5 (89 bytes).
#
226 File send OK.
89 bytes received in 0.00104 secs (84 Kbytes/sec)
mget kernel-options.txt? yes
227 Passive mode entered ( X,X,X,X ,187,130)
150 Opening BINARY mode data connection for kernel-options.txt (16128 bytes).
#####
226 File send OK.
16128 bytes received in 0.00633 secs (2.5e+03 Kbytes/sec)
ftp> quit
221 Goodbye.
[baumansc@management ftp.companyx.com]# md5sum kernel-options.txt; cat kernel-op
tions.md5
a00eb3574fbeb4bd6fca99c97cb6c1c9  kernel-options.txt
a00eb3574fbeb4bd6fca99c97cb6c1c9  /home/ftp/users/./security/baumansc/kernel-opt
ions.txt
[baumansc@management ftp.companyx.com]#

```

P	7	Manually inspect the system for loadable kernel modules by executing “lsmod” on the ftp.companyx.com system. The output from the command should be: lsmod: QM MODULES: Function not implemented
---	---	--

I found that the system does not implement loadable kernel modules. Figure 20 depicts the output from the “lsmod” command.

Figure 20 – Loadable Kernel Modules



```
ftp.companyx.com
[root@ftp root]# lsmod
Module                Size  Used by    Not tainted
lsmod: QM_MODULES: Function not implemented

[root@ftp root]#
```

Checklist Results: The system failed this checklist. There are several known vulnerabilities with the currently run kernel. For the system to become compliant, the administrative team must update the kernel to at least 2.4.20-20.9, which is the latest from the RedHat site.

System 8 Checklist Execution: DoS

Steps for Testing Compliance: P		
P/F	Step	Description of Step
P	1	Log in to the security-test Linux system as root. Start the Nessus daemon by executing “/usr/local/sbin/nessusd -D”. Start the Nessus client by executing “/usr/local/bin/nessus”. Log in to the client using a previously created Nessus user account that has privileges to test. <i>(trivial step, no screen shots provided)</i>
P	2	Configure Nessus to scan for DoS vulnerabilities by navigating to the Plugins -> Denial Of Service plugin selection. Ensure that all options are checked. Click on Target Selection and specify the IP address of the ftp.companyx.com server, and select “Start the scan” to begin the test. The system is compliant if the Nessus reports no vulnerabilities.

I configured the Nessus tool to test for DoS attacks against the ftp.companyx.com server. Figure 21 shows the Nessus GUI. The Nessus scan revealed that the system was vulnerable to one DoS attack, an attack against the FTP server software. My research revealed that this was a false alarm; the vulnerability report indicated that the attack could crash the server by reading certain devices or accessing certain file names. I logged in to the ftp.companyx.com server to verify that the software had not crashed. In addition, the /var/log/messages file only indicated that Xinetd had denied access to the security-test system because it was not in the /etc/hosts.allow file. This Nessus plugin caused the false alarm because the libwrap utility abruptly closed the connection during the test. Figure 22 displays the Nessus report.

Figure 21 – Nessus DoS Configuration

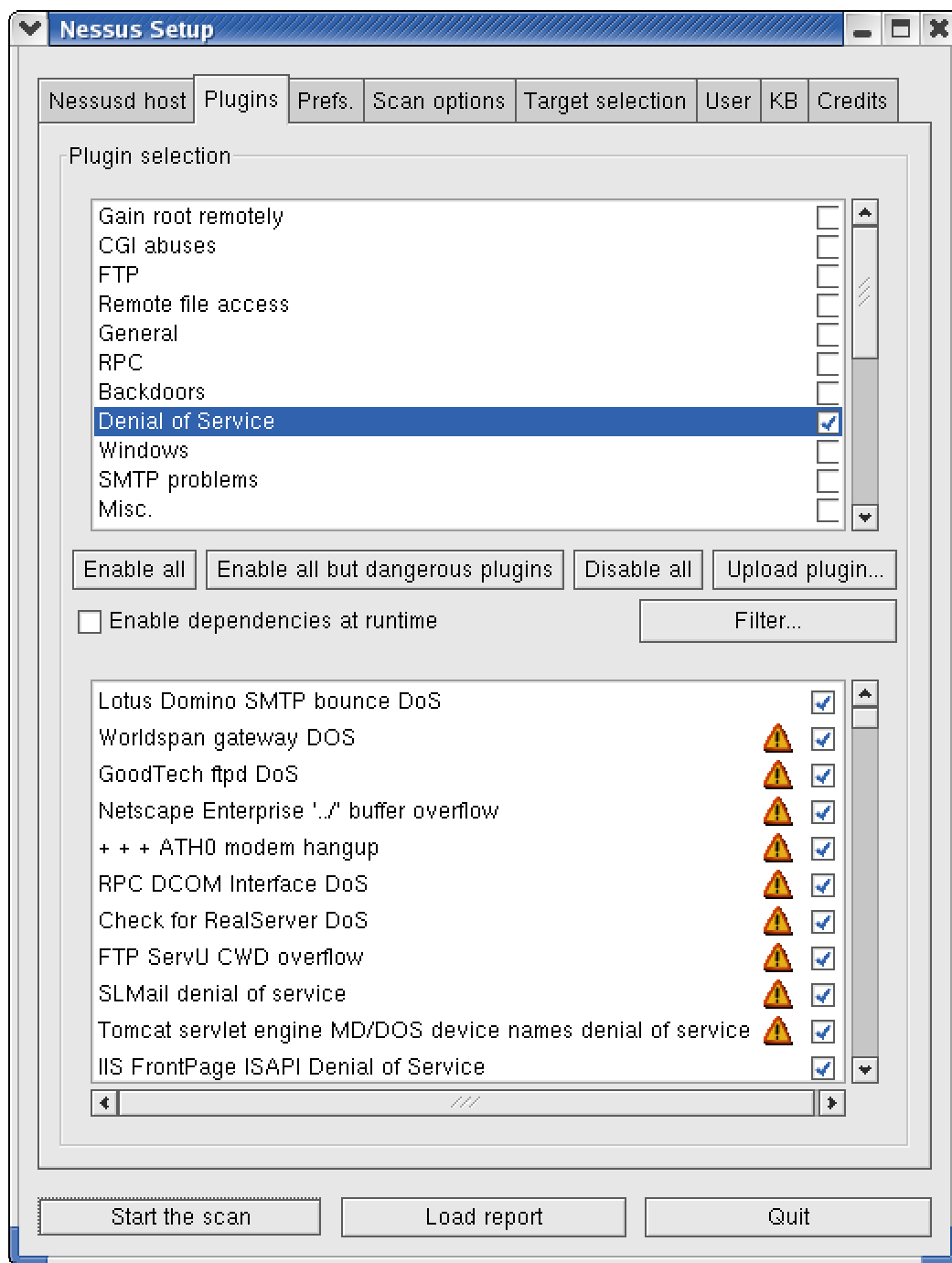
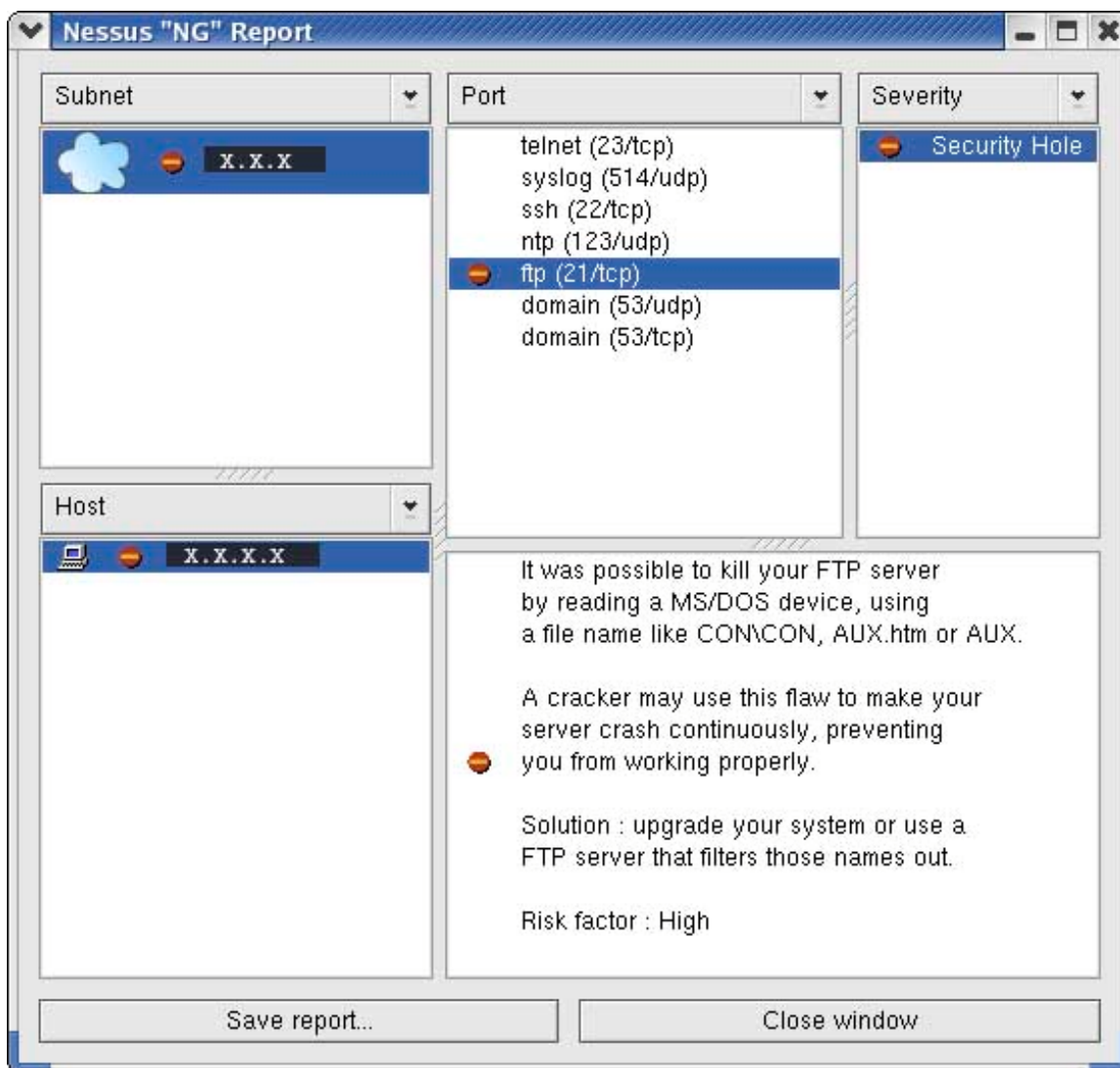


Figure 22 – Nessus DoS Report



P	3	<p>The Xinetd service is a replacement for the inetd service. It is used to provide access control, along with libwrap (hosts.allow, hosts.deny), for telnet, FTP, and OpenSSH. Verify that Xinetd is configured to provide fifty simultaneous connections, where twenty five can originate from the same host. These settings are useful for preventing a DoS attack. The following lines should be present in the ftp.companyx.com /etc/xinted.conf file:</p> <pre>instances = 50 log_on_success = HOST PID DURATION log_on_failure = HOST USERID per_source = 25</pre>
---	---	---

I verified that the Xinetd settings are correct. Figure 23 shows the contents of the file.

Figure 23 – Xinetd Settings

```

[baumannsc@management /etc]# cat xinetd.conf
#
# Simple configuration file for xinetd
#
# Some defaults, and include /etc/xinetd.d/
#
# USERID option for log_on_success causes an auth/ident request to
# be made.  If the remote end does not allow ident or if router
# blocks it, time to login prompt is terribly long.
#

defaults
{
    instances                = 50
    log_type                  = SYSLOG authpriv
    log_on_success             = HOST PID DURATION
    log_on_failure             = HOST USERID
    cps                       = 25 30
    per_source                 = 25
}

includedir /etc/xinetd.d
[baumannsc@management /etc]#

```

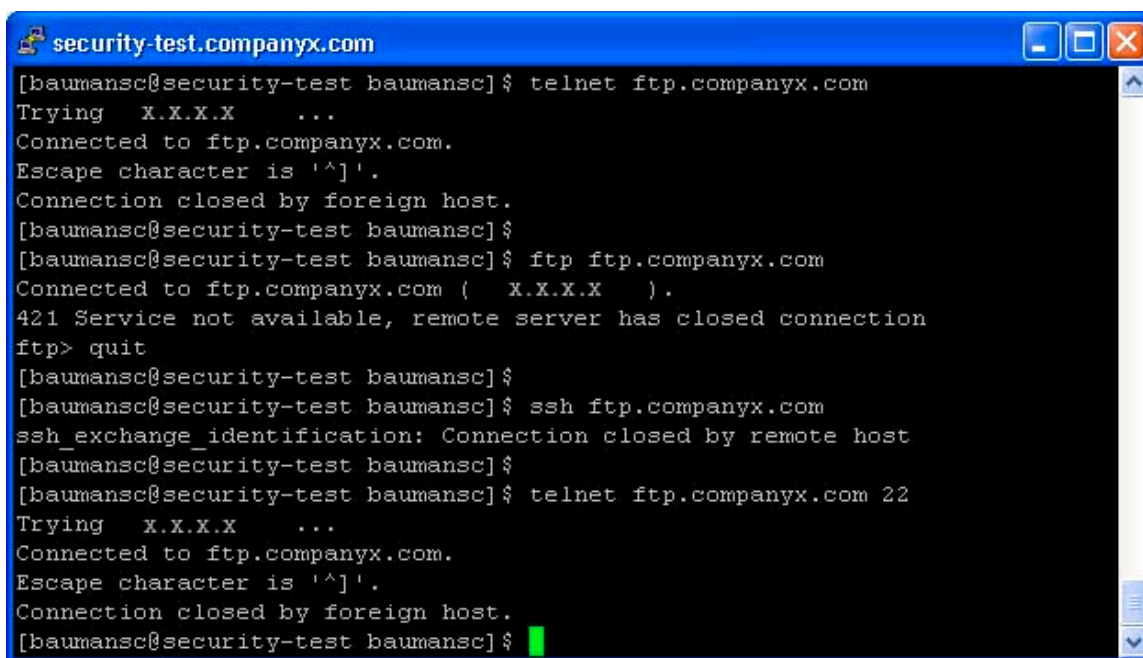
Checklist Results: This system has passed this checklist; I have found it to be resistant to all of the Nessus plugins for DoS attacks. In addition, the Xinetd configuration is correct.

System 9 Checklist Execution: System Reconnaissance

Steps for Testing Compliance: F		
P/F	Step	Description of Step
P	1	From the security-test Linux system, retrieve banner information for both telnet and FTP. Banner information should not reveal any information about the OS of the system or version of the software.

I used the security-test system, which was not in the `/etc/hosts/allow` file, to gather the banner information. The system did not reveal any important information (besides the fact that it is running something on those ports). Figure 24 shows the results:

Figure 24 – Unauthorized Banners

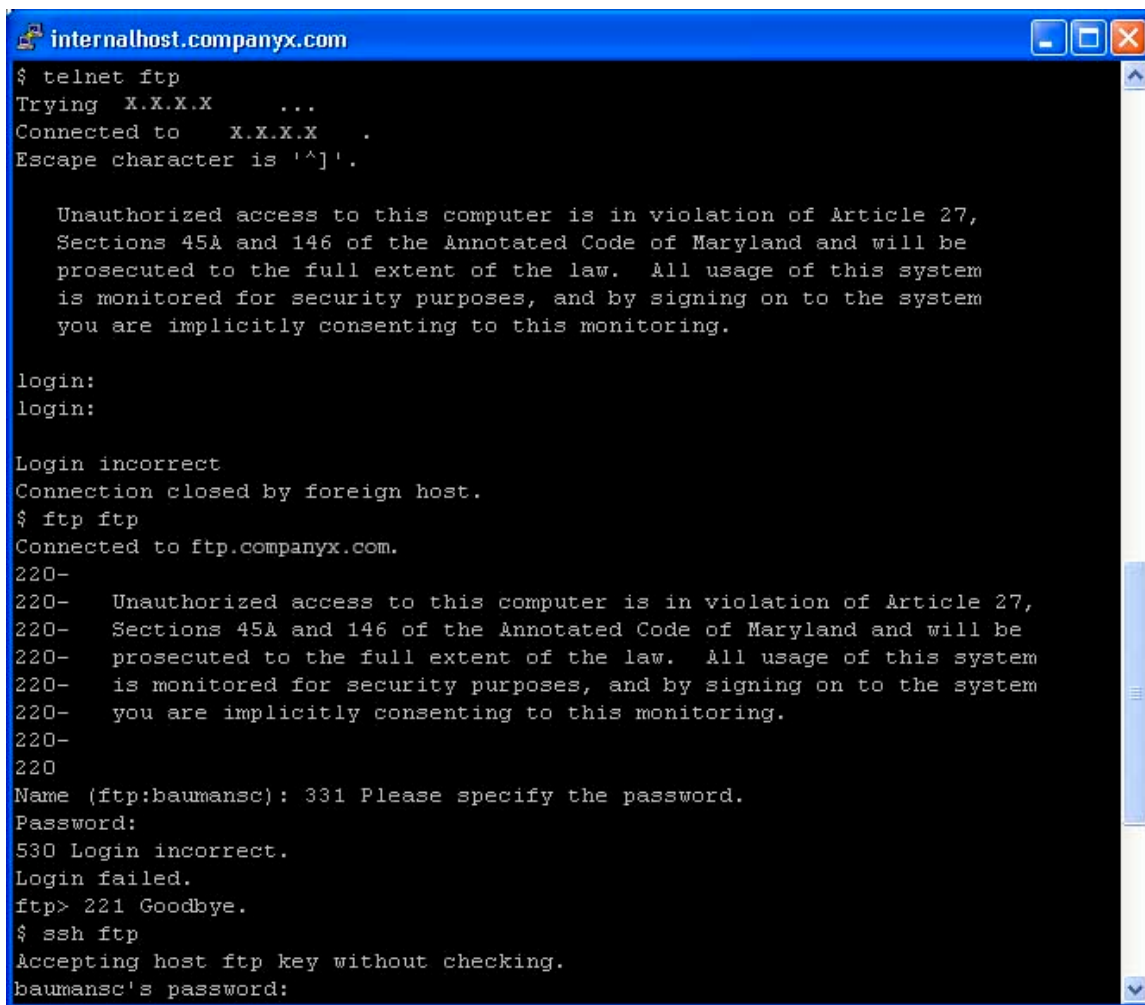


