



Interested in learning
more about security?

SANS Institute InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

Auditing a Corporate Log Server

Copyright SANS Institute
Author Retains Full Rights

AD

DEEPARMOR[®]

Auditing a Corporate Log Server

GIAC Gold Certification

GIAC Systems and Network Auditor (GSNA)

Roger Meyer
17 September 2006

© SANS Institute 2006, Author retains full rights.

Table of Contents

1 Abstract.....	3
2 Introduction.....	4
2.1 Definition of risk.....	4
2.2 Introduction to Computer Security Log Management.....	4
2.3 The Need for Log Management.....	5
2.4 The Challenges in Log Management.....	6
2.5 Audit Methodology.....	6
3 Identification.....	8
3.1 Scope: The Corporate Log Server.....	8
3.1.1 The log collection.....	8
3.1.2 The log processing.....	10
3.1.3 The log presentation (web interface).....	10
3.1.4 Server operation.....	11
3.1.5 Server configuration.....	11
3.2 The Log Infrastructure.....	11
4 Risk Analysis.....	13
4.1 Three identified risks.....	13
4.1.1 Risk 1: Unauthorized access of logs (loss of confidentiality).....	13
4.1.2 Risk 2: Unauthorized modification of logs.....	14
4.1.3 Risk 3: Loss of logs.....	14
4.1.4 Exposure.....	15
5 Testing.....	16
5.1 Risk 1: Unauthorized access of logs (loss of confidentiality).....	16
5.2 Risk 2: Unauthorized modification of logs.....	18
5.3 Risk 3: Loss of logs.....	22
6 Audit.....	24
6.1 Audit Planning / setting up an audit.....	24
6.2 Summary of findings and recommendations.....	40
7 Conclusion.....	41
8 References.....	42

Illustration Index

Illustration 1: Four-Stage Testing Methodology.....	7
Illustration 2: Overall Architectural Log Collection Diagram.....	9
Illustration 3: Firewall Accepts / Minute: 48 hours view.....	10
Illustration 4: Number of Messages: 8 days view.....	11
Illustration 5: Log Management Infrastructure Tiers.....	12
Illustration 6: Ethereal screen shot showing session initiation and transfer.....	25
Illustration 7: SSLDigger scanning for supported ciphers.....	28
Illustration 8: Nessus Scan Options.....	34
Illustration 9: Nessus Report.....	34
Illustration 10: NetBackup: Restore Files.....	39

1 Abstract

This paper was written to fulfill requirements for GIAC Gold for the GSNA¹ (GIAC Systems and Network Auditor) Certification.

This paper details an audit of a corporate log server. The goal of the audit is to measure if implemented security controls are adequate on the server and to validate the configuration, since prevention is always better than cure.

The ever increasing number of computer devices located in a corporation produce a vast amount of log data. This log data contains information related to specific events that have occurred on a system. Collection and storage of these logs is important for reasons like traceability, statistics and identifying security events. Increasingly compliance requirements, laws, and industry regulations dictate the storage, and analysis of security events.

The audited log server collects log data from multiple sources including firewalls, VPN, routers and other security related systems, stores them and permits analysis and monitoring.

The remainder of this paper is divided into 5 parts:

It starts with the **Introduction** which explains the background material like risk and log management. The **Identification** describes the audited device, in this case the log server and surrounding infrastructure. The **Risk Analysis** defines the terms risk, threat, vulnerability and impact. Three risks to the log server are isolated and analyzed in detail. The next part – **Testing** – compiles a list of tests to determine the vulnerabilities chosen in the preceding part. The **Audit** performs the tests developed in the previous chapter.

The methodology used to evaluate the risk was derived from the function that risk is a multiplication of the threat likelihood and the threat impact. The vulnerabilities with the biggest impacts were chosen and analyzed. The tests showed that the identified risks were addressed appropriately with some minor recommendations for improvements.

1 GIAC Systems and Network Auditor (GSNA): <http://www.giac.org/certifications/audit/gsna.php>

2 Introduction

This introductory section defines risk and risk management and gives an introduction to computer security log management.

2.1 Definition of risk

In the NIST Special Publication 800-30² 'Risk Management Guide for Information Technology Systems', risk is defined as follows:

Risk is a function of the likelihood of a given threat-source's exercising a particular potential vulnerability, and the resulting impact of that adverse event on the organization.

Let's explain each of these terms³:

- Risk:** Risk is the potential impact (positive or negative) to an asset. It's a combination of the likelihood that a particular vulnerability will be either intentionally or unintentionally exploited by a threat-source and the magnitude of the potential harm that could result.
Example: loss of confidentiality
- Threat:** A threat is an unwanted event that may result in harm to an asset.
Examples: natural threats (earthquake), human threats (attacker)
- Vulnerability:** A vulnerability is a weakness or flaw in a system that can be accidentally triggered or intentionally exploited.
Example: unpatched operating system
- Impact:** Impact is the magnitude of the potential loss or seriousness of the event.
Example: violation of an organization's mission, reputation or interest

The determination of risk for a particular threat/vulnerability can be derived by multiplying the threat likelihood (e.g., probability) and the threat impact:

$$\text{Derived Risk} = \text{Threat likelihood} * \text{Threat Impact}$$

If either the threat likelihood or the threat impact is zero, there is no risk. Example: If there is an unpatched server which is not connected to the network, the threat impact may be high, but the threat likelihood is zero, thus no risk.

2.2 Introduction to Computer Security Log Management

The NIST⁴ Special Publication 800-92 'Guide to Computer Security Log Management'⁵ guide defines a log as follows:

A log is a record of the events occurring within an organization's systems and networks. Logs are composed of log entries; each entry contains information related to a specific event that has occurred within a system or network.

2 NIST Special Publication 800-30: Risk Management Guide for Information Technology Systems
<http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf>

3 Ideas from the NIST SP800-30 Risk Management Guide

4 NIST - National Institute of Standards and Technology (<http://www.nist.gov/>)

5 NIST Special Publication 800-92: Guide to Computer Security Log Management, section ES-1
(<http://www.csrc.nist.gov/publications/drafts/DRAFT-SP800-92.pdf>)

Logs serve many functions in today's computer systems, such as troubleshooting problems, investigating malicious activity and recording the actions of users and systems. In the remainder of this document, computer security events are discussed even though there are many other events one can log.

In today's corporate networks a wide variety of networked servers and devices are generating an ever-increasing number of computer security logs. This has created the need for a process to manage those logs. Managing logs means defining a process for the generation, transmission, storage, analysis and deletion of the logs. The log server plays the central role in all those processes.

2.3 The Need for Log Management

Organizations are facing larger quantities, volumes and variety of computer security logs, and also need to address requirements to analyze and retain certain logs to comply with federal legislation and regulations, including FISMA⁶, HIPAA⁷, Basel II⁸ and the Sarbanes-Oxley Act (SOX)⁹. There is a report from the Enterprise Strategy Group¹⁰ called 'Security Information Lifecycle: Data Retention of Event Logs for Compliance'¹¹ which states the data retention requirements for different regulations. It ranges from 3 years for FISMA, 5 years for SOX, 6 years for HIPAA to 7 years for Basel II. The log retention requirements for the different regulations is somewhat controversial as to what kind of logs have to be archived. SOX requires in section 103¹² retention of at least 7 years: "... prepare, and maintain for a period of not less than 7 years, audit work papers, and other information related to any audit report, in sufficient detail to support the conclusions reached in such report." The Basel II guidelines call for the retention of log data over a period of three to seven years.

As a result, many organizations have a greater need for computer security log management. Log management assists in ensuring that computer security records are stored in sufficient detail for an appropriate period of time.

In addition to the laws and regulations, there are many other ways an organization can benefit of log management. Log reviews can help in the following areas:

- identifying security events
- identifying policy violations
- identifying fraudulent activity
- identifying operational problems
- performance auditing
- forensic analysis
- establishing baselines
- identifying operational trends

6 FISMA - Federal Information Security Management Act (<http://csrc.nist.gov/sec-cert/>)

7 HIPAA - Health Insurance Portability and Accountability Act (<http://www.cms.hhs.gov/HIPAAGenInfo/>)

8 Basel II Capital Accord: http://en.wikipedia.org/wiki/Basel_II

9 Sarbanes-Oxley Act: http://en.wikipedia.org/wiki/Sarbanes-Oxley_Act

10 Enterprise Strategy Group: <http://www.enterprisestrategygroup.com/>

11 Eric Ogren. Security Information Lifecycle: Data Retention of Event Logs for Compliance (http://www.enterprisestrategygroup.com/_documents/Report/Attachment1ID613.pdf)

12 SOX section 103: <http://fl1.findlaw.com/news.findlaw.com/hdocs/docs/gwbush/sarbanesoxley072302.pdf>

2.4 The Challenges in Log Management

The biggest challenge is the vast amount of ever increasing log data. It's a balance of keeping the right amount of data and still being able to analyze them.

Log Generation

Every organization has many different operating systems, security devices like firewalls, routers and other applications which each generate their own logs in their own format. These different log formats make the analysis and storage process challenging. Each device produces log data in it's (sometimes proprietary) format. For example Linux uses syslog (explained later) and Windows uses the event log service. The diversity of formats creates issues in the log transfer and the log analysis. Proprietary log formats have to be exported to a format where they can be easily sent to the log server. The log server then has to normalize the data for further analysis and correlation between the different devices. The normalization process brings logs into a consistent format.

Log Protection

Logs have to be protected from viewing and altering by unauthorized people, as they can contain confidential data. They not only have to be protected on the log server but also on the transmission from the generating device to the log server. Once an attacker has gained access to a system, he will try to hide his presence on the system by altering the system logs. This can be prevented by sending the logs in real time to the log server (also referred to as secondary logging).

Log Analysis

The log analysis, particularly the correlation of entries from multiple log sources to the same event is a major challenge. This last step of the log management process is one of the most important, but considered boring by most administrators. A serious analysis is a very time consuming task which requires knowledge in many different areas such as networking protocols (TCP/IP), operating systems (Linux, Windows) and applications (Checkpoint, Apache). It also takes the right tools for correlating entries from multiple log sources.

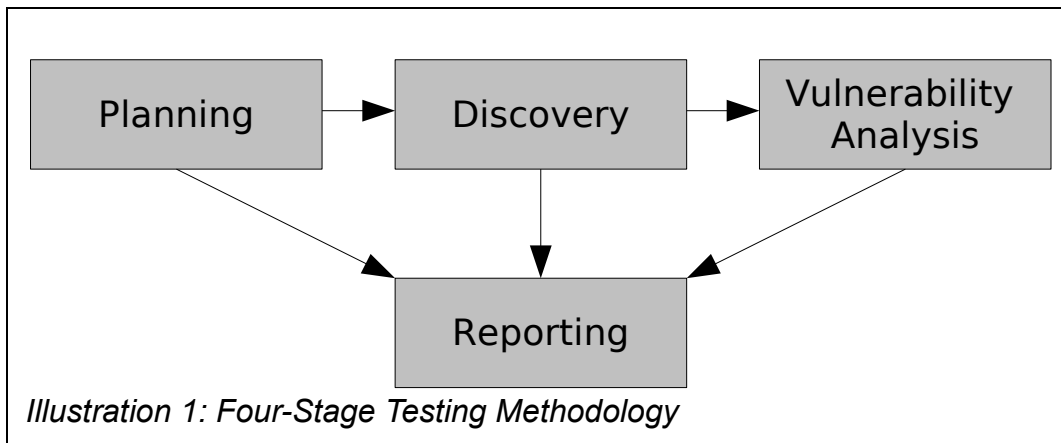
2.5 Audit Methodology

A methodology defines the methods, procedures, and techniques used to conduct an audit. The audit methodology used for this audit is a compliance-driven audit process to assess the risk of a given system and to compare it to best practices standards like the NIST Guideline on Network Security Testing¹³.

The audit methodology consists of four phases (see Illustration 1):

¹³ NIST, Special Publication 800-42: Guideline on Network Security Testing:

<http://csrc.nist.gov/publications/nistpubs/800-42/NIST-SP800-42.pdf>



In the **planning phase**, the target is identified and the business risks and regulatory challenges are evaluated in order to assess their significance to the business (Chapter 3 – Identification). Further, the management approval is finalized and the testing goals set. The vulnerabilities with the biggest impacts are chosen and analyzed (Chapter 4 – Risk Analysis).

The **discovery phase** starts the actual testing (Chapters 5 and 6 – Testing and Audit). Here, we try to get as much information about our target network(s)/host(s) as possible from DNS interrogation, searching web server(s) for information, network scanning, packet capture, and different kind of enumeration techniques like NetBIOS and Banner grabbing.

In the **vulnerability analysis phase** the results from the discovery phase are compared against vulnerability databases, and the design and operating effectiveness of the controls are tested (Chapter 6 – Audit).