```
security-test.companyx.com
[baumansc@security-test baumansc]$ telnet ftp.companyx.com
Trying X.X.X.X ...
Connected to ftp.companyx.com.
Escape character is '^]'.
Connection closed by foreign host.
[baumansc@security-test baumansc]$
[baumansc@security-test baumansc]$ ftp ftp.companyx.com
Connected to ftp.companyx.com ( X.X.X.X ).
421 Service not available, remote server has closed connection
ftp> quit
[baumansc@security-test baumansc]$
[baumansc@security-test baumansc]$ ssh ftp.companyx.com
ssh_exchange_identification: Connection closed by remote host
[baumansc@security-test baumansc]$
[baumansc@security-test baumansc]$ telnet ftp.companyx.com 22
Trying X.X.X.X ...
Connected to ftp.companyx.com.
Escape character is '^]'.
Connection closed by foreign host.
[baumansc@security-test baumansc]$
```

P	2	From a system within Company X, retrieve banner information for both telnet and FTP. The banners shown for hosts present in the <code>/etc/hosts.allow</code> may be different from unknown hosts. Banner information should not reveal any information about the OS of the system or the version of the software.
---	---	--

I used my desktop system, which is on the Company X internal network, to gather the banner information from an “authorized system.” Figure 25 depicts the test results. I was not able to gather any version information from the banners. The system passed this step.

Figure 25 – Authorized Banners



```
internalhost.companyx.com
$ telnet ftp
Trying X.X.X.X ...
Connected to X.X.X.X .
Escape character is '^]'.

  Unauthorized access to this computer is in violation of Article 27,
  Sections 45A and 146 of the Annotated Code of Maryland and will be
  prosecuted to the full extent of the law.  All usage of this system
  is monitored for security purposes, and by signing on to the system
  you are implicitly consenting to this monitoring.

login:
login:

Login incorrect
Connection closed by foreign host.
$ ftp ftp
Connected to ftp.companyx.com.
220-
220-   Unauthorized access to this computer is in violation of Article 27,
220-   Sections 45A and 146 of the Annotated Code of Maryland and will be
220-   prosecuted to the full extent of the law.  All usage of this system
220-   is monitored for security purposes, and by signing on to the system
220-   you are implicitly consenting to this monitoring.
220-
220
Name (ftp:baumansc): 331 Please specify the password.
Password:
530 Login incorrect.
Login failed.
ftp> 221 Goodbye.
$ ssh ftp
Accepting host ftp key without checking.
baumansc's password:
```

P	3	From the security-test Linux system, query the DNS server for version information by issuing the command “dig @ftp.companyx.com version.bind chaos txt”. Obtain the version information for the package installed by querying the RPM database with the command “rpm -qa grep bind”. The two version numbers should not match.
---	---	--

Using the dig command from the security-test system, I found that the BIND version was very outdated. This was obviously not the true version number, since the results of the System 2 checklist proved otherwise. Figure 26 shows the results of the dig command, where figure 27 shows the true version of BIND that is running.

Figure 26 – BIND Version Dig

```

baumansc@security-test:~
[baumansc@security-test baumansc]$ dig @ftp version.bind chaos txt

; <<>> DiG 9.2.1 <<>> @ftp version.bind chaos txt
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1475
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;version.bind.                CH      TXT

;; ANSWER SECTION:
version.bind.                0      CH      TXT      "4.9.3"

;; Query time: 3 msec
;; SERVER:      X.X.X.X  #53(ftp)
;; WHEN: Wed Sep 17 16:11:04 2003
;; MSG SIZE  rcvd: 48

[baumansc@security-test baumansc]$

```

Figure 27 – BIND RPM Version

```

baumansc@ftp:/etc
[baumansc@ftp etc]$ rpm -qa | grep bind
bind-utils-9.2.1-16
bind-9.2.1-16

```

F	4	Utilize nmap to attempt to guess the OS type of the system. From the security-test system, execute the command “nmap -O ftp.companyx.com”. Check fingerprint information for information listed for the system. If the system type is obfuscated, than this step is passed.
---	---	---

The some system was revealed by the nmap scan. The strings shown in figure 28 indicate that the system is running RedHat. However, it does not pinpoint the version of RedHat that is running. Nevertheless, the administrative team should research this further to determine if the OS type and version could be further hidden from reconnaissance attempts.

Figure 28 – Nmap OS Fingerprint

```

security-test.companyx.com
[root@security-test root]# nmap -O ftp.companyx.com

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on ftp.companyx.com ( X.X.X.X ):
(The 1597 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
53/tcp    open       domain
No exact OS matches for host (If you know what OS is running on it, see http://www.insecure.org/cgi-bin/nmap-submit.cgi).
TCP/IP fingerprint:
SInfo(V=3.00%P=i386-redhat-linux-gnu%D=9/17%Time=3F68C2CA%O=21%C=1)
TSeq(Class=RI%gcd=1%SI=3B86E2%IPID=Z%TS=U)
TSeq(Class=RI%gcd=1%SI=3B86CF%IPID=Z%TS=U)
TSeq(Class=RI%gcd=1%SI=3B86EE%IPID=Z%TS=U)
T1(Resp=Y%DF=Y%W=16D0%ACK=S++%Flags=AS%Ops=M)
T2(Resp=Y%DF=Y%W=100%ACK=O%Flags=UAPF%Ops=)
T2(Resp=Y%DF=N%W=0%ACK=O%Flags=UAPR%Ops=)
T2(Resp=Y%DF=N%W=0%ACK=O%Flags=UAPRS%Ops=)
T3(Resp=Y%DF=Y%W=16D0%ACK=S++%Flags=AS%Ops=M)
T4(Resp=Y%DF=Y%W=0%ACK=O%Flags=R%Ops=)
T5(Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=Y%W=0%ACK=O%Flags=R%Ops=)
T7(Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
PU(Resp=Y%DF=N%TOS=CO%IPLEN=164%RIPTL=148%RID=E%RIPCK=E%UCK=E%ULEN=134%DAT=E)

Nmap run completed -- 1 IP address (1 host up) scanned in 12 seconds
[root@security-test root]#

```

Checklist results: The system passed all checklist items, except for number four. Nmap was able to grab enough information for an attacker to determine the OS type. Overall, the system has failed the checklist since it has not totally met the control objectives.

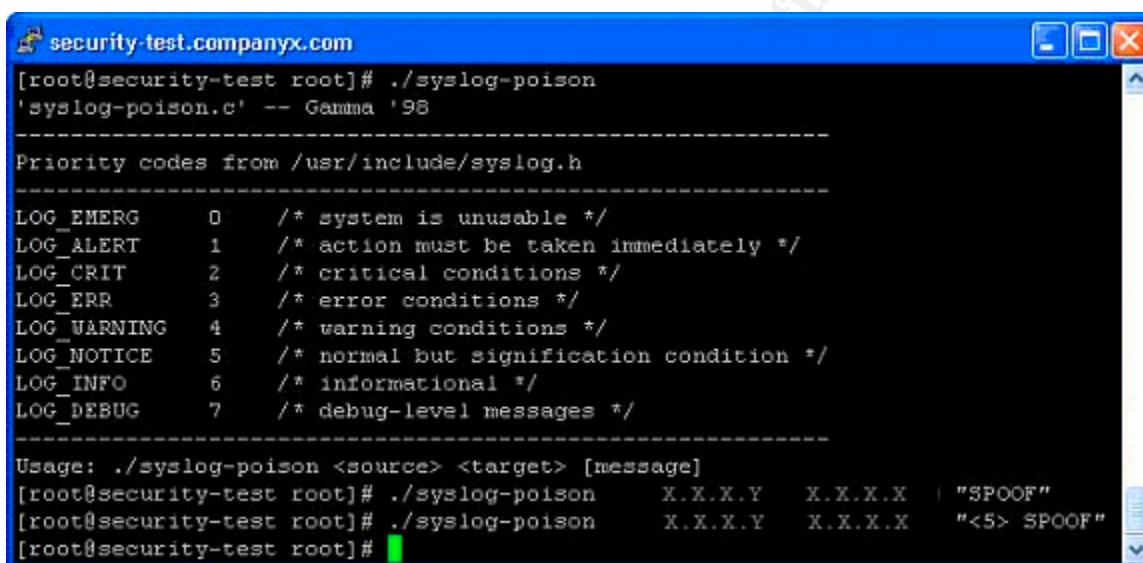
System 10 Checklist Execution: Syslog Audit

Steps for Testing Compliance: F		
P/F	Step	Description of Step
P	1	Obtain the syslog-poison.c code (Gamma '98) and compile on the security-test Linux system. Download the file from http://content.443.ch/pub/linfiles/Gnusoftware/spoofcode/syslog-poison.c . Compile the code using "gcc syslog-poison.c -o syslog-poison". (Trivial)
F	2	Generate a spoofed syslog message that uses the Company X internet router's IP address as

the source. To do this, execute “syslog-poison ir.companyx.com ftp.companyx.com “SPOOFED””. Examine the /var/log/messages file for evidence of the spoofed syslog alert: “grep SPOOFED /var/log/messages”. This system passes this step if the alert is not logged.

Unfortunately, the system did not pass this test. I was able to generate spoofed syslog packets from the security-test system, and the ftp.companyx.com system logged them to its /var/log/messages file. This is a flaw with the protocol, not necessarily a flaw with the system. However, these results show that the administrative team has not mitigated this risk with system configuration. Theoretically, an attacker could fill the entire /var partition by logging an exorbitant amount of syslog alerts.

Figure 29 – Syslog Poison



```
security-test.companyx.com
[root@security-test root]# ./syslog-poison
'syslog-poison.c' -- Gamma '98

-----
Priority codes from /usr/include/syslog.h
-----
LOG_EMERG      0   /* system is unusable */
LOG_ALERT      1   /* action must be taken immediately */
LOG_CRIT       2   /* critical conditions */
LOG_ERR        3   /* error conditions */
LOG_WARNING    4   /* warning conditions */
LOG_NOTICE     5   /* normal but signification condition */
LOG_INFO       6   /* informational */
LOG_DEBUG      7   /* debug-level messages */
-----

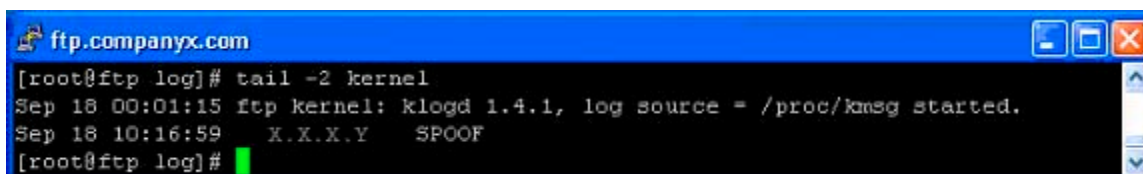
Usage: ./syslog-poison <source> <target> [message]
[root@security-test root]# ./syslog-poison      X.X.X.Y   X.X.X.X   "SPOOF"
[root@security-test root]# ./syslog-poison      X.X.X.Y   X.X.X.X   "<5> SPOOF"
[root@security-test root]#
```

Figure 30 – Syslog Spoofed Messages



```
ftp.companyx.com
[root@ftp log]# tail -2 messages
Sep 18 10:16:34  X.X.X.Y   SPOOF
Sep 18 10:17:21  X.X.X.Y   SPOOF
[root@ftp log]#
```

Figure 31 – Syslog Spoofed Kernel

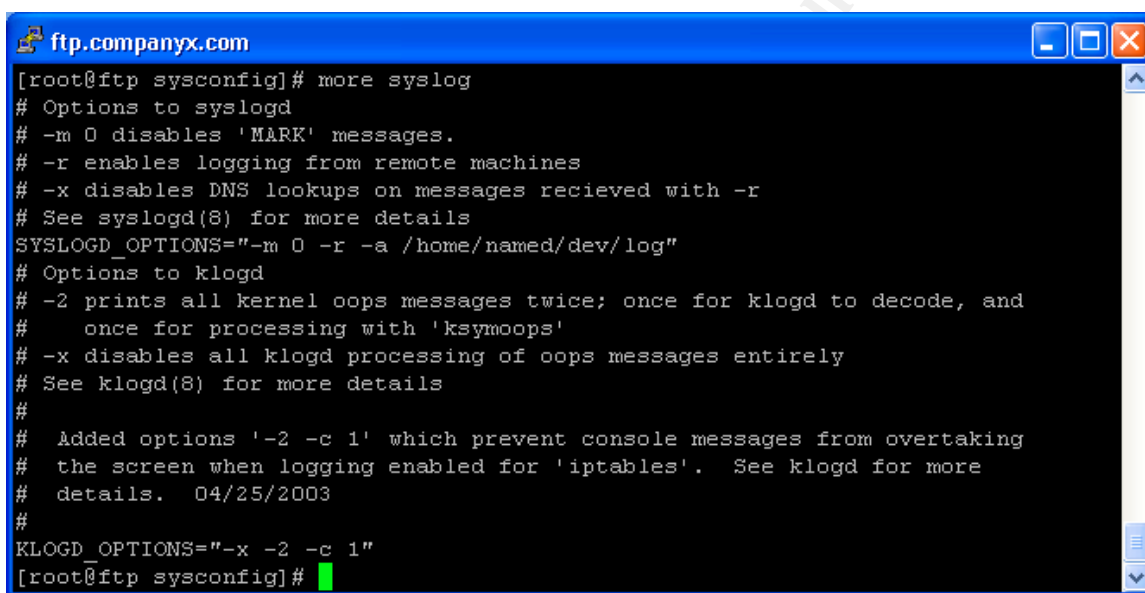


```
ftp.companyx.com
[root@ftp log]# tail -2 kernel
Sep 18 00:01:15 ftp kernel: Klogd 1.4.1, log source = /proc/kmsg started.
Sep 18 10:16:59  X.X.X.Y   SPOOF
[root@ftp log]#
```

P	3	<p>Verify that the BIND chroot environment can log to the syslog. The following option must be found in the <code>/etc/sysconfig/syslog</code> file:</p> <pre>SYSLOGD_OPTIONS="-m 0 -r -a /var/named/dev/log"</pre> <p>Ensure that named is logging properly to the <code>/var/log/messages</code> file by examining messages after a daemon restart. Restart the daemon as the root user by executing <code>"/etc/rc.d/init.d/named restart"</code>. Immediately examine the <code>/var/log/messages</code> file for new entries from the named daemon. This step is passed if the daemon is logging properly.</p>
---	---	---

I found that the system was configured to allow the chroot named daemon to log to the `/var/log/messages` file. Figures 32 and 33 show the results of the test.

Figure 32 – Named Syslog Configuration



```

ftp.companyx.com
[root@ftp sysconfig]# more syslog
# Options to syslogd
# -m 0 disables 'MARK' messages.
# -r enables logging from remote machines
# -x disables DNS lookups on messages recieved with -r
# See syslogd(8) for more details
SYSLOGD_OPTIONS="-m 0 -r -a /home/named/dev/log"
# Options to klogd
# -2 prints all kernel oops messages twice; once for klogd to decode, and
#   once for processing with 'ksymoops'
# -x disables all klogd processing of oops messages entirely
# See klogd(8) for more details
#
# Added options '-2 -c 1' which prevent console messages from overtaking
# the screen when logging enabled for 'iptables'. See klogd for more
# details. 04/25/2003
#
KLOGD_OPTIONS="-x -2 -c 1"
[root@ftp sysconfig]#

```

Figure 33 – Named Syslog Messages

```

root@ftp:~
[root@ftp root]# /etc/rc.d/init.d/named restart
Stopping named:
Starting named: [ OK ]
[root@ftp root]# tail /var/log/messages
Sep 19 09:37:24 ftp named[4402]: running
Sep 19 09:37:24 ftp named[4402]: zone zonec.com /IN: sending notifies (serial 2003082801)
Sep 19 09:37:24 ftp named[4402]: zone X.X.X.in-addr.arpa/IN: sending notifies (serial 2003082904)
Sep 19 09:37:24 ftp named[4402]: zone companyx.com /IN: sending notifies (serial 2003061701)
Sep 19 09:37:46 ftp named[4402]: lame server resolving '32.176.9.128.in-addr.arpa' (in '9.128.in-addr.arpa?'): 192.187.8.2#53
Sep 19 09:37:47 ftp named[4402]: shutting down: flushing changes
Sep 19 09:37:47 ftp named[4402]: stopping command channel on 127.0.0.1#953
Sep 19 09:37:47 ftp named[4402]: no longer listening on 127.0.0.1#53
Sep 19 09:37:47 ftp named[4402]: no longer listening on X.X.X.X #53
Sep 19 09:37:47 ftp named[4402]: exiting
[root@ftp root]#

```

P	4	<p>Examine the <code>/etc/syslog.conf</code> file on <code>ftp.companyx.com</code>. Ensure that logging from the Cisco router is directed to the <code>/var/log/ciscolog</code> file. The option should be:</p> <pre>local3.debug /var/log/ciscolog</pre> <p>Verify that the syslog daemon is capturing new alert entries in the <code>ciscolog</code> file by examining new input to the file. Execute the following to verify:</p> <pre>tail -f /var/log/ciscolog</pre> <p>The system will pass this step if new entries are written to the log.</p>
---	---	---

I also found that the syslog daemon had been correctly configured to capture the syslog alerts from the Cisco router. I did not include a screen capture of the tail output; I would have had to sanitize too much information for it to be useful. The system passed this step.

Figure 34 – Cisco Syslog Facility

```

ftp.companyx.com
[root@ftp /etc]# grep local3.debug syslog.conf
local3.debug                                /var/log/ciscolog
[root@ftp /etc]#

```

Checklist results: While the syslog daemon is correctly configured for logging alerts from named and the Cisco router, it is susceptible to spoofed alerts. Therefore, the system failed to pass this checklist.

System 11 Checklist Execution: Named Configuration

Steps for Testing Compliance: **F**

P/F	Step	Description of Step
F	1	Log in to the system using ssh, become the root user by executing “su -“, and obtain the BIND version number with the command “named -v”. Compare the version number to that of the latest release at http://www.isc.org . This system will pass this step if the version numbers are the same.

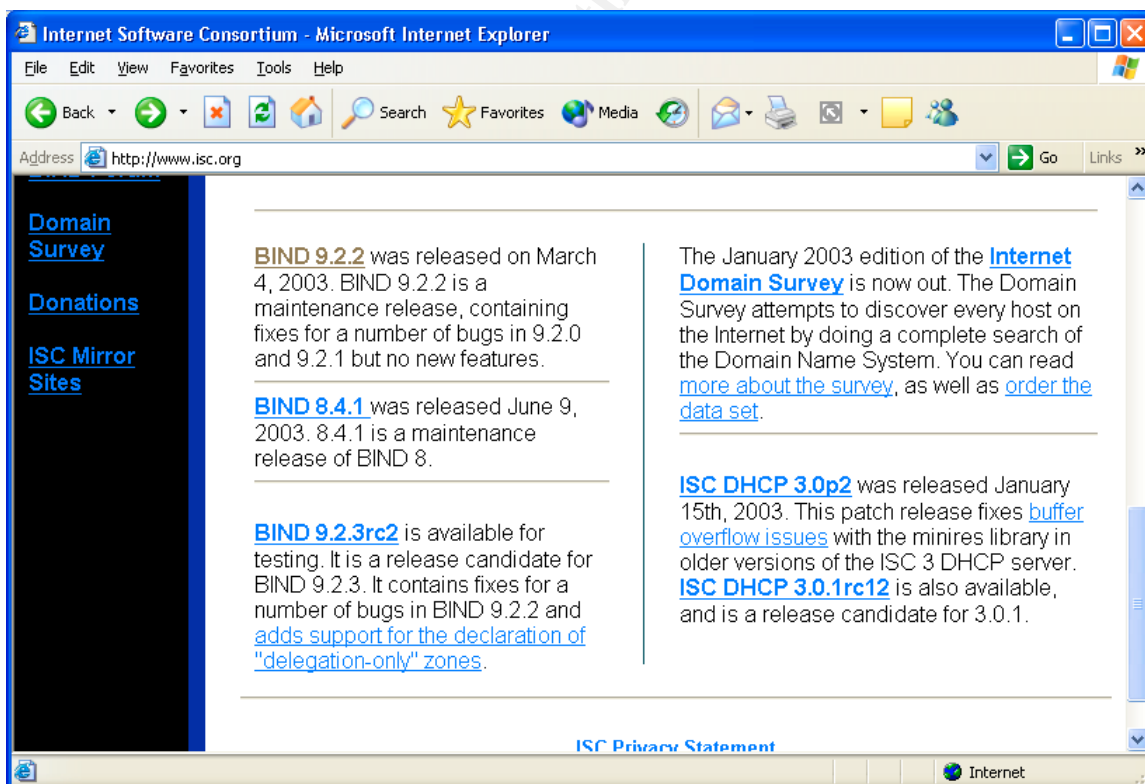
The system is running an older version of BIND than what is available at the ISC website. The BIND RPM on the system is version 9.2.1, while the ISC has version 9.2.2 available. The system did not pass this step. Figures 35 and 36 show the results of the tests.

Figure 35 – Bind Version



```
ftp.companyx.com
[root@ftp ~]# named -v
BIND 9.2.1
[root@ftp ~]#
```

Figure 36 – ISC Website



P	2	Ensure that the named daemon is running within the chroot environment. The <code>/etc/sysconfig/named</code> file should have the following variable definition:
---	---	--

		<p>“ROOTDIR=/home/named”</p> <p>The directory /var/named should be a symbolic link of the /home/named directory. This can be checked by using the “ls -al /var/named” command. The actual files reside in the /home partition because it is, by far, the largest partition on the system. The auditor can verify which daemon is currently being executed by running the command “ps -aux grep named”. The system will pass this step if the results are “/home/named/usr/sbin/named -u named -t /home/named”.</p>
--	--	--

Using the procedures provide above, I confirmed that the named daemon is running in the proper chroot environment. Figure 37 is the output from the commands.

Figure 37 – BIND chroot

```

ftp.companyx.com
[root@ftp ~]# grep ROOTDIR /etc/sysconfig/named
# ROOTDIR="/some/where" -- will run named in a chroot environment.
#                               at startup. Don't add -t here, use ROOTDIR instead.
ROOTDIR=/home/named
[root@ftp ~]# ls -al /var/named
lrwxrwxrwx  1 root  root      11 Apr 29 11:28 /var/named -> /home/named
[root@ftp ~]# ps -aux | grep named
root      850  0.2  0.5 1656 692 ?        S   10:56   1:38 syslogd -m 0 -r -a /home/named/dev/log
named     915  0.1  1.7 30312 2168 ?      S   10:56   0:56 /home/named/usr/sbin/named -u named -t /home/named
root     3087  0.0  0.5 3576 640 pts/0    S   22:23   0:00 grep named
[root@ftp ~]#

```

P	3	<p>Verify that zone transfers are set to “restricted” in the /var/named/var/etc/named.conf file. Each DNS zone should have an “allow-transfer” section with the addresses of Company X’s internet provider DNS servers listed.</p>
---	---	--

I examined the named.conf file and found that it was correctly configured for zone transfers. Only Company X’s internet provider and the localhost are authorized to transfer the zone files. Figure 38 shows a sanitized version of the /var/named/etc/named.conf file.

Figure 38 – named.conf zone transfer



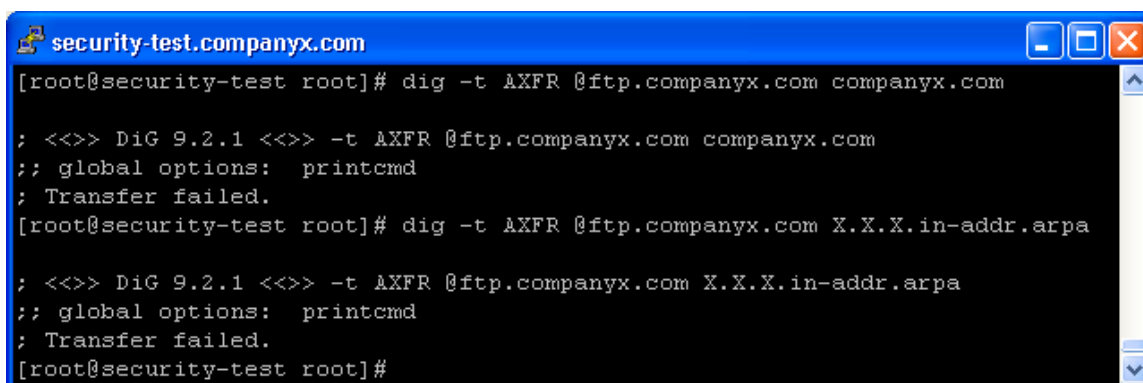
```
ftp.companyx.com
zone " zonec.com " {
    type master;
    file " zonec.com ";
    allow-update { none; };
    allow-transfer {
        127.0.0.1;          // localhost
        X.X.X.X;
        X.X.X.Y;
        X.X.X.Z;
    };
};

zone "         companyx.com         " {
    type master;
    file "         companyx.com         ";
    allow-update { none; };
    allow-transfer {
        127.0.0.1;          // localhost
        X.X.X.X;
        X.X.X.Y;
        X.X.X.Z;
    };
};
```

P	4	Ensure that restricted zone transfers are enabled by attempting to transfer zones using the security-test Linux system as the source. Log in to the security-test system, and utilize dig to transfer several DNS zones. The following command should be used twice: “dig -t AXFR @ftp.companyx.com <zone>”, where <zone> is first the forward zone, and second is the reverse zone. The results should be “Transfer failed.”
---	---	---

My attempts to transfer zones from the ftp.companyx.com system failed. The results of the test can be found in figure 39.

Figure 39 – Zone transfer attempt



```
security-test.companyx.com
[root@security-test root]# dig -t AXFR @ftp.companyx.com companyx.com

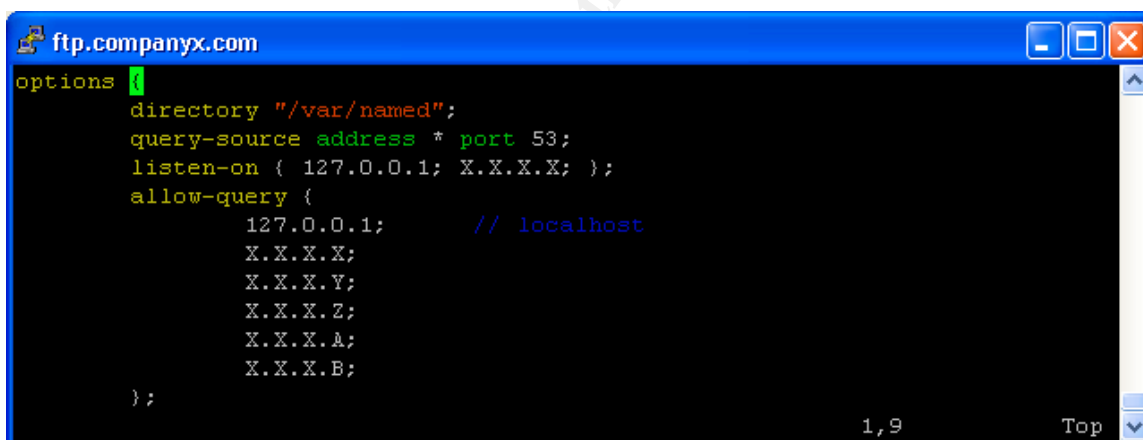
; <<>> DiG 9.2.1 <<>> -t AXFR @ftp.companyx.com companyx.com
;; global options: printcmd
; Transfer failed.
[root@security-test root]# dig -t AXFR @ftp.companyx.com X.X.X.in-addr.arpa

; <<>> DiG 9.2.1 <<>> -t AXFR @ftp.companyx.com X.X.X.in-addr.arpa
;; global options: printcmd
; Transfer failed.
[root@security-test root]#
```

P	5	Ensure that restricted DNS queries are enabled by examining the “allow-query” section in the /var/named/var/etc/named.conf file. This block should include Company X’s external hosts (that require recursive lookups) and Company X’s internet provider DNS servers.
---	---	---

I found that the `named.conf` file was correctly configured. It only allowed queries from external Company X hosts. Figure 40 is a sanitized version of the `named.conf` file. The system passed this step.

Figure 40 – named.conf allow-query