The **reporting phase** occurs in all other three phases. It's where the findings are evaluated, the opinions of the audit are formed and the findings are reported (Chapter 7 – Conclusion).

© SANS Institute 2006. All rights reserved.

3 Identification

The subject of this audit is a corporate log server. This chapter covers the scope of the audit – the corporate log server – and the log infrastructure. It's the planning phase in the four-stage testing methodology.

3.1 Scope: The Corporate Log Server

This section defines the scope of the audit. A clear scope is very important for a successful audit. The scope of the audit is to check if the server is configured correctly and if the controls are appropriate.

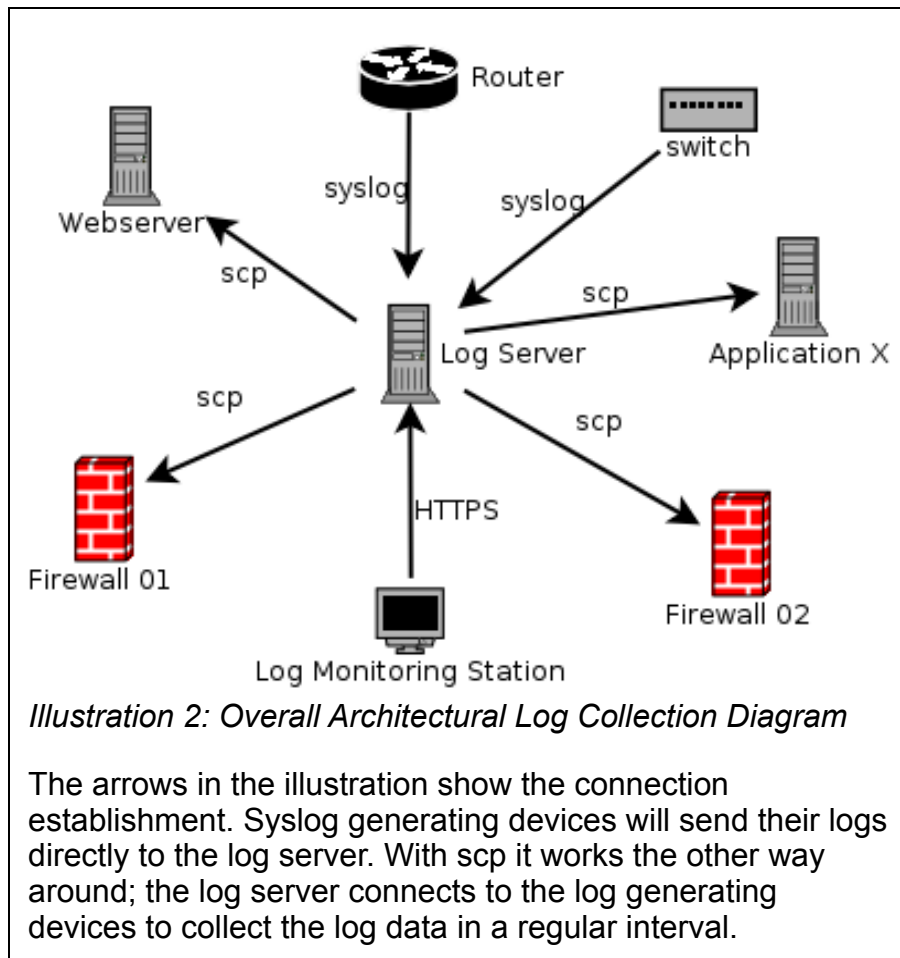
The corporate log server is a Red Hat Enterprise Linux machine. It acts as the collector and receiver of all log data.

3.1.1 The log collection

The log collection process is the part where the actual log data is transferred and collected on the log server. The communication between the log generating devices and the log server occurs over the organization's regular networks. There are many different ways this can be achieved. In our environment there are two different mechanisms in use:

- scp (secure file transfer over SSH)
- syslog¹⁴

¹⁴ RFC 3164 - The BSD syslog Protocol (<http://tools.ietf.org/html/3164>)



scp

scp (**Secure Copy**) is a program which is part of the OpenSSH¹⁵ package. SSH (**Secure Shell**) provides a way to make a secure connection between two systems. scp uses such a connection to copy files over a network. Additionally, it allows an automated login (passwordless) with the help of certificates. The transfer is initiated daily by the log server which makes a connection to the log generating device and copies the relevant log data. This means that the logs are not available in real-time. The big plus over the syslog transfer is that it is encrypted (provides confidentiality and integrity with SSL certificates).

Syslog

Syslog is described in the Request for Comments (RFC) 3164. It consists of log generators and one or more syslog server(s) which act as the collector for the log data. Syslog uses a standardized log format, which simplifies the storage of many different log sources.

The big disadvantage is the usage of the User Datagram Protocol (UDP)¹⁶ protocol for the transfer of the logs (by default on UDP port 514). UDP does not provide the reliability and ordering as TCP does. Syslog does not provide any basic security controls and does not preserve the confidentiality, integrity or availability of the logs. The advantage of syslog over scp is that it's operating in near real-time.

¹⁵ OpenSSH is a free version of the SSH connectivity tools (<http://www.openssh.org/>)

¹⁶ UDP - User Datagram Protocol (http://en.wikipedia.org/wiki/User_Datagram_Protocol)

Due to the provided confidentiality and integrity of scp, the company is using the SSH transfers wherever possible. Syslog transfers are only deployed if the generating device does not support a secure way of transport.

3.1.2 The log processing

Once the log files are on the log server, some of them have to be normalized and/or converted to a format where the logs can be further processed, e.g. to be able to create statistics. Normalizing log and security events enables comparisons of dissimilar systems. It's this process which brings logs into a consistent format. For example the log data from two different firewall manufacturers are normalized so they can be compared easily. Some log processing programs also require normalized log entries as they don't understand the log format of every firewall manufacturer. The original log data is always stored and is sometimes used for detailed analysis where the normalization process reduces and eliminates some information for simplification reasons.

The generated statistics are useful for the analysis as they offer an overview of many log entries in just a glance. Wading through thousands of log lines is a very cumbersome task and yields often only minimal results.

3.1.3 The log presentation (web interface)

The log server provides a web interface to present the formatted logs. It's running an Apache web server with PHP support over HTTPS. The web interface is protected by usernames and passwords implemented using .htaccess files¹⁷. .htaccess files provide a way to limit access to the web server on a per-directory basis by giving each user a username and password to authorize themselves. Each user has an account and is assigned to a corresponding group. Each group has a different menu and can only access the logs it is supposed to see.