```
ftp.companyx.com
options {
    directory "/var/named";
    query-source address * port 53;
    listen-on { 127.0.0.1; X.X.X.X; };
    allow-query {
        127.0.0.1;          // localhost
        X.X.X.X;
        X.X.X.Y;
        X.X.X.Z;
        X.X.X.A;
        X.X.X.B;
    };
};
```

P	6	Verify that restricted DNS queries are functioning properly by attempting to resolve DNS queries from the security-test Linux system. Log in to the security-test system, and execute the following command: “dig @ftp.companyx.com www.companyx.com”. If DNS queries are restricted, the result should be a “status: REFUSED” message.
---	---	---

The result of the query from the security-tests system was “REFUSED.” The system’s `named` configuration for limiting queries is functioning correctly. Figure 41 shows the results of the test.

Figure 41 – DNS query attempt



```
[root@security-test root]# dig @ftp.companyx.com www.companyx.com

; <<>> DiG 9.2.1 <<>> @ftp.companyx.com www.companyx.com
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: REFUSED, id: 64552
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.companyx.com.                IN      A

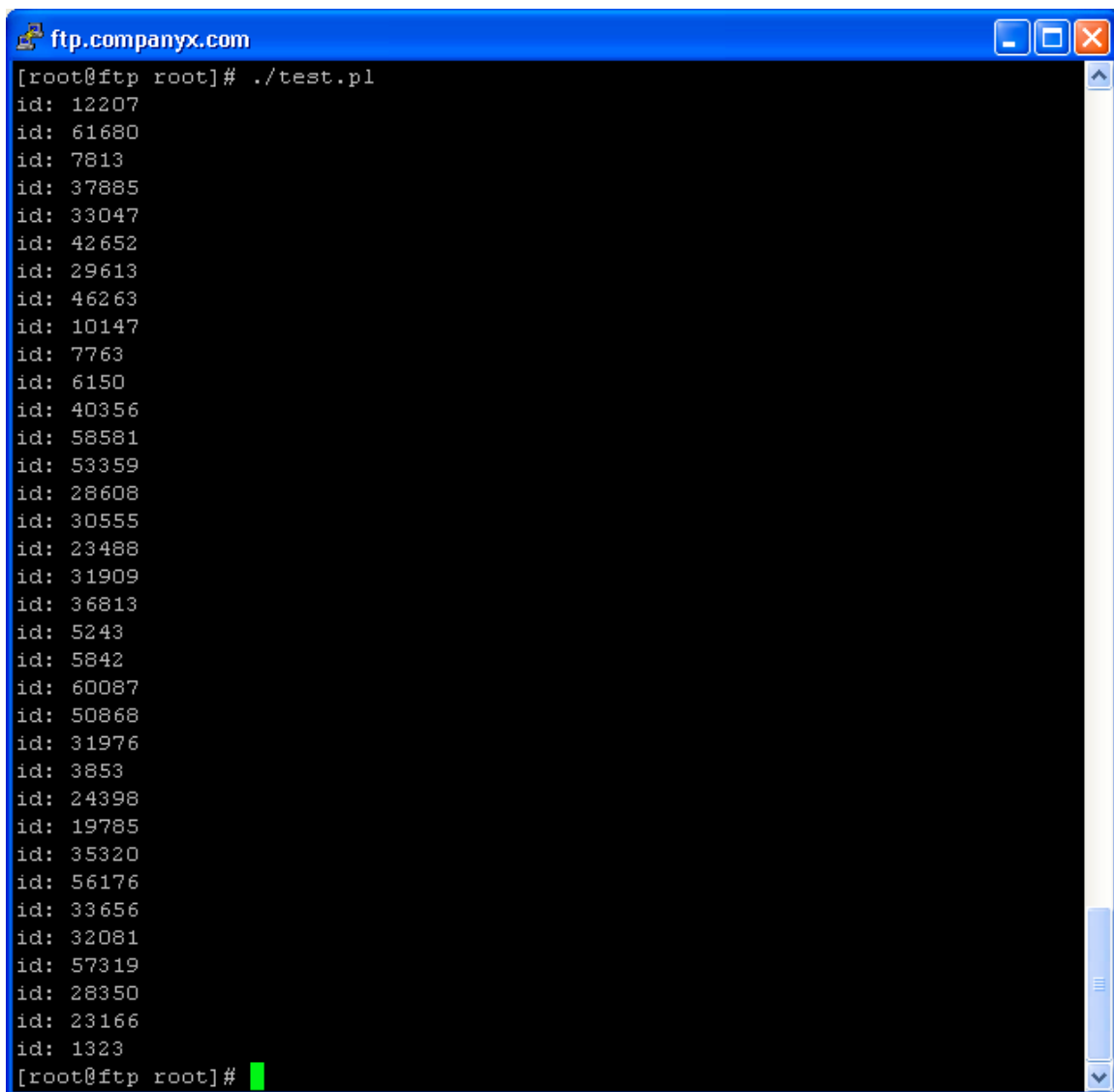
;; Query time: 2 msec
;; SERVER:      X.X.X.X #53 (ftp)
;; WHEN: Thu Sep 18 22:55:28 2003
;; MSG SIZE  rcvd: 32

[root@security-test root]#
```

P 7	<p>Ensure that query identification numbers (ids) are arbitrary by executing 35 queries in succession. Log in to the ftp.companyx.com system and execute the following script:</p> <pre>#!/usr/bin/perl \$iterations = 35; for (\$i = 0; \$i < \$iterations; \$i++) { (\$id) = (`dig \@ftp\.companyx\.com www\.netscape\.com` =~ /id: \d+/g); print "\$id\n"; }</pre> <p>Inspect the output from the above script to ensure that the id numbers are non-sequential.</p>
--------	--

When executing the provided perl code, I found the query id numbers to be quite random. It would definitely not be a trivial task to deduce a pattern from them. The system passed this step. I have provided the results as figure 42.

Figure 42 – DNS query id numbers



```
ftp.companyx.com
[root@ftp root]# ./test.pl
id: 12207
id: 61680
id: 7813
id: 37885
id: 33047
id: 42652
id: 29613
id: 46263
id: 10147
id: 7763
id: 6150
id: 40356
id: 58581
id: 53359
id: 28608
id: 30555
id: 23488
id: 31909
id: 36813
id: 5243
id: 5842
id: 60087
id: 50868
id: 31976
id: 3853
id: 24398
id: 19785
id: 35320
id: 56176
id: 33656
id: 32081
id: 57319
id: 28350
id: 23166
id: 1323
[root@ftp root]#
```

Checklist results: The system has failed this checklist because of the BIND server version. The administrative team must complete more research before deciding if the named daemon should be upgraded. Otherwise, all of the configuration items were correctly configured and functioning properly.

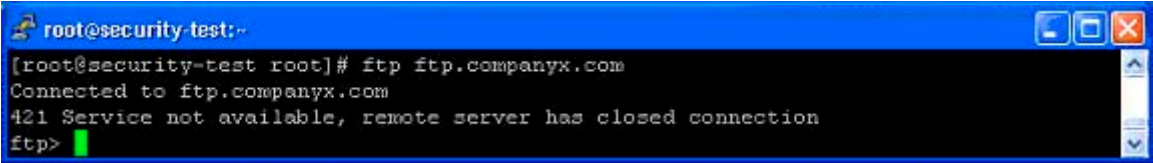
System 13 Checklist Execution: FTP Configuration

Steps for Testing Compliance: P		
P/F	Step	Description of Step
P	1	Connect to the ftp.companyx.com FTP service from a machine that is not permitted access. Examine the error message displayed for information regarding versions or software packages.

		The system will pass this step if the information returned is non-specific.
--	--	---

The system responded to the test with a generic error message, it did not provide any version information.

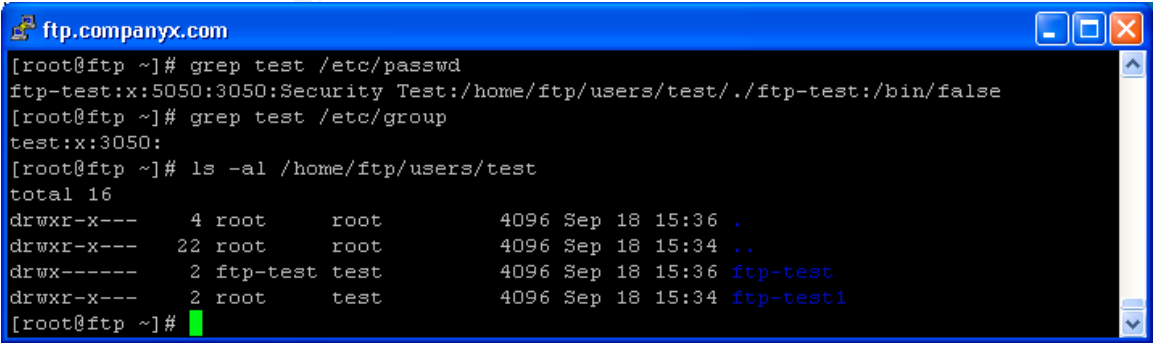
Figure 43 – Unauthorized FTP version



P	2	<p>Log in to the ftp.companyxx.com system using ssh and execute “su -” to become the root user. Create a test user account on the system using the following procedures:</p> <ul style="list-style-type: none"> o mkdir /home/ftp/users/test o mkdir /home/ftp/users/test/ftp-test1 o mkdir /home/ftp/users/test/ftp-test o Edit the /etc/group and add a group entry for security-test with gid 3050 o useradd -u 5050 -g 3050 -c “Security Test” -s /bin/false -d /home/ftp/users/test/./ftp-test ftp-test o chown root:test /home/ftp/users/test/ftp-test1 o passwd ftp-test <p>Proceed through step four to complete this part of the test.</p>
---	---	--

This step is preparation for the tests in step four of this checklist. The following screen capture shows the results of the user addition procedures.

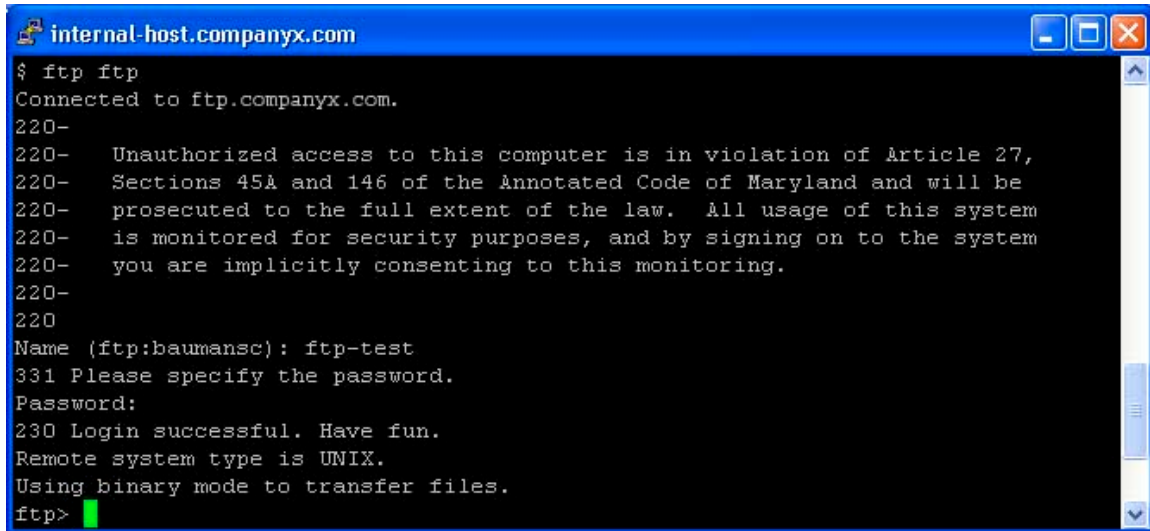
Figure 44 – ftp-test user



P	3	Log in to the ftp.companyxx.com FTP server using the ftp-test account created in step two. Examine the log in process for information that may reveal the FTP server version.
---	---	---

The login process for a legitimate user did not reveal any sensitive about the system. This system passed this step.

Figure 45 – Authorized FTP version



```
$ ftp ftp
Connected to ftp.companyx.com.
220-
220-  Unauthorized access to this computer is in violation of Article 27,
220-  Sections 45A and 146 of the Annotated Code of Maryland and will be
220-  prosecuted to the full extent of the law.  All usage of this system
220-  is monitored for security purposes, and by signing on to the system
220-  you are implicitly consenting to this monitoring.
220-
220
Name (ftp:baumansc): ftp-test
331 Please specify the password.
Password:
230 Login successful. Have fun.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

P	4	<p>Attempt to traverse directories and glean information from the system. Utilize the following tests:</p> <ul style="list-style-type: none">○ Examine file and directory listings for user ids and group ids. The vsftpd daemon should rewrite all uids and gids as the user and group ftp.○ Execute a “pwd” to find out what the root directory is for the ftp-test user.○ Change the working directory to the top most directory, and try to change directory beyond the top directory. To do this, execute three “cd ..” commands. Execute a “pwd” and an “ls” to determine the current working directory.○ Attempt to change directory to /test and /home/ftp/users.
---	---	--

The system passed all of the checks in step four of this checklist. The vsftpd daemon successfully hid the user and group names associated with the directories from the ftp-test user. In addition, the chroot environment kept the ftp-test user from traversing beyond the /home/ftp/users/test directory, while the ftp-test user did not even know that this directory existed (All the user could see was /). The only directories the ftp-test user can enter, write to, or read from were the subdirectories of/home/ftp /users /test. The following screen capture displays the results.

Figure 46 – FTP directory traversal

```

internal-host.companyx.com
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
226 Directory send OK.
ftp> pwd
257 "/ftp-test"
ftp> cd ..
250 Directory successfully changed.
ftp> cd ..
250 Directory successfully changed.
ftp> cd ..
250 Directory successfully changed.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwx----- 2 ftp      ftp      4096 Sep 18 19:36 ftp-test
drwxr-x--- 2 ftp      ftp      4096 Sep 18 19:34 ftp-test1
226 Directory send OK.
ftp> pwd
257 "/"
ftp> cd /home/ftp/users
550 Failed to change directory.
ftp> cd /test
550 Failed to change directory.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwx----- 2 ftp      ftp      4096 Sep 18 19:36 ftp-test
drwxr-x--- 2 ftp      ftp      4096 Sep 18 19:34 ftp-test1
226 Directory send OK.
ftp>

```

P	5	Manually examine the <code>/etc/passwd</code> file on the <code>ftp.companyx.com</code> system to ensure that all user accounts are created with the <code>chroot jail</code> option listed in the home directory. (e.g. <code>baumansc:x:4002:3000:Sean Baumann</code> <code>x3342:/home/ftp/users/./security/baumansc:/bin/ksh</code>)
---	---	--

I manually examined the file, and verified that all user accounts were configured with the proper `chroot` syntax. I have not provided a screenshot for this step, this is a trivial visual verification.

P	6	Manually examine the permissions for subdirectories of the <code>/home/ftp/users</code> directory. Subdirectories should not contain any “other” permission. Execute the command “ <code>find /home/ftp/users/ -perm +o=rwx -type d -print</code> ”. The auditor can ignore the <code>/home/ftp/users</code> and the <code>/home/ftp/users/lost+found</code> directories for this test. These directories require “other” permissions. The system is compliant if all subdirectories have no permissions for the “others”.
---	---	--

I found that the proper permissions had been assigned to all subdirectories of `/home/ftp/users`.

Figure 47 – “Other” permissions

```
ftp.companyx.com
[root@ftp users]# find /home/ftp/users -perm +o=rwx -type d -print
/home/ftp/users
/home/ftp/users/lost+found
[root@ftp users]#
```

Checklist results: The system is compliant with the checklist objects listed.