There are many different scripts in use for the graphical presentation. Some of them have been written by the administrators, some are open source tools like RRDtool¹⁸ and awstats¹⁹, which generate graphical statistics. Here are two example graphics created for the graphical analysis using RRDtool and MRTG²⁰:

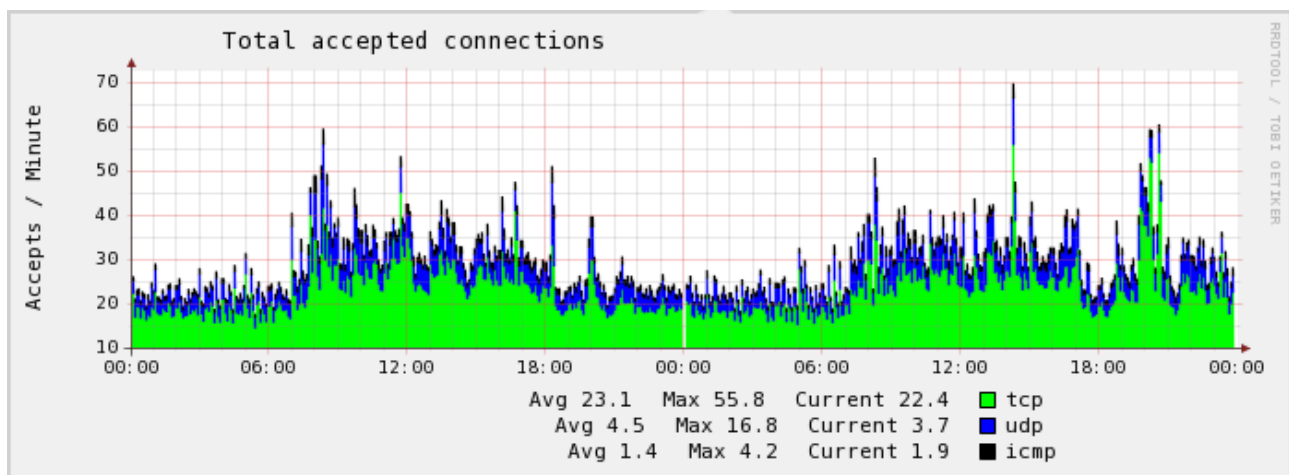


Illustration 3: Firewall Accepts / Minute: 48 hours view

¹⁷ Apache: .htaccess files (<http://httpd.apache.org/docs/2.0/howto/htaccess.html>)

¹⁸ RRDtool is a Round Robin Database tool (<http://oss.oetiker.ch/rrdtool/>)

¹⁹ AWStats is a free tool that generates web, streaming, ftp or mail server statistics, graphically. (<http://awstats.sourceforge.net/>)

²⁰ MRTG - The Multi Router Traffic Grapher (<http://oss.oetiker.ch/mrtg/>)

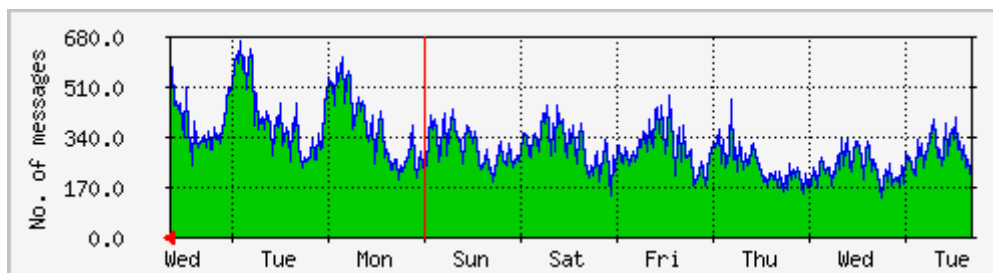


Illustration 4: Number of Messages: 8 days view

3.1.4 Server operation

There are many different kind of log data collected. The sensitivity of those logs differ, they can range from confidential to informational only. Amongst the more important logs are those from VPN (Virtual Private Network Access log), HTTP access logs, firewalls and routers.

For sensitivity reasons the log server is set up and operated by the security team. The security team does not operate other servers, firewalls, routers or any other devices. This has the goal of ensuring the separation of duties. The analysis and reporting of the log data is also conducted by the security team members.

3.1.5 Server configuration

The log server is running on a HP ProLiant DL380 machine with a dual CPU, 2GB of RAM, 1.4TB of hard disk, redundant power supplies and fans. The system was set up according to the company Linux guidelines from the central Linux install server. It's a Red Hat Enterprise Linux WS release 4 (Nahant Update 3) version. The following network daemons are enabled:

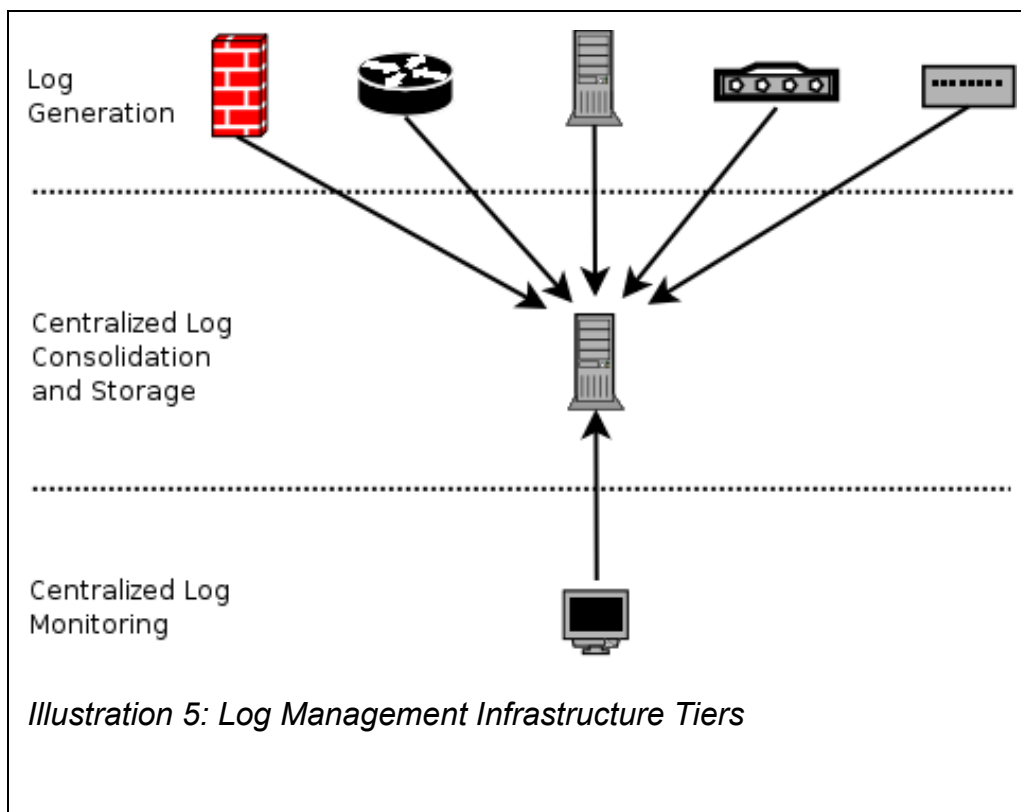
- SSH (TCP port 22)
- Apache (TCP port 443)
- Syslog (UDP port 514)
- NTP (TCP/UDP port 123)
- Veritas NetBackup (TCP 13722, 13724, 13782 and 13783)

All log processing scripts and programs are running under an unprivileged user account. Access to the server is strictly limited to the security team.

3.2 The Log Infrastructure

The log management is composed of three tiers:

- the log generation
- the centralized log consolidation and storage
- the log monitoring and alerting



Looking at Illustration 5 the log generation tiers contains the log generating devices like firewalls, routers and servers. The second layer contains the log server. It's role is to collect log data from multiple sources, normalize and store (archive) them to be accessed later through the log monitoring station (third layer – machine at the bottom). The monitoring station does not copy the logs from the log server, it accesses the (graphical) reports (and logs) through the web interface which is running on the log server.

© SANS Institute

4 Risk Analysis

This part deals with risk management – a part of planning phase. It is the process of identifying risk, assessing risk, and taking steps to reduce risk to an acceptable level. The objective of risk management is to give the management a good basis for decision making by supporting documentation resulting from the performance of risk management. It allows the IT managers to spend the right amount of protective measures to protect the IT systems and to achieve their organization's missions. The overall goal is to minimize the negative impact to an organization and to spend the right amount to protect the organization.

4.1 Three identified risks

Three risks with the most serious impacts were selected and will be described here with their impacts on the system and the primary vulnerabilities that could lead to these three impacts. There are many more risks to the system. The three risks should give a demonstration of the audit methodology. There are also many different vulnerabilities for each risk with the same or similar impact.

4.1.1 Risk 1: Unauthorized access of logs (loss of confidentiality)

Unauthorized access is possible through gaining access to the server and/or log data without the express permission of the owner. Transmitting the log data in the clear or using a weak encryption algorithm could also lead to a loss of confidentiality.

Unauthorized access of logs can happen through several ways:

- web interface (vulnerability in web application)
- hacked password (e.g. sniffed or guessed password)
- running unnecessary services (e.g. telnet)
- use of weak or no encryption during transmission (sniffing of clear text log transfer)

Impact

Unauthorized access to log data could lead to a loss of confidentiality. Log files can contain confidential data like user names, passwords and many other access and usage statistics. A potential attacker could also get useful information about network topology and vulnerabilities which could ease further attacks.

This can result in legal issues if laws and regulations are not followed. There are several laws like SOX which instruct how to handle log data. For more information please see chapter '2.3 The Need for Log Management'.

Vulnerabilities which could lead to this risk

- vulnerabilities in the web interface
- weak or no encryption during transfer of log data
- weak or no encryption for remote login
- weak or no encryption for accessing the web interface

Controls

A number of controls are implemented to mitigate the risks.

- Access to the system is limited to system administrators via password protected access
- The majority of transfers are encrypted file transfers to reduce the likelihood of

- sniffing clear text log transfers
- Remote administration access is only allowed with encrypted protocols like SSH
- Access to the web server (web interface) only over HTTPS with strong cyphers

4.1.2 Risk 2: Unauthorized modification of logs

The modification of log data consists of reading, modifying or deleting the content of log files. The unnoticed modification of data might be the most destructive as it can lead to wrong conclusions in the analysis and thereby to wrong accusations and decisions. A modification can be the deletion of one or more lines in a log file or the erasure of some or all log files. The purposeful alteration of specific log entries (e.g. changing IP addresses or user names) can be very difficult to detect.

Impact

The integrity and availability may be compromised by altering the logs. Unauthorized full access to log data can have several consequences. The alteration of specific log entries can lead to false conclusions like overlooking important events. Through the manipulation of evidence malicious activity may go unnoticed and the identity of a malicious party may be concealed.

Vulnerabilities which could lead to this risk

- careless handling of passwords by employees (e.g. choosing weak passwords, writing them down)
- vulnerability in OS
- vulnerability in a daemon (e.g. SSHD, Apache) or application (PHP web interface)
- misconfiguration by the system administrator
- vulnerability in report generating programs

Controls (to mitigate the risk)

- The use of strong passwords and good password processes
- The regular patching of the system
- The safe configuration as a default setting (e.g. no guest accounts)
- All unused services and daemons are disabled

4.1.3 Risk 3: Loss of logs

The complete or partial loss of logs can be intentional or unintentional. The intentional removal of some or all log data through a malicious person can have the goal to conceal a malicious activity and the identity of this malicious activity. The unintentional loss can occur through natural threats (earthquakes, floods) or human threats (malicious person) if there is no backup or if it was destroyed through the same means.

In his book 'Auditing Business Continuity', Rolf von Roessing states in the preface that:

Around 85% of Business Continuity Plans fail when first tested. [...] Over 50% of Business Continuity Plans are never tested.²¹

Impact

The most obvious impact is the missing analysis of the log data for all the listed benefits under 2.2 like identifying policy violations. The loss will most probably be a violation of laws and regulations which state what and how long the data has to be archived.

²¹ Rolf von Roessing. Auditing Business Continuity. ISBN #1-931332-15-0

Vulnerabilities which could lead to this risk

- hardware failure (e.g. hard disk failure, power outage) which leads to data corruption
- missing backup
- broken backup process or mechanism
- intentional or unintentional deletion of logs (improper handling)

Controls

- Maintaining a valid backup is a critical component to be able to recover in case of an emergency
- A backup process which is tested regularly
- The deployment of redundant hardware
- Physical access control

4.1.4 Exposure

This is a general note to the exposure of all of the above risks. Since the log server is in the internal network of the company – protected by (correctly configured) firewalls – it is not directly accessible from the Internet. This reduces the exposure significantly. There is still a significant threat from external sources of attack through rogue access points, VPNs, modems, etc. The '2005 FBI Computer Crime Survey'²² has some statistics regarding the relation of inside/outside attacks:

Overall, there were over twice as many unauthorized access incidents coming from outside the organization than there were from within, [...]