System 14 Checklist Execution: User Quotas

Steps for Testing Compliance: F		
P/F	Step	Description of Step
P	1	Log in to the ftp.companyx.com system from a system with the proper hosts.allow permissions; use the auditor’s own ftp account. Transfer large files to the system, while monitoring the disk usage.
F	2	Attempt to exhaust disk space by transferring large files to the system. If the /home file system becomes exhausted by one user, then the system fails this checklist item.

I was able to exhaust the /home file system by transferring large files to the system. The system has failed this step. The following screen capture shows the files transferred during the process.

Figure 48 – FTP large files

```
root@ftp:/home/ftp/users/security/baumansc
-rw-r----- 1 ftp ftp 38960035 Sep 18 20:39 data8
-rw-r----- 1 ftp ftp 38960035 Sep 18 20:39 data9
226 Directory send OK.
ftp> ls
227 Entering Passive Mode (127,0,0,1,164,205)
150 Here comes the directory listing.
-rw-r----- 1 ftp ftp 38960035 Sep 18 20:37 data1
-rw-r----- 1 ftp ftp 38960035 Sep 18 20:39 data10
-rw-r----- 1 ftp ftp 935040840 Sep 18 20:44 data11
-rw-r----- 1 ftp ftp 1012960910 Sep 18 20:47 data12
-rw-r----- 1 ftp ftp 1012960910 Sep 18 20:49 data13
-rw-r----- 1 ftp ftp 1012960910 Sep 19 03:43 data14
-rw-r----- 1 ftp ftp 910364672 Sep 19 03:46 data15
-rw-r----- 1 ftp ftp 38960035 Sep 18 20:38 data2
-rw-r----- 1 ftp ftp 77920070 Sep 18 20:40 data3
-rw-r----- 1 ftp ftp 38960035 Sep 18 20:39 data4
-rw-r----- 1 ftp ftp 116880105 Sep 18 20:41 data5
-rw-r----- 1 ftp ftp 194800175 Sep 18 20:41 data6
-rw-r----- 1 ftp ftp 311680280 Sep 18 20:42 data7
-rw-r----- 1 ftp ftp 38960035 Sep 18 20:39 data8
-rw-r----- 1 ftp ftp 38960035 Sep 18 20:39 data9
226 Directory send OK.
ftp>
```

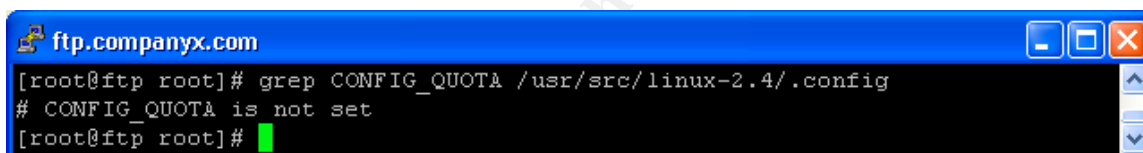
P	3	<p>If the <code>/home</code> file system can be completely filled, ensure that the system is remains functional.</p> <ul style="list-style-type: none"> ○ Attempt to log in to the system using ssh. ○ Attempt to remove files.
---	---	---

I was able to gain remote root access to the system and remove the files. Only the `/home` partition was filled.

F	4	<p>Log in to the <code>ftp.companyx.com</code> system, and execute “<code>su -</code>” to become the root account. Examine the kernel settings for user quota support. View the <code>/usr/src/kernel-2.4/.config</code> file and search for “<code>CONFIG_QUOTA</code>”. If this option is enabled in the kernel, the option will be set to “<code>y</code>.” This system passes this step if this setting is enabled.</p>
---	---	---

When I started researching for this audit, I was under the impression the administrative team had implemented a quota system. Unfortunately, that was not the case. While it is a requirement of the system, from a system availability and security standpoint, it was not included in the production system. The system failed this item. Figure 49 shows that the kernel option is not enabled.

Figure 49 – Quota kernel option

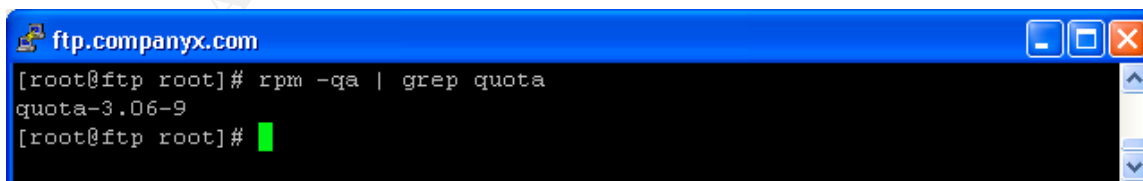


```
ftp.companyx.com
[root@ftp root]# grep CONFIG_QUOTA /usr/src/linux-2.4/.config
# CONFIG_QUOTA is not set
[root@ftp root]#
```

P	5	<p>Execute “<code>rpm -qa grep quota</code>” to determine if quota software has been installed on the system. If the software is present on the system, then the system passes this step.</p>
---	---	---

The quota software was included in the system build, but it is not used. The system passes this step for having the RPM installed.

Figure 50 – Quota Version



```
ftp.companyx.com
[root@ftp root]# rpm -qa | grep quota
quota-3.06-9
[root@ftp root]#
```

F	6	<p>Examine the <code>/etc/fstab</code> file for the “<code>usrquota</code>” option for the <code>/home</code> file system.</p>
---	---	--

Since the system kernel does not support user quotas, there was no reason for the administrative team to mount any file systems using the “quota” option. The system failed this test.

Figure 51 – /etc/fstab quota option

```

ftp.companyx.com
[root@ftp root]# cat /etc/fstab
/dev/ida/c0d0p1      /               ext3      defaults    1 1
/dev/ida/c0d0p5      /var            ext3      defaults    1 2
/dev/ida/c0d0p6      /home           ext3      defaults    1 2
none                /dev/pts        devpts    gid=5,mode=620 0 0
none                /proc           proc       defaults    0 0
none                /dev/shm        tmpfs     defaults    0 0
/dev/ida/c0d0p3      swap            swap      defaults    0 0
/dev/cdrom           /cdrom          udf,iso9660 noauto,owner,kudzu,ro 0 0
/dev/fd0             /floppy         auto      noauto,owner,kudzu 0 0
[root@ftp root]#

```

Checklist results: There was no need to continue executing the checklist; the system has failed to meet the objectives. The system has the proper quota software installed, however the kernel was compiled without quota support and none of the file systems have been mounted with the quota option. As a result, a user could easily exhaust the free drive space and cause a DoS of the FTP service.

System 15 Checklist Execution: System File Integrity

Steps for Testing Compliance: P		
P/F	Step	Description of Step
P	1	Log in to the ftp.companyx.com system and become the root user by executing “su -”. Verify that the tripwire software is installed by executing “rpm -qa grep tripwire”.

The tripwire software is present on the system, loaded as an RPM from RedHat.

Figure 52 – /etc/fstab quota option

```

ftp.companyx.com
[root@ftp ~]#
[root@ftp ~]# rpm -qa | grep tripwire
tripwire-2.3.1-17
[root@ftp ~]#

```

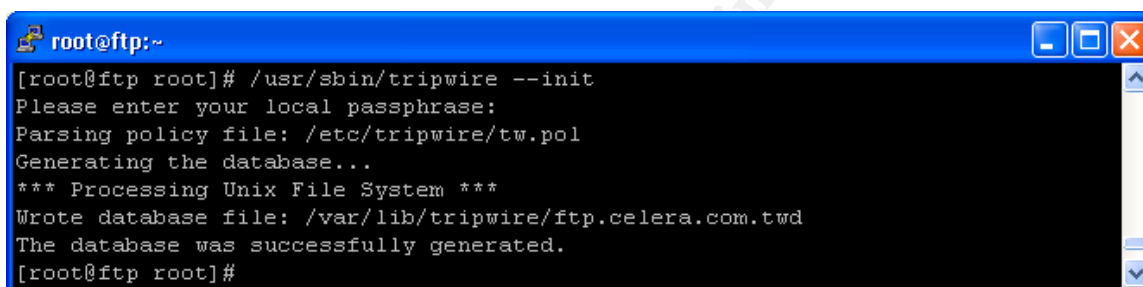
P	2	Examine and visually compare the /etc/twpol.txt file with the baseline copy saved on the management system. The file is short and organized enough to inspect this manually.
---	---	--

I visually inspected the files as described in step two. The file contents were identical. Tripwire was configured to monitor the correct files and directories, as determined by the administrative team at the beginning of the system's production lifecycle. A list of all monitoring options is beyond the scope of this assignment, so I have not included a screenshot of the configuration (See the RedHat configuration guide for more details).

P	3	Reinitialize the tripwire database to keep extraneous data from interfering with results. Execute <code>"/usr/sbin/tripwire --init"</code> . Utilize the local passphrase to begin the initialization. If the initialization is completed successful, the system passes this step.
---	---	--

I was able to reinitialize the database using the provided command. Figure 53 shows the results of the command execution.

Figure 53 – Initialize tripwire DB



```

root@ftp:~
[root@ftp root]# /usr/sbin/tripwire --init
Please enter your local passphrase:
Parsing policy file: /etc/tripwire/tw.pol
Generating the database...
*** Processing Unix File System ***
Wrote database file: /var/lib/tripwire/ftp.celera.com.twd
The database was successfully generated.
[root@ftp root]#

```

P	4	<p>Alter several files on the system and rerun the integrity check. These changes will ensure that multiple functions of the tripwire software are reliable.</p> <ul style="list-style-type: none"> o "touch" the <code>/etc/hosts</code> file o Create the directory <code>/home/test</code> o Rotate the log files in <code>/var/log</code> by executing <code>"logrotate -f /etc/logrotate.conf"</code>. o Rerun the integrity check by executing <code>"/usr/sbin/tripwire --check"</code>. <p>Identify the changes in the generated report. The system has passed this step if the report indicates that the above changes were made.</p>
---	---	--

I altered the files as described in step four. I received errors when executing the "logrotate" command; however, I verified that they still rotated. The output from the tripwire check, provided below, indicated that the software was functioning correctly. It detected the log rotations, the creation of a new directory in `/home` and the date change on the `/etc/hosts` file. These tests were designed specifically for the tripwire policy implemented on `ftp.companyx.com`. The system has passed this step.

Figure 54 – rotate logs

```
ftp.companyx.com
[root@ftp /etc]# logrotate -f /etc/logrotate.conf
/tmp/logrotate.KcuMdu: line 4: /usr/sbin/killall: No such file or directory
error running postrotate script
/tmp/logrotate.OdpXRC: line 4: /usr/sbin/killall: No such file or directory
error running postrotate script
[root@ftp /etc]#
```

```
[root@ftp ~]# /usr/sbin/tripwire --check
Parsing policy file: /etc/tripwire/tw.pol
*** Processing Unix File System ***
Performing integrity check...
Wrote report file: /var/lib/tripwire/report/ftp.companyx.com-20030918-
181350.twr
```

Tripwire(R) 2.3.0 Integrity Check Report

```
Report generated by:      root
Report created on:       Thu 18 Sep 2003 06:13:50 PM EDT
Database last updated on: Never
```

Report Summary:

```
Host name:                ftp.companyx.com
Host IP address:          X.X.X.X
Host ID:                  None
Policy file used:         /etc/tripwire/tw.pol
Configuration file used:  /etc/tripwire/tw.cfg
Database file used:       /var/lib/tripwire/ftp.companyx.com.twd
Command line used:        /usr/sbin/tripwire --check
```

Rule Summary:

Section: Unix File System

Rule Name	Severity Level	Added	Removed
Modified			
Tripwire Data Files	0	0	0

* Monitor Filesystems	0	1	0	0
OS Binaries and Libraries	0	0	0	0
Tripwire Binaries	0	0	0	0
User Binaries and Libraries	0	0	0	0
Temporary Directories	0	0	0	0
RPM Checksum Files	0	0	0	0
* System Boot Changes	0	1	0	
57				
OS Devices and Misc Directories	0	0	0	0
* Global Configuration Files	0	0	0	2
OS Boot Files and Mount Points	0	0	0	0
Root Directory and Files	0	0	0	0

Total objects scanned: 101331

Total violations found: 61

=====
Object Summary:
=====

Section: Unix File System

Rule Name: System Boot Changes (/var/log)
Severity Level: 0

Added:

"/var/log/up2date.3.gz"

Modified:

"/var/log/boot.log"
"/var/log/boot.log.11.gz"
"/var/log/boot.log.3.gz"
"/var/log/boot.log.5.gz"
"/var/log/boot.log.7.gz"
"/var/log/boot.log.9.gz"
"/var/log/cron"
"/var/log/cron.1.gz"
"/var/log/cron.10.gz"
"/var/log/cron.11.gz"
"/var/log/cron.2.gz"
"/var/log/cron.4.gz"
"/var/log/cron.6.gz"
"/var/log/cron.7.gz"
"/var/log/iplog"
"/var/log/iplog.1"
"/var/log/iptables"
"/var/log/iptables.1"

```
"/var/log/kernel"
"/var/log/kernel.10.gz"
"/var/log/kernel.12.gz"
"/var/log/kernel.4.gz"
"/var/log/kernel.6.gz"
"/var/log/loginlog"
"/var/log/loginlog.10.gz"
"/var/log/loginlog.3.gz"
"/var/log/loginlog.5.gz"
"/var/log/loginlog.6.gz"
"/var/log/loginlog.8.gz"
"/var/log/maillog"
"/var/log/maillog.1.gz"
"/var/log/maillog.11.gz"
"/var/log/maillog.3.gz"
"/var/log/maillog.5.gz"
"/var/log/maillog.9.gz"
"/var/log/messages"
"/var/log/messages.1.gz"
"/var/log/messages.12.gz"
"/var/log/messages.3.gz"
"/var/log/messages.8.gz"
"/var/log/rpmpkgs"
"/var/log/secure"
"/var/log/secure.11.gz"
"/var/log/secure.3.gz"
"/var/log/secure.5.gz"
"/var/log/secure.6.gz"
"/var/log/secure.7.gz"
"/var/log/secure.8.gz"
"/var/log/syslog"
"/var/log/syslog.11.gz"
"/var/log/syslog.12.gz"
"/var/log/syslog.4.gz"
"/var/log/syslog.6.gz"
"/var/log/syslog.8.gz"
"/var/log/up2date"
"/var/log/up2date.2.gz"
"/var/log/wtmp"
```

```
-----
-----
```

```
Rule Name: Global Configuration Files (/etc)
Severity Level: 0
```

```
-----
-----
```

```
Modified:
"/etc/hosts"
"/etc/ntp/drift"
```

```
-----
-----
```

```
Rule Name: Monitor Filesystems (/home)
Severity Level: 0
```

```
-----
-----
```

```
Added:
"/home/test"
```

```
=====
=====
```

```
Error Report:
```

```
=====
=====
```

```
No Errors
```

```
-----
-----
```

```
*** End of report ***
```

```
Tripwire 2.3 Portions copyright 2000 Tripwire, Inc. Tripwire is a
registered
trademark of Tripwire, Inc. This software comes with ABSOLUTELY NO
WARRANTY;
for details use --version. This is free software which may be
redistributed
or modified only under certain conditions; see COPYING for details.
All rights reserved.
Integrity check complete.
[root@ftp ~]#
```

P	5	Update the tripwire database using the report generated in step four. Execute the command <code>"/usr/sbin/tripwire --update --twrfile /var/lib/tripwire/report/<file>.twr"</code> , where <code><file></code> is the name of the report generated in step four. If the procedure has correctly updated the database, then the system has passed this step.
---	---	---

The tripwire update completed flawlessly. It merged the changes found in step four with the existing tripwire database. An administrator would use this command on a regular basis to “accept” changes that have been made to the system. Figure 55 provides a screen capture of the process.

Figure 55 – Tripwire DB update

```

ftp.companyxx.com
OS Boot Files and Mount Points 0 0 0 0
Root Directory and Files 0 0 0 0

Total objects scanned: 101331
Total violations found: 61

=====
Object Summary:
=====

# Section: Unix File System

-----

Rule Name: System Boot Changes (/var/log)
Severity Level: 0

-----

Remove the "x" from the adjacent box to prevent updating the database
with the new values for this object.

Added:
[x] "/var/log/up2date.3.gz"

Modified:
[x] "/var/log/boot.log"
[x] "/var/log/boot.log.11.gz"
[x] "/var/log/boot.log.3.gz"
[x] "/var/log/boot.log.5.gz"
[x] "/var/log/boot.log.7.gz"
[x] "/var/log/boot.log.9.gz"
[x] "/var/log/cron"
[x] "/var/log/cron.1.gz"
[x] "/var/log/cron.10.gz"
[x] "/var/log/cron.11.gz"
[x] "/var/log/cron.2.gz"
[x] "/var/log/cron.4.gz"
[x] "/var/log/cron.6.gz"
[x] "/var/log/cron.7.gz"
[x] "/var/log/iplog"
[x] "/var/log/iplog.1"
[x] "/var/log/iptables"
[x] "/var/log/iptables.1"
[x] "/var/log/kernel"
[x] "/var/log/kernel.10.gz"
[x] "/var/log/kernel.12.gz"
  
```

P	6	Remove the /home/test directory and reinitialize the database with the command "/usr/sbin/tripwire -init" (<i>trivial step, no screen capture included</i>)
---	---	---

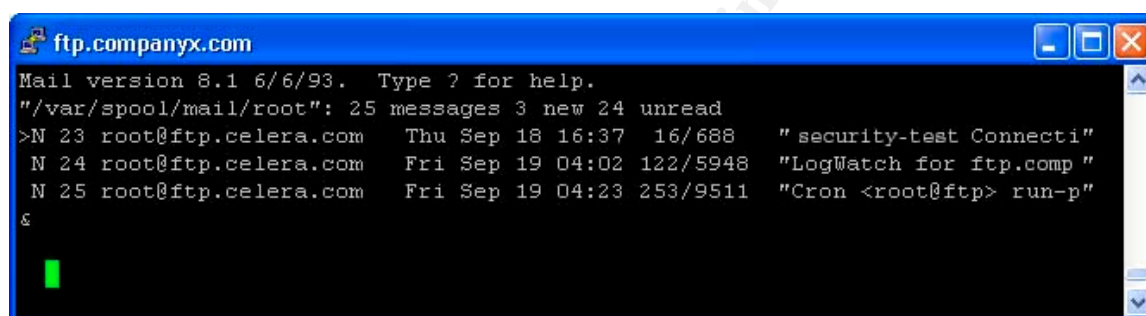
I was also able to complete the second database initialize. This step is for "clean-up" of changes made during step four. I did not include a screen shot

because the output was very similar to what is show in Figure 53. The system passed this step.

P	7	Examine the root user's email box to determine if the administrative team has processed, and acted upon, the previously generated tripwire reports. If the reports are unread, then this step is failed, and processes and procedures should be altered to include tripwire monitoring.
---	---	---

By examining the root user's email box, I found that the administrative team has been keeping up with the automatically generated tripwire reports. However, this check does not guarantee that the reports have been processed, just that the reports have been deleted. A quick interview with the administrative team members revealed that the reports are read on a regular basis. Note in figure 56, message number twenty-five is the tripwire report from the previous day; all other reports had been processed. The system has passed this step.

Figure 56 – Root's email



Checklist results: The system has met all of the defined control objectives for file integrity. The administrative team consistently maintains the tripwire database, and all tripwire processes are functioning correctly. The system has passed this audit item.

System 17 Checklist Execution: IP Tables

Steps for Testing Compliance: F		
P/F	Step	Description of Step
F	1	The purpose of the system is clearly defined within the SLA and the system documentation. The auditor should examine the IPTables security policy file, /usr/local/etc/iptables, to determine if the minimal number of services is permitted. Those services should include ssh, telnet, ftp, dns, and ntp (which were the services identified under the checklist for daemons and open ports). NAT services are not required, so they can be ignored.

I carefully examined the IPTables policy file, and compared its contents with the well known access requirements of the ftp.companyx.com system. I derived the access requirements from the SLA statement and the administrative team's

written and verbal processes. I found that the implementation of IPTables on the system was useless. A policy of permit everything, and only deny X traffic was currently implemented. The original system designer indicated that the IPTables implementation on the system was only for basic testing, and that the administrative team did not have time to create a fully functional access policy before the system was “productionized.” The original intention was to provide a test for the interactivity between the IPTables software and the rest of the system’s services. The system designer also indicated that it was his intention to develop the use of IPTables in future releases of Company X’s Linux template. The system has failed this step, since the administrative team has not configured the IPTables software to deny all traffic, except for what is specifically required. Figure 57 shows an excerpt of the /usr/local/etc/iptables policy file.

Figure 57 – IPTables file

```
ftp.companyx.com
IPTABLES=/sbin/iptables

#
# Set default policy
#
$IPTABLES --table filter --policy INPUT ACCEPT
$IPTABLES --table filter --policy OUTPUT ACCEPT
$IPTABLES --table filter --policy FORWARD DROP

$IPTABLES --table nat --policy PREROUTING ACCEPT
$IPTABLES --table nat --policy POSTROUTING ACCEPT
$IPTABLES --table nat --policy OUTPUT ACCEPT

$IPTABLES --table mangle --policy PREROUTING ACCEPT
$IPTABLES --table mangle --policy OUTPUT ACCEPT

#
--More-- (52%)
```

P	2	Verify that the system starts IPTables at boot. Execute “chkconfig --list” and identify that iptables is enabled for init levels two through five.
---	---	--

The system starts IPTables at boot, which I have verified in Figure 58.

Figure 58 – IPTables chkconfig

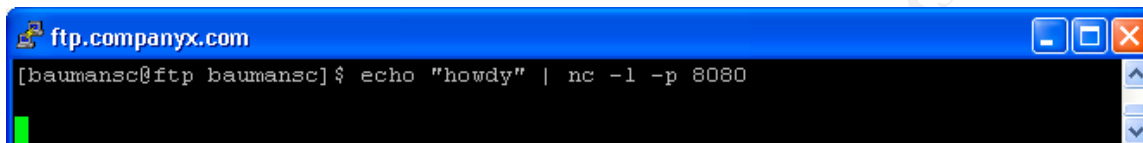
```
root@ftp:~
[root@ftp root]# chkconfig --list | grep iptables
iptables          0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@ftp root]#
```

F	3	Verify that IPTables is blocking inbound traffic by utilizing netcat. Set up a netcat listener on the ftp.companyx.com system by executing “echo “Howdy” nc -l -p 8080”. From the security-test system, connect to the ftp.companyx.com using telnet on port 8080. If the
---	---	---

	message “Howdy” is displayed, then the IPTables firewall is not functioning properly.
--	---


This test proved that the IPTables software is not providing any real protection for the system. As a normal system user, I was able to create a listener on port 8080. From the security-test system, I was able to connect to the listener. The system has failed this step. Figures 59 and 60 show the results of the test.

Figure 59 – Netcat listener



```
ftp.companyx.com
[baumansc@ftp baumansc]$ echo "howdy" | nc -l -p 8080
```

Figure 60 – Connecting to listener

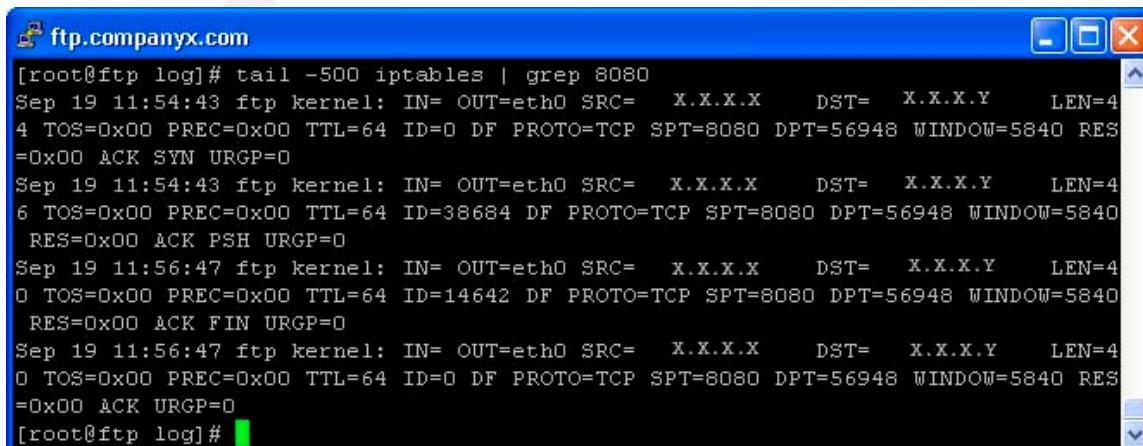


```
security-test.companyx.com
# telnet ftp.companyx.com 8080
Trying X.X.X.X ...
Connected to ftp.companyx.com.
Escape character is '^]'.
howdy
```

P	4	Verify log entries for the traffic generated in step three. Log entries reside in the /var/log/iptables file.
---	---	---

The IPTables software successfully logged the connections to port 8080 from the security-test system, which I generated in step three. Below, Figure 61 shows the outbound response traffic. The administrative team has configured the IPTables software to log all inbound, outbound, and forwarded traffic. All of the log messages are written to the /var/log/iptables file. The system passed this step.

Figure 61 – IPTables log entries



```
ftp.companyx.com
[root@ftp log]# tail -500 iptables | grep 8080
Sep 19 11:54:43 ftp kernel: IN= OUT=eth0 SRC= X.X.X.X DST= X.X.X.Y LEN=4
4 TOS=0x00 PREC=0x00 TTL=64 ID=0 DF PROTO=TCP SPT=8080 DPT=56948 WINDOW=5840 RES
=0x00 ACK SYN URGP=0
Sep 19 11:54:43 ftp kernel: IN= OUT=eth0 SRC= X.X.X.X DST= X.X.X.Y LEN=4
6 TOS=0x00 PREC=0x00 TTL=64 ID=38684 DF PROTO=TCP SPT=8080 DPT=56948 WINDOW=5840
RES=0x00 ACK PSF URGP=0
Sep 19 11:56:47 ftp kernel: IN= OUT=eth0 SRC= X.X.X.X DST= X.X.X.Y LEN=4
0 TOS=0x00 PREC=0x00 TTL=64 ID=14642 DF PROTO=TCP SPT=8080 DPT=56948 WINDOW=5840
RES=0x00 ACK FIN URGP=0
Sep 19 11:56:47 ftp kernel: IN= OUT=eth0 SRC= X.X.X.X DST= X.X.X.Y LEN=4
0 TOS=0x00 PREC=0x00 TTL=64 ID=0 DF PROTO=TCP SPT=8080 DPT=56948 WINDOW=5840 RES
=0x00 ACK URGP=0
[root@ftp log]#
```


Checklist results: The system failed to meet all of the control objectives for this audit item. During the creation of the checklists, I was under the impression that the IPTables software was used as a fully functional host based firewall. That, however, was not the case. The firewall was barely configured, with only enough functionality to log traffic and block access to X. I interviewed the original system designer and found that the intention was to test the basic functionality and interoperability between IPTables and the rest of the system's services. He plans to implement a tighter security policy in the next release of the Linux system template.

Residual Risk

A system administrator can never fully mitigate system security risks, especially when the system provides services over the Internet. System security measures, designed to limit exposures, can only diminish the risks. Residual risk is risk that remains once these measures have been implemented. At this point in the audit process, there exists a high level of residual risk. This residual risk is mainly associated with the failed control objectives presented in this assignment. In Assignment 4, I have provided a list of these risks and the costs associated with them.

All of the security shortcoming detected in the above audit steps can be addressed in some way. The decision to address these risks must be made by evaluating the benefits as opposed to the total cost to the organization. I believe that the control objectives outlined in Assignment 2 adequately address the business concerns of the system, as they relate to system security. However, the system did not successfully meet all of the control objects. Of particular note, the system was not patched to the highest revision possible for many software packages, which including sendmail, BIND, and the Linux kernel. This alone could result in the compromise of the system.

My recommendation to the administrative team is to conduct research on the seriousness of these vulnerabilities and shortcomings, and calculate the expected cost to fix them. Checklists that were considered "passed", have an acceptable level of residual risk. I designed the checklists, from the administrator's perspective, so that a passing grade would indicate the administrative team had accepted all residual risk.

Is the System Auditable?

The system is auditable. Since I designed this audit from an administrator's perspective, it was relatively easy to mold it fit the actual requirements of the system. With the knowledge that I possess of the system, I was able to evade areas of the system that would be useless to audit. Those areas include processes like change control, management oversight, as well as items not

directly under the control of the administrative team like system monitoring and response (which is handled by another organization), and manufacturer warrantee service. I was able to narrow the focus to the critical items that would most mitigate the security risks of the system. In the future, I could use the format and processes created within this assignment to formulate additional checklists for control objectives that were not directly within the scope of this assignment.

© SANS Institute 2003, Author retains full rights.

Assignment 4 – Risk Assessment

Summary

I found the system to be noncompliant with a number of the original control objectives. To be specific, the administrative team has failed to keep the system's software and kernel up to date, correctly configure user quotas and the IPTables firewall, and reduce the risk of OS fingerprinting and spoofed syslog entries with the security mechanisms of the ftp.companyx.com system. These are serious issues, which the administrative team must address. In this assignment, I will discuss the residual risks associated these security shortcomings.

The objective of the audit, to ensure the secure configuration of the ftp.companyx.com server, which provides DNS, FTP, and Syslog service, was the focus of this audit. The audit process was a success, in that it uncovered a number of areas of concern. The information gathered during this process will aid the administrative team in further securing the system, and designing more secure systems in the future.