The national Computer Emergency Response Team for Australia (AusCERT) has some similar numbers. In their '2006 Australian Computer Crime and Security Survey'²³ they state:

Consistent with previous years' trends, most of these attacks were again sourced externally (83%) compared to internally (only 29%).

Regardless of the location of the log server, there is a direct and indirect exposure:

Direct exposure

A direct exposure is limited to internal attackers only. Direct attacks can involve attacks over the internal networks or physical attacks.

Indirect exposure

Indirect exposure can happen through specially crafted log entries. Most log entries are text elements which contain a certain known set of characters. If an attacker can control what an application will write into the log file, he may exploit analyzing programs, which may fail badly if they are not handling unexpected characters well. These log entries will pass all protective measures like firewalls easily, as they are assumed to be trusted content. This form of attack is called "Second Order Code Injection Attacks"²⁴.

²² 2005 FBI Computer Crime Survey, pp. 8

<http://www.digitalriver.com/v2.0-img/operations/naievigi/site/media/pdf/FBIccs2005.pdf>

²³ AusCERT. 2006 Australian Computer Crime and Security Survey. ISBN: 1-86499-849-0 (<http://www.auscert.org.au/images/ACCSS2006.pdf>)

²⁴ Next Generation Security Software Ltd. Second Order Code Injection Attacks <http://www.ngssoftware.com/papers/SecondOrderCodeInjection.pdf>

5 Testing

In the preceding chapter three major risks were described including their impacts and vulnerabilities. The corresponding tests for each risk will now be developed to identify the presence of vulnerabilities. For each test the criteria for determining whether the test passes or fails is explained.

The primary reason for testing the security of an operational system is to identify potential vulnerabilities so they can be fixed. As the number of reported vulnerabilities is growing daily, the testing for new vulnerabilities and misconfigurations is getting more and more important.

The audit will be conducted with the presence of the system administrator responsible for the operation and maintenance of the log server. Any commands that require root level access will be entered by the administrator and observed/noted by the auditor. Both the system administrator and the auditor have agreed to this testing methodology.

5.1 Risk 1: Unauthorized access of logs (loss of confidentiality)

Control 1
<p>Control objective: Verify that encryption is being used during transmission of the log data</p>
<p>Control activity: Make sure the server is configured to use encryption for the log transfer.</p>
<p>Test 1: Inspect the scp (SSH) transfer. The SSH daemon must answer on port TCP 22 and the transfer must be encrypted (not readable).</p> <ul style="list-style-type: none"> ● Connect to port 22 (default SSH port) with nc²⁵ to verify the daemon is running. <pre>\$ nc log_server 22</pre> <ul style="list-style-type: none"> ● Use Ethereal²⁶ to capture the log transfer. Ethereal is a popular network protocol analyzer. <p>As the root user start Ethereal. In the menu 'Capture' click on 'Interfaces...'. A dialog will appear with all interfaces. Choose the interface on which the SSH traffic will be on and click on the 'Prepare' button. Write the following in the field 'Capture Filter':</p> <pre>tcp port 22</pre> <p>This will limit the dumped traffic to the TCP port 22 which is the default port for SSH. Now start a log transfer and click 'Stop' once the transfer has occurred. Verify that after the initial key exchange you'll only see encryption requests and responses.</p> <p>The test passes if the SSH daemon answers on port 22 with a version string and there are no clear text log messages.</p>
<p>Test 2: Inspect the syslog transfer.</p> <p>Syslog traffic is sent in clear over UDP port 514. This makes it easy to intercept, spoof and fill the log server with bogus syslog entries. It is unnecessary to intercept syslog traffic since it will be unencrypted by default. Instead it will be verified if the syslog server</p>

²⁵ Netcat is a networking utility which reads and writes data across network connections (<http://netcat.sourceforge.net/>)

²⁶ Ethereal is a network protocol analyzer (<http://www.ethereal.com/>)

Control 1

can be filled with bogus entries.

We're going to use a small program written in the c programming language called `syslog_deluxe`²⁷. Once the source code is downloaded, we have to compile it:

```
$ gcc -o syslog_deluxe syslog_deluxe.c
```

This creates the binary executable `syslog_deluxe`. Execute it as root with the following parameters:

```
# syslog_deluxe src_hostname dst_hostname
```

This will send one syslog entry as specified in the source code (facility: LOG_DAEMON; priority: LOG_INFO). If the syslog server just received such an entry, bogus entries can create a Denial of Service (DoS). The test fails if the `syslog_deluxe` program successfully creates syslog entries.

Control 2

Control objective: Verify that the log server does not support weak encryption algorithms for the web access.

Control activity: Verify that the web server is configured to use strong encryption algorithms only.

Test 1: Inspect the Apache configuration for allowed cyphers. The test succeeds if only strong ciphers are allowed. Strong protocols are considered to be SSL3.x and TLS, SSL < 3 is not considered strong. Strong ciphers are starting at 128 bits.

Apache allows with the 'SSLProtocol' directive to set the allowed protocols. The configuration has to look something like this (disable all, enable TLSv1 and SSLv3):

```
SSLProtocol -all +TLSv1 +SSLv3
```

The ciphers are set with the 'SSLCipherSuite' directive. Apache has four parameter settings, HIGH (>168 bits), MEDIUM (128 bits) and LOW (<56 bits) cipher suites; and NULL (no encryption). There are a couple of other settings like aNULL and EXPORT which have to be disabled. Example configuration:

```
SSLCipherSuite HIGH:MEDIUM:!aNULL:+SHA1:+MD5:+HIGH:+MEDIUM
```

There are some other directives to force the usage of a SSL/TLS connection:

```
SSLEngine on
SSLOptions +StrictRequire
<Directory />
    SSLRequireSSL
</Directory>
```

Reference: A series of three articles dedicated to configuring Apache 2.0 with SSL/TLS support²⁸ on SecurityFocus by Artur Maj.