System 2 Checklist step 3: Sendmail version

Background:

In this checklist, the software revision levels were compared to the latest versions of the software available. I utilized the RedHat up2date program to automate most of the process. The program indicated that several software packages were out of date, one of which was sendmail. I conducted research, and discovered that the version of sendmail running was susceptible to an exploit. The exploit was a buffer overflow that could lead to the execution of arbitrary code. The following two links provide details:

<https://rhn.redhat.com/errata/RHSA-2003-283.html>

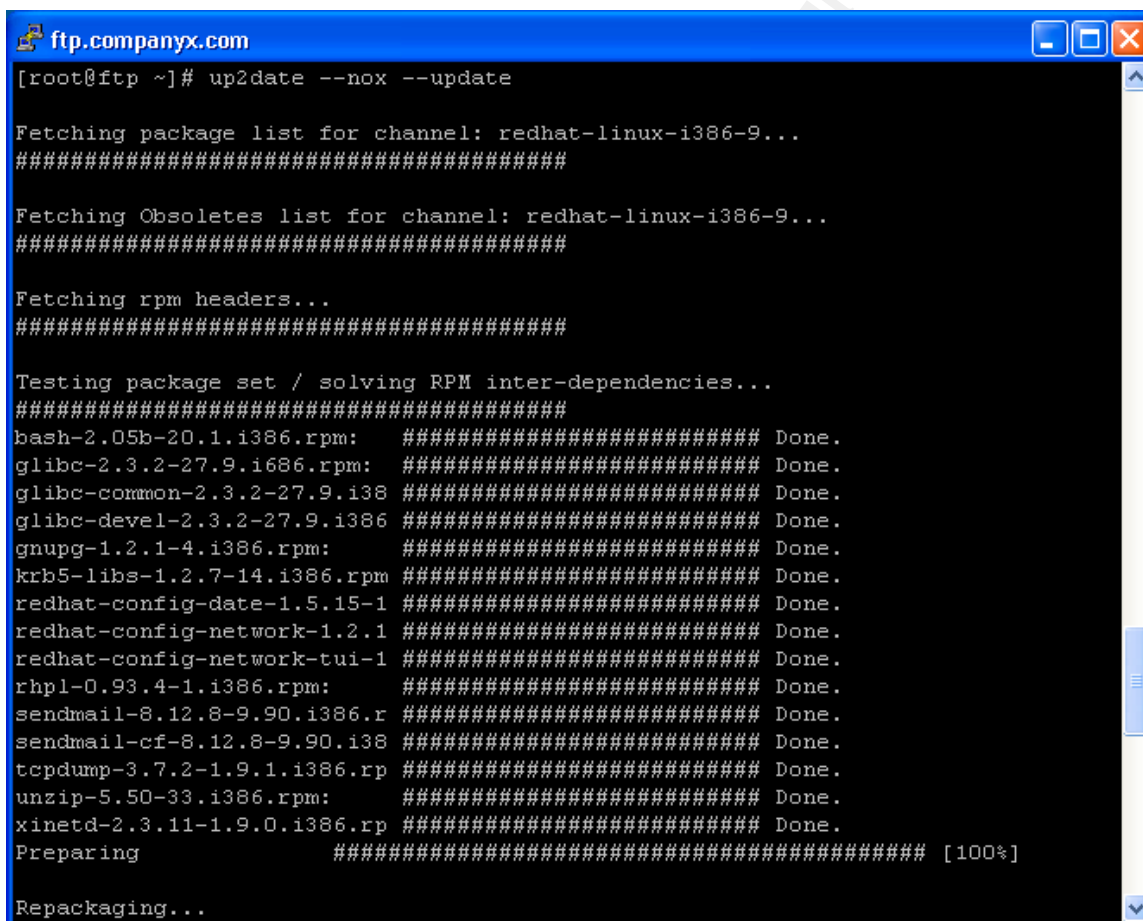
<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0694>

The risk of this threat being realized is minimal, because sendmail is configured to accept connections only from the localhost (as tested in the System 4 Checklist, step 5). However, a local user on the system could potentially exploit this vulnerability. The risk associated with this vulnerability is moderate, taking into account the threat and possibility of its realization. I decided that the best course of action would be an upgrade of all RPMs on the system.

System Changes:

To upgrade the RPMs, I utilized the up2date program. The administrative team had previously configured the program to create a “roll-back”, which could be used to recover from upgrade failures. By executing the “up2date --nox --update” command, the system was automatically updated. Figures 62 and 63 provide evidence of the system update. A system reboot was also required to restart the system daemons. By rerunning the up2date program, I was able to determine that all system RPMs were at the highest available version. The system is now compliant with this control objective, and the residual risk is acceptable.

Figure 62 – Up2date update

A screenshot of a terminal window titled 'ftp.companyx.com'. The terminal shows the execution of the 'up2date --nox --update' command. The output includes several status messages: 'Fetching package list for channel: redhat-linux-i386-9...', 'Fetching Obsoletes list for channel: redhat-linux-i386-9...', 'Fetching rpm headers...', 'Testing package set / solving RPM inter-dependencies...', and a list of RPMs being updated, each followed by 'Done.'. The list includes packages like bash, glibc, gnupg, krb5, redhat-config, rhpl, sendmail, tcpdump, unzip, and xinetd. The process ends with 'Preparing' at 100% and 'Repackaging...'.

```
ftp.companyx.com
[root@ftp ~]# up2date --nox --update

Fetching package list for channel: redhat-linux-i386-9...
#####

Fetching Obsoletes list for channel: redhat-linux-i386-9...
#####

Fetching rpm headers...
#####

Testing package set / solving RPM inter-dependencies...
#####
bash-2.05b-20.1.i386.rpm: ##### Done.
glibc-2.3.2-27.9.i686.rpm: ##### Done.
glibc-common-2.3.2-27.9.i386 ##### Done.
glibc-devel-2.3.2-27.9.i386 ##### Done.
gnupg-1.2.1-4.i386.rpm: ##### Done.
krb5-libs-1.2.7-14.i386.rpm ##### Done.
redhat-config-date-1.5.15-1 ##### Done.
redhat-config-network-1.2.1 ##### Done.
redhat-config-network-tui-1 ##### Done.
rhpl-0.93.4-1.i386.rpm: ##### Done.
sendmail-8.12.8-9.90.i386.r ##### Done.
sendmail-cf-8.12.8-9.90.i38 ##### Done.
tcpdump-3.7.2-1.9.1.i386.rp ##### Done.
unzip-5.50-33.i386.rpm: ##### Done.
xinetd-2.3.11-1.9.0.i386.rp ##### Done.
Preparing ##### [100%]
Repackaging...
```

Figure 63 – Up2date complete

```

ftp.companyx.com
Repackaging...
  bash ##### [100%]
  glibc ##### [100%]
  glibc-common ##### [100%]
  glibc-devel ##### [100%]
  gnupg ##### [100%]
  krb5-libs ##### [100%]
  redhat-config-date ##### [100%]
  redhat-config-network ##### [100%]
  redhat-config-network-t ##### [100%]
  rhpl ##### [100%]
  sendmail ##### [100%]
  sendmail-cf ##### [100%]
  tcpdump ##### [100%]
  unzip ##### [100%]
  xinetd ##### [100%]

Installing...
  1:glibc-common ##### [100%]
  2:glibc ##### [100%]
  3:bash ##### [100%]
  4:rhpl ##### [100%]
  5:redhat-config-network-t ##### [100%]
  6:krb5-libs ##### [100%]
  7:glibc-devel ##### [100%]
  8:gnupg ##### [100%]
  9:redhat-config-date ##### [100%]
  10:redhat-config-network ##### [100%]
  11:sendmail warning: /etc/mail/sendmail.cf created as /etc/mail/
sendmail.cf.rpmnew
warning: /etc/mail/submit.cf created as /etc/mail/submit.cf.rpmnew
##### [100%]
  12:sendmail-cf ##### [100%]
  13:tcpdump ##### [100%]
  14:unzip ##### [100%]
  15:xinetd ##### [100%]

The following Packages were marked to be skipped by your configuration:

Name                               Version    Rel Reason
-----
kernel                             2.4.20     20.9 Pkg name/pattern
kernel-source                       2.4.20     20.9 Pkg name/pattern
openssl                             0.9.7a     5     Pkg name/pattern

[root@ftp ~]#

```

System 5 Checklist steps 3, 4 and 5: Kernel version

Background:

The control objective of these checklist steps was to determine if the kernel version was up to date and free of known vulnerabilities. I utilized the RedHat up2date program, and the kernel.org website to determine if the currently used kernel was acceptable. I found that the kernel was susceptible to a plethora of

vulnerabilities. The following link discusses the vulnerabilities:

<https://rhn.redhat.com/errata/RHSA-2003-238.html>

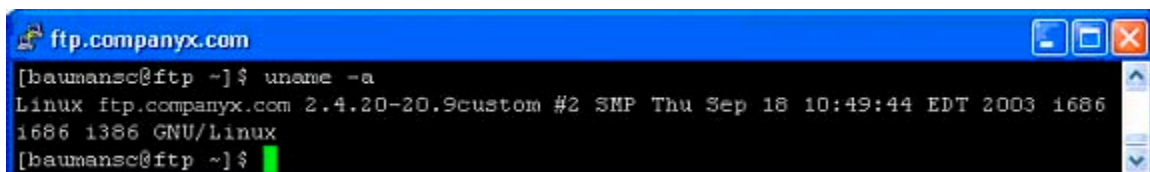
While the vulnerabilities seem to be rather obscure, the sheer volume of them mandates that the kernel must be updated. The risk to the system is high, since any one of these vulnerabilities could be used to exploit the system. If an attacker (system user or not) was able to exploit one of these vulnerabilities, they could create a DoS situation, or steal proprietary company information. I decided that a kernel upgrade was the best course of action. I also conducted research as to which kernel I should use. In step four of this checklist, I determined that kernel.org had a newer kernel available than RedHat. However, I could not find any vulnerability associated with the kernel version offered from RedHat, I decided that “bleeding edge” was not necessary in this situation. The cost of this fix is minimal, since I already have experience with kernel upgrades.

System Changes:

To upgrade to the kernel 2.4.20-20.9, I utilized the RPM source available from RedHat. I manually transferred the file by using the FTP protocol. I compared the checksum provided by RedHat with that of the kernel source RPM and found it to be identical. I installed the rpm source by executing “rpm -ivh kernel-source-2.4.20-20.9.i386.rpm”, and cleaned the kernel source by changing directory to /usr/src/kernel-2.4.20-20.9 and executing “make mrproper”. Next, I copied the previous kernel’s configuration file /usr/src/kernel-2.4.20-8/.conf to the new kernel source directory /usr/src/kernel-2.4.20-20.9. I used “make xconfig” to ensure that all of the previous kernel’s settings were viable with the newer kernel. This was a manual process. After saving the settings, I completed the compilation process by executing “make dep”, “make clean”, “make bzImage”, and finally “make install”. To make the system boot the new kernel, I modified the /etc/lilo.conf to utilize the new image and wrote out the new configuration by executing “lilo -v”. The entire custom kernel compilation process is discussed in the official Red Hat Customization guide at: <http://ftp.snt.utwente.nl/pub/linux/redhat/8.0/en/doc/RH-DOCS/rhl-cg-en-8.0/ch-custom-kernel.html>. To utilize the new system kernel, I restarted the system.

To retest the system, I simply ran the “uname -a” command. Figure 64 shows the results of this test. The system has now passed this control objective.

Figure 64 – Kernel update

A terminal window titled 'ftp.companyx.com' with standard window controls. The prompt is '[baumannsc@ftp ~]\$'. The command 'uname -a' has been entered and executed. The output is: 'Linux ftp.companyx.com 2.4.20-20.9custom #2 SMP Thu Sep 18 10:49:44 EDT 2003 i686 i686 GNU/Linux'. The prompt is now '[baumannsc@ftp ~]\$' with a green cursor.

```
[baumannsc@ftp ~]$ uname -a
Linux ftp.companyx.com 2.4.20-20.9custom #2 SMP Thu Sep 18 10:49:44 EDT 2003 i686
i686 i386 GNU/Linux
[baumannsc@ftp ~]$
```

System 9 Checklist step 4: OS Fingerprint

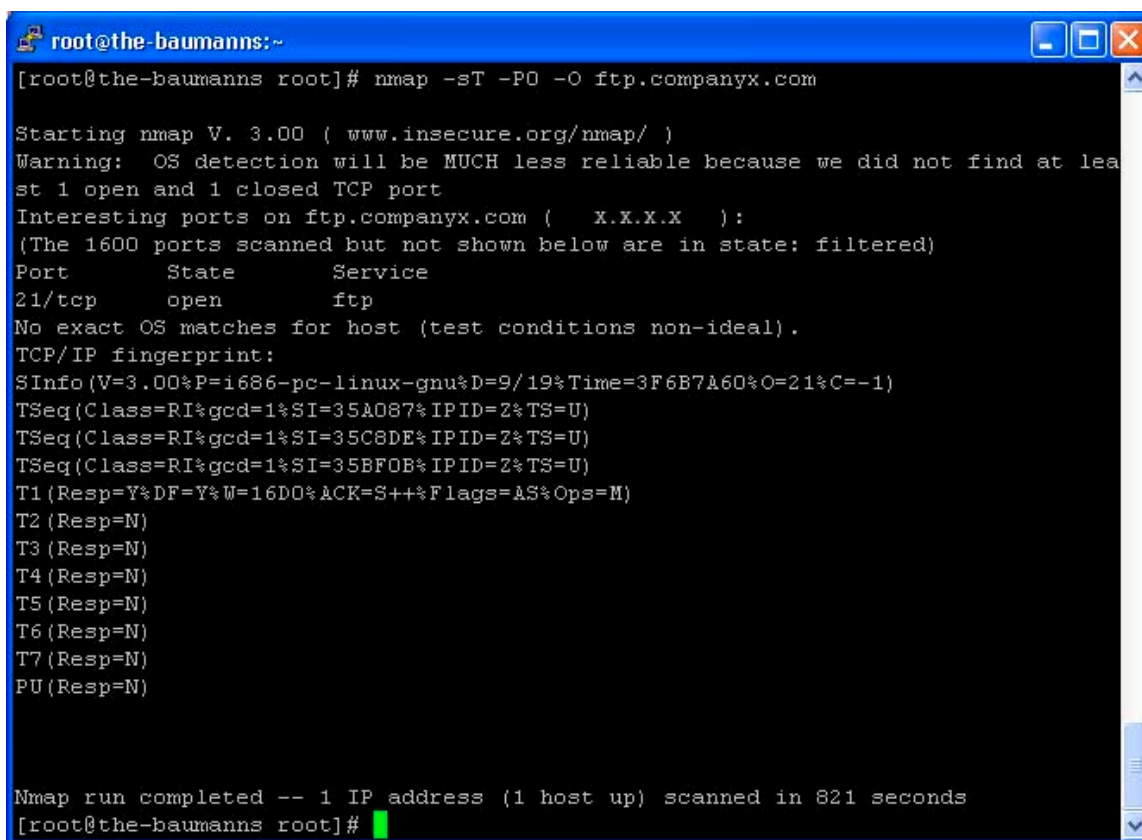
Background:

The control objective associated with checklist nine is the affirmation that hacker reconnaissance would provide no useful information. This would prevent a hacker from conducting specifically targeted attacks, instead of relying on “shots in the dark” tactics. While an nmap scan was not able to directly identify the systems OS, it did uncover enough information for a hacker to gain insight into what OS is being utilized. I will call this a partial fingerprint. It is difficult to gauge the risk level associated with this partial fingerprint, but it definitely gives an attacker an advantage. Since the system has already passed System Checklists one through eight, I will assign this a low-level of residual risk. The system’s software and kernel are already up to date, so this information will not put the system under more risk until such time that upgrades are once again required. The cost associated with fixing this issue is quite high. The administrative team has already implemented the iplog program (System 1 Checklist, step 6) to fool programs like nmap. Additional research would be needed to mitigate this risk, which I have estimated at sixteen to eighteen person-hours. This research would not guarantee a solution, just a firmer grasp of the actual problem.

System Justification:

The administrative team has chosen not to pursue a fix for this issue. When comparing the benefit of a fix to the amount of time involved in research and implementation, we found that the fix just was not worth the trouble. However, this item will be added to a “wish list” of enhancements that will be addressed in future releases of Company X’s Linux template. For now, the additional protection provided by iplog and the Internet firewalls is sufficient. Figure 65 shows the results of an nmap scan conducted from a system outside of the Company X network. The scan shows that only one port, the FTP port, is available to external systems (except for the ISPs DNS servers). Nmap was not able to provide the same level of detail from this system as compared to the security-test system. This result provides proof that the residual risk is minor.