²⁷ `syslog_deluxe.c` - This program sends a spoofed syslog message (http://packetstormsecurity.org/Exploit_Code_Archive/syslog_deluxe.c)

²⁸ Apache 2 with SSL/TLS: Step-by-Step, Part 1: <http://www.securityfocus.com/print/infocus/1818>

Control 2

Test 2: Verify the used encryption algorithms (using `openssl`). No connection with weak cyphers (protocol < 3; ciphers < 128 bits) are allowed.

With the help of the OpenSSL²⁹ package we check manually what cyphers the web server supports. Let's try to connect with SSLv2 / SSLv3 and weak algorithms.

```
$ openssl s_client -connect log_server:443 -ssl2 -cipher 'LOW:NULL:aNULL:EXP'
$ openssl s_client -connect log_server:443 -ssl3 -cipher 'LOW:NULL:aNULL:EXP'
```

If there's an error message like the following, the test succeeded:

```
CONNECTED(00000003)
10243:error:1407F0E5:SSL routines:SSL2_WRITE:ssl handshake failure:s2_pkt.c:428:
```

A successful connection means the server accepted one of those low encryption algorithms and the test failed.

Test 3: Use SSLDigger³⁰ (Foundstone) to check for accepted ciphers. SSLDigger is a tool to assess the strength of SSL servers by testing the ciphers supported. This test is basically an extension of test 2, this time using an automated tool.

Once SSLDigger is running, type the web address of the server in the "Address" location bar and hit "Go". The results will be displayed on the screen and can be saved to an HTML file.

The same criteria apply as for test 1 & 2: protocol >= 3; ciphers >= 128 bits.

5.2 Risk 2: Unauthorized modification of logs

Control 3

Control objective: Verify that no unneeded services and daemons are running.

Control activity: Check the system configuration files and verify running services.

Test 1: Use the tool `chkconfig` to see what services are running. Only the services which are needed for the operation of the server are allowed.

The tool `chkconfig` queries runlevel information for system services. On a Linux system each service can be started or stopped in different runlevels. A runlevel is a mode of operation, they are ranging from 0 to 6, with 0 halting the system and 6 rebooting.

The following command lists all services which `chkconfig` knows about, and whether they are stopped or started in each runlevel:

```
# chkconfig --list
```

Test 2: Check the `/etc/xinet.d/` directory for running `xinetd`³¹ services. Only the `xinetd` services which are needed for the operation of the server are allowed.

Apache 2 with SSL/TLS: Step-by-Step, Part 2: <http://www.securityfocus.com/print/infocus/1820>

Apache 2 with SSL/TLS: Step-by-Step, Part 3: <http://www.securityfocus.com/print/infocus/1823>

²⁹ The OpenSSL Project is a toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols (<http://www.openssl.org/>)

³⁰ SSLDigger™ is a tool to assess the strength of SSL servers by testing the ciphers supported. (<http://www.foundstone.com/resources/proddesc/ssldigger.htm>)

³¹ `xinetd` - the extended Internet services daemon (<http://www.xinetd.org/>)

Control 3

In the directory `/etc/xinet.d/` are the configuration scripts for each service. Each service has a directive called 'disable'. It has to be set to 'yes' for those services which are not necessary (`disable = yes`).

Test 3: Use the `ps` and `netstat` tools to verify that no unneeded processes are running. It passes the test if only the processes which are needed for the operation of the server are running.

The program `ps` reports a snapshot of the current processes. It allows us to verify all running processes. The following command lists all running processes:

```
$ ps aux
```

Options:

- a – the -a option lists the processes of all users
- u – the -u option tells `ps` to provide detailed information about each process
- x – the -x option adds the processes that have no controlling terminal, such as daemons

`Netstat` can print a wealth of information, we're interested in the network connections. The following listing shows all open TCP and UDP ports including their process IDs:

```
$ netstat -lnp --tcp --udp
```

Options:

- l – show only listening sockets
- n – numeric only; don't use name lookups (DNS)
- p – show the PID and name of the program to which each socket belongs
- tcp – show TCP connections
- udp – show UDP connections

Test 4: Use `nmap`³² to verify that no unneeded ports are open. `Nmap` is a very popular network port scanner which allows us to see what ports are open. In the previous test we verified the open ports from the inside, now we're verifying the open ports from the outside. The test passes if only allowed services are running, i.e. only allowed ports are open. Those are SSH (TCP 22), NTP (TCP/UDP 123), syslog (UDP 514), HTTPS (TCP 443) and the NetBackup ports (TCP 13722, 13724, 13782 and 13783).

The following scans all TCP ports:

```
# nmap -sS -O -v -P0 -sV -p1-65535 log_server
```

Options:

- sS – TCP SYN scan
- O – enable OS detection
- v – increase verbosity level
- P0 – treat all hosts as online – skip host discovery (no ping)

³² `Nmap` ("Network Mapper") is a free open source utility for network exploration or security auditing (<http://www.insecure.org/nmap/>)

Control 3

-sV – probe open ports to determine service/version info
-p1-65535 – specifies the port range, in this case all TCP ports

After scanning the TCP ports, we're scanning all UDP ports:

```
# nmap -sU -P0 -p1-65535 log_server
```

Options:

-sU – UDP scan

Control 4

Control objective: Identify common vulnerabilities in the OS.

Control activity: Use a vulnerability scanner to check for missing patches and misconfigurations.

Test 1: Use Nessus³³ to scan the system for remote vulnerabilities. The test passes if no security holes are found.

Nessus is a very popular remote vulnerability scanner with an easy to use graphical interface. The installation process is not described here, there is a good introduction on SecurityFocus 'Introduction to Nessus'³⁴. Once the Nessus client is running and the user has logged in, several configuration options have to be set. The following options have to be adjusted:

- In the 'Plugins' tab, select the '**Enable dependencies at runtime**' checkbox to enable the plugins that are dependent on the set of plugins which were selected.
- In the 'Plugins' tab, click the '**Enable all**' button to select all plugins
- In the 'Scan Options' tab, enter the following in the 'Port range' text box: **0-65535**
- In the 'Scan Options' tab, uncheck the '**Safe checks**' checkbox
- In the 'Scan Options' tab, under 'Port scanner', verify that the '**Nmap (NSSL wrapper)**' is checked, so that Nessus will use Nmap for the port scan.
- In the 'Target' tab, enter the following in the 'Target(s)' text box: **log_server**
- In the 'Prefs.' tab, enable the 'UDP Scan' check box
- In the 'Prefs.' tab, enable the 'SSL' check box

Once those settings are set, the scan can be started by clicking the 'Start the scan' button at the bottom. When the scan is finished, there will be an option to save the results. It is recommended to save it in the native NBE format for re-importation by any Nessus client. There are several other output methods like 'HTML with Pies and Graphs' which is useful for a presentation.