Figure 65 – Nmap OS detection



```
root@the-baumanns:~  
[root@the-baumanns root]# nmap -sT -PO -O ftp.companyxx.com  
  
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )  
Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port  
Interesting ports on ftp.companyxx.com ( X.X.X.X ):  
(The 1600 ports scanned but not shown below are in state: filtered)  
Port      State      Service  
21/tcp    open       ftp  
No exact OS matches for host (test conditions non-ideal).  
TCP/IP fingerprint:  
SInfo (V=3.00%P=i686-pc-linux-gnu%D=9/19%Time=3F6B7A60%O=21%C=-1)  
TSeq(Class=RI%gcd=1%SI=35A087%IPID=Z%TS=U)  
TSeq(Class=RI%gcd=1%SI=35C8DE%IPID=Z%TS=U)  
TSeq(Class=RI%gcd=1%SI=35BFOB%IPID=Z%TS=U)  
T1 (Resp=Y%DF=Y%W=16D0%ACK=S++%Flags=AS%Ops=M)  
T2 (Resp=N)  
T3 (Resp=N)  
T4 (Resp=N)  
T5 (Resp=N)  
T6 (Resp=N)  
T7 (Resp=N)  
PU (Resp=N)  
  
Nmap run completed -- 1 IP address (1 host up) scanned in 821 seconds  
[root@the-baumanns root]#
```

System 10 Checklist step 2: Syslog Spoof

Background:

This system checklist was created to test the security of the syslog service on the ftp.companyxx.com system. In step two, I utilized the syslog-poison.c program to generate spoofed syslog packets. The ftp.companyxx.com system accepted and logged these alerts. Theoretically, an attacker could spoof syslog alerts to reduce the administrative team's ability to investigate network or system intrusions by generating confusing or excessive alerts (needle in a haystack), or to cause a DoS of the system by exhausting the /var partition.

Justification:

The actual problem is not with the system, it is with the syslog protocol. The syslog protocol does not include any type of authentication, so there is no real way of verifying the sender of a syslog alert (except perhaps by MAC address, which an attacker could also spoof). While this service could be used as a DoS, there are even easier ways to create the same type of DoS. For example, an attacker that does not have /etc/hosts.allow permissions to connect to the FTP server would generate one alert to the /var/log/messages file, and one

email alert to the root user's email box, from every connection attempt. This, in effect, is even more potent than the simple syslog spoof attack, since it creates more logs with one attempt. This is where defense-in-depth becomes important. The system itself cannot mitigate this risk, but Company X's router can. By adding anti-spoof ACL entries on the internet router, spoofed syslog packets cannot reach the ftp.companyx.com system. Fortunately, these entries already exist on the internet router, so this risk is potentially mitigated. A quick test revealed that spoofed packets generated from my home workstation did not reach the ftp.companyx.com system. This could be due to the ACLs, or by egress filtering on the part of my internet provider. Either way, the attack was not successful.

System 10 Checklist step 1: BIND version

Background:

The failed control objective of this checklist audit item was to determine if the BIND software running on the system was current and up to date. Through the research conducted in step one of this checklist, I found that the ISC had released version 9.2.2 which addresses many known vulnerabilities. The following is a link to the ISC website, where they have listed the vulnerabilities: <http://www.isc.org/products/BIND/bind-security.html>. Of particular interest are the two critical vulnerabilities: the "ntx bug", and the "tsig bug." Exploits are known to exist for these bugs. An attacker could gain access to the system, and potentially alter the DNS zone files or configuration settings. These high-risk (labeled by the ISC) vulnerabilities are worth the time expense required to address them.

Justification:

Continued research yielded that the version of BIND that RedHat distributes is not vulnerable to the bugs listed on the ISC website. The following is a link to the RedHat errata page where they discuss these vulnerabilities: <http://rhn.redhat.com/errata/RHSA-2002-133.html>. RedHat claims that the software shipped with version 9 of the OS is not vulnerable to any of the exploits, they have back-ported the non-vulnerable resolver library code to function with their version 9.2.1 of the BIND software. Because of this, the system can remain in its current state, and it is compliant with the control objectives.

System 14 Checklist steps 2, 4, 5 and 6: User Quotas

Background:

It was my understanding that the system was supposed to implement and support user quotas. User quotas were going to be used to limit the amount of drive space that any one user, or group, could consume. This would ensure that the service would not suffer a DoS due to lack of drive space. Because quotas are not enforced, a legitimate FTP user could consume all available drive space, which would cause other users not to be able to utilize the service. The risks associated with this are moderate, as this has not been a problem historically. If free drive space were totally exhausted, the administrative team could easily remove the files (even out of business hours).

Correctly configuring user quotas is not trivial. However, the administrative team does have experience in this area. As a result, the cost of implementation would be rather low, about six to eight person-hours to configure the service, and fourteen person-hours to determine and implement the quota sizes.

Justification:

During follow-up interviews with the administrative team, I discovered that there existed two business reasons for not implementing the quotas. First, many of the system users are involved in projects where they may require 80% of the system's available drive space at any one time. In addition, these users could require these resources during non-standard business hours. The SLA stipulates that the administrative team will not perform end user "emergency" changes outside of business hours, and the users must submit these changes to the change management committee. To minimize the residual risk, the system designer implemented two methods of dealing with drive space issues. First, he created a shell script that deletes any files that are older than seven days. This script is executed through the root user's crontab every night at midnight. Second, he implemented a script that monitors the drive space, and sends an alert email to the administrative team's alert email box when drive space approaches critical limits. Because of these mitigating items, the residual risk is small and acceptable to the administrative team.

System 17 Checklist steps 1 and 3: IPTables Implementation

Background:

The control objective of this audit checklist was to ensure the proper and effective configuration of the IPTables firewall as a host based firewall. The administrative team had not fully configured the firewall before the system went in to production status. With the IPTables firewall not completely configured, it does not offer the system any significant protection. The purpose of the firewall was to only allow access to the system's legitimate processes such as FTP, telnet, SSH, and DNS. Without the firewall's protection, system users can establish open sockets (as demonstrated in System 17 Checklist step 3), attackers can fully probe the system for vulnerabilities or faulty system

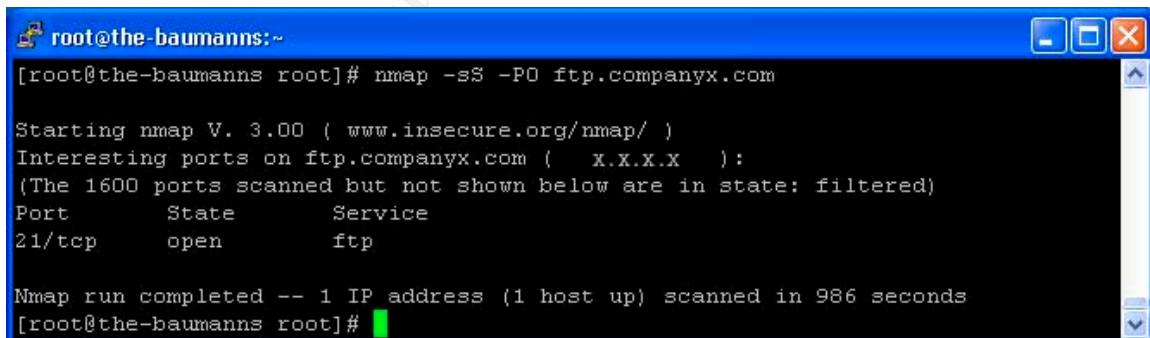
configurations, and potential back-door malware programs could “phone home.” The risk associated with these failed control objectives is moderate. If the administrative team correctly configures the system, keeps all software packages up to date, and the system configuration remains static in nature, then the risk has been essentially minimized. The cost for repair of the IPTables software is large, I estimate that it would take the administrative team thirty to forty person-hours to gain a basic understanding of how the IPTables software functions, another twenty person-hours to generate a proper firewall security policy, and at least twenty more person-hours to test the system. From the research that I have conducted, IPTables is nontrivial to understand.

Justification:

The mitigating factor in this situation is again defense-in-depth. Company X does not rely solely on the security of the system; it also relies on a highly layered set of security devices and processes. The ftp.companyx.com system is homed directly behind a pair of highly available stateful packet filtering firewalls. The security team has implemented a very specific rule set on these firewalls, allowing only the specifically required services to pass. The administrative team, however, has added the full implementation of IPTables to their “wish list” for future versions of the Linux template system.

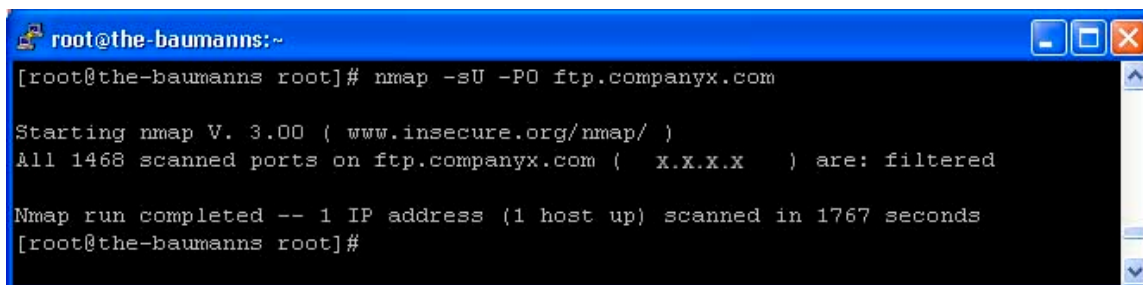
To retest these control items, I conducted a scan from an Internet hosts. I have provided the results of the scan in figures 66 and 67. An external host can only connect to the FTP service. A netcat listener, set on any other port number, was unreachable from the Internet host.

Figure 66 – Final TCP nmap scan



```
root@the-baumanns:~  
[root@the-baumanns root]# nmap -sS -PO ftp.companyx.com  
  
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )  
Interesting ports on ftp.companyx.com ( X.X.X.X ):  
(The 1600 ports scanned but not shown below are in state: filtered)  
Port      State      Service  
21/tcp    open       ftp  
  
Nmap run completed -- 1 IP address (1 host up) scanned in 986 seconds  
[root@the-baumanns root]#
```

Figure 67 – Final UDP nmap scan



```
root@the-baumanns:~  
[root@the-baumanns root]# nmap -sU -PO ftp.companyx.com  
  
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )  
All 1468 scanned ports on ftp.companyx.com ( X.X.X.X ) are: filtered  
  
Nmap run completed -- 1 IP address (1 host up) scanned in 1767 seconds  
[root@the-baumanns root]#
```

Conclusions

Due to the risk analysis conducted in Assignment 4, I now consider the system compliant with all of the control objectives defined in Assignment 2. In addition, the administrative team has accepted all residual risk. While the audit process is long and arduous, it can reveal many things that a system administrator does not know about his or her systems. While the research effort may mimic processes that my organization has already followed, the organizational methods required to complete this audit have opened new door of discovery for us.

References

- Andaesson, Oskar. "IPTables Tutorial." 2003. URL:<http://iptables-tutorial.frozentux.net/iptables-tutorial.html> (September 19, 2003)
- Brockmeier, Joe. "Using IPTables." April 2001. URL: <http://www.unixreview.com/documents/s=1236/urm0104l/0104l.htm> (September 19, 2003)
- CERT, "Problems with the FTP PORT Command." Carnegie Mellon University. 1999. URL: http://www.cert.org/tech_tips/ftp_port_attacks.html (September 19, 2003)
- Deraison, Renaud. "Nessus Demonstration." 2002. URL:<http://www.nessus.org/demo/index.html> (September 19, 2003)
- Forbes, Liam. "The First Ten Steps to Securing a UNIX Host." URL:<http://www.arisc.edu/~lforbes/cug/HHPaper.html> (September 19, 2003)
- Fyodor. "Nmap Network Security Scanner Man Page." 2003. URL:http://www.insecure.org/nmap/data/nmap_manpage.html (September 19, 2003)
- Green, John. "UNIX System Auditing and Forensics." SANS Institute, 2003.
- Hayes, Bill. "Conducting a Security Audit: An Introductory Overview." May 26, 2003. URL:<http://www.securityfocus.com/infocus/1697> (September 19, 2003)
- Hannett, Dan. "Other BIND Gems." April 4, 2000. URL:<http://www.freebsdjournal.org/bind-version.php> (September 19, 2003)
- Hobbit. "The FTP Bounce Attack." URL:<http://www.geocities.com/SiliconValley/1947/Ftpbounc.htm> (September 19, 2003)
- Hobbit. "Netcat 1.10 README." 2003. URL:http://www.atstake.com/research/tools/network_utilities/nc110.txt (September 19, 2003)
- Hoelzer, David, "7.1 Auditing Principles and Concepts." SANS Institute. 2003.
- Householder, Allen and King, Brian. "Securing an Internet Name Server." Cert Coordination Center, August 2003. URL: <http://www.cert.org/archive/pdf/dns.pdf> (September 19, 2003)
- Knowledgeleader.com. "Physical Security Audit Checklist." 2003. URL:<http://www.knowledgeleader.com/iafreewebsite.nsf/content/TechnologyAuditPhysicalSecurityAuditChecklist?OpenDocument> (September 19, 2003)
- Laude, Mary. "Auditing Red Hat Linux 7.0." July 23, 2001. URL: http://www.giac.org/practical/Mary_Laude_GSNA.zip (September 19, 2003)
- Mourani, Gerhard. "Securing and Optimizing Linux: RedHat Edition." OpenDocs, LLC. 2000. URL: <http://www.tldp.org/LDP/solrhe/Securing-Optimizing-Linux-RH-Edition-v1.3> (September 19, 2003)

- Paul, Brooke. "Building an In-Depth Defense." July 9, 2001. URL: <http://www.networkcomputing.com/1214/1214ws1.html> (September 19, 2003)
- Red Hat. "The Official Red Hat Linux Customization Guide." Red Hat, Inc. 2002. URL: <http://ftp.snt.utwente.nl/pub/linux/redhat/8.0/en/doc/RH-DOCS/rhl-cg-en-8.0/ch-custom-kernel.html> (September 19, 2003)
- Red Hat. "Quota Mini-HOWTO." Red Hat, Inc. August 1997. URL: <http://www.europe.redhat.com/documentation/mini-HOWTO/Quota-4.php3> (September 19, 2003)
- Red Hat. "RedHat Linux 8.0: The Official Red Hat Reference Guide." Red Hat, Inc. 2002 URL: <http://www.redhat.com/docs/manuals/linux/RHL-8.0-Manual/ref-guide/ch-tripwire.html> (September 19, 2003)
- Sax, Doug. "DNS Spoofing (Malicious Cache Poisoning)." 2000. URL: http://www.giac.org/practical/gsec/Doug_Sax_GSEC.pdf (September 19, 2003)
- Stewart, Joe. "DNS Cache Poisoning – The Next Generation." January 27, 2003. URL: <http://www.securityfocus.com/guest/17905> (September 19, 2003)
- Teoh, Cheng C. "Defense in Depth for DNS", SANS Institute, <http://www.sans.org/rr/paper.php?id=867>, 2003.
- VanMeter, Charlene. "Defense in Depth: A Primer", SANS Institute, February 19, 2001.
- Welte, Harald. "IPTables FAQ." August 16, 2002. URL: <http://www.netfilter.org/documentation/FAQ/netfilter-faq.html> (September 19, 2003)

Vulnerability Reference sites:

- CERT: http://www.cert.org/nav/index_red.html
- Red Hat Errata: <https://rhn.redhat.com/errata/rh9-errata.html>
- Security Focus: <http://www.securityfocus.com/bid>

Tools:

- Bastille Linux: <http://www.bastille-linux.org>
- Netcat: http://www.atstake.com/research/tools/network_utilities/
- Nmap: http://www.insecure.org/nmap/nmap_download.html
- Nessus: <http://www.nessus.org/download.html>
- Syslog_poison.c: <http://content.443.ch/pub/linfiles/Gnusoftware/spoofcode/syslog-poison.c>