Control 5

Control objective: Verify that the OS is up to date with security patches.

Control activity: Verify OS patch level.

³³ Nessus is a remote security scanner (<http://www.nessus.org/>)

³⁴ Introduction to Nessus: <http://www.securityfocus.com/print/infocus/1741>

Nessus, Part 2: Scanning: <http://www.securityfocus.com/print/infocus/1753>

Nessus, Part 3: Analysing Reports: <http://www.securityfocus.com/print/infocus/1759>

Control 5

Test 1: Use the Red Hat tool `up2date` to check the patch level. All current patches must have been applied.

The Red Hat Update Agent (`up2date`) is a program for managing and updating RPM packages on Red Hat Linux systems. We're assuming the Red Hat system has been set up correctly to use the Red Hat Network (RHN)³⁵ for the update process. To check if there are any new packages available, type:

```
$ up2date --dry-run
```

This will give an output of all available downloads with dependencies. The test succeeds if there are no new packages available.

Control 6:

Control objective: Verify user accounts and their privileges

Control activity: Inspect the user accounts, their rights and their password strength.

Test 1: Check `/etc/passwd` for `UID == 0`. The only account allowed to have UID 0 is root.

Use the `awk`³⁶ command to list accounts with UID 0:

```
$ awk -F: '($3==0){print $1}' /etc/passwd
```

Test 2: Check `/etc/shadow` for password less accounts. No enabled user account is allowed to have a null password.

Use the `awk` command to list any accounts that have empty password fields:

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

Test 3: Examine the `/etc/passwd` file for any unnecessary user accounts. No unneeded user accounts are allowed on the system.

```
$ cat /etc/passwd
```

Test 4: Use the John the Ripper³⁷ password cracker to verify the password strength. The purpose is to detect weak system passwords. However, there is no hard limit what a weak password is. The following characteristics are commonly seen as a weak password:

- any word in a dictionary
- passwords shorter or equal than 6 characters
- names, birthdays or any other characteristics which could lead to this person

This test passes if John doesn't find the password within 1 hour with the default options, which are "single crack" mode first, then using a word list with rules, and finally "incremental" mode. Single crack mode will use the login names, "GECOS" / "Full Name" fields, and users' home directory names as candidate passwords, also with a large set of mangling rules applied. In wordlist mode you supply john a word list and you can optionally enable word mangling rules to as some simple modifications like numbers at the end. Incremental mode is the most powerful cracking mode. It tries all possible passwords (brute force).

³⁵ Red Hat Network is a systems management platform (<http://www.redhat.com/rhn/>)

³⁶ AWK is a general purpose computer language that is designed for processing text-based data (<http://en.wikipedia.org/wiki/awk>)

³⁷ John the Ripper is a password cracker (<http://www.openwall.com/john/>)

Control 6:

First, we need to get a copy of the password file. John provides the 'unshadow' utility to obtain the traditional Unix password file, as root type:

```
# unshadow /etc/passwd /etc/shadow > mypasswd
```

To let John use its default order of cracking modes, type:

```
$ john mypasswd
```

John will display the cracked passwords on the screen.

5.3 Risk 3: Loss of logs

Control 7

Control objective: Verify that the backup procedure is in place and is working correctly.

Control activity: Inspect the backup mechanism.

Test 1: Perform a restore of several test files to ensure that backup can be restored. All tested files must be able to restore completely.

First, some arbitrary files have to be deleted. Then we're going to use the NetBackup graphical user interface (NetBackup Administration Console), a Java-based, graphical user interface to restore the file (`jnbSA`) command. There is also a character-based, menu interface that is started by running the `bpadm` command.

Start the NetBackup Administration Console:

```
# /usr/opensv/netbackup/bin/jnbSA
```

Log in with your user name and password. In the 'Restore files' tab, select the directory we've just deleted in the file tree. Now click on the "Restore..." button. In the "Restore Files" dialog, verify that "Restore everything to its original location" is checked, then click "Start Restore". When the restore is finished, the status will go to "Successful". Now try to verify that the restore completed successfully.

Control 8

Control objective: Prevent hardware outages.

Control activity: Determine if the environment is appropriate.

Test 1: Verify that both power supplies are working. Both power supplies must be working.

Pull one power plug and check if the server is continuing to run with the second power supply. Then pull the plug of the second power supply. The test succeeds if the server continues running with only one power supply plugged in.

Test 2: Verify that the UPS (uninterruptible power supply) is working.

An uninterruptible power supply (UPS) is a device or system that maintains a continuous supply of electric power to an equipment that must not be shut down unexpectedly. We

Control 8

are verifying that the UPS system is inspected and tested in a regular interval. To succeed this test the UPS system has to be inspected once a year including a test to see if it is working as expected.

Control 9

Control objective: Prevent physical access for unauthorized people

Control activity: Determine if physical security is adequate

Test 1: This test succeeds if the physical protection mechanisms in use is adequate for the server room. The following has to inspected:

- The doors to the computer room must be locked.
- It must be inspected who has access to the rooms.
- There must be a documentation and formal process who is allowed to access those rooms.
- The entrance has to have an access log which logs the date, time, room number and the name of the employee who accessed the room.

© SANS Institute 2006, Author retains full rights.

6 Audit

In this part the actual audit is conducted – it's the discovery and vulnerability analysis part of the testing methodology. The elaborated tests in the preceding chapter will be executed here. Each test will be explained in detail, the findings will be discussed and – depending on the result – a recommendation will be given. But first let's elaborate how to prepare for carrying out an audit.

6.1 Audit Planning / setting up an audit

Before the real audit work can be started, there are some important preparation steps to be taken which, when left out, can have severe consequences (e.g. accusation of unauthorized scanning). Probably the most important step in the audit planning part is to get written permission from the appropriate staff member(s) (or authorities), usually the CIO or any other management level employee with the right to authorize the audit.

Before every audit, the scope is defined. Without a clear scope, the auditor and/or the audit requester won't be satisfied with the result. The scope of this audit has been detailed in chapter '3.1 Scope'.

Further, you have to prepare yourself for the audit with the right computer equipment and tools. Be familiar with the used scanners and test them in a lab before using them on production systems. You also have to do research on the internal policy documents and industry best practices.

As a last step, set up a time for the audit and inform the necessary people like system and network administrators and the management when it's going to happen. Be ready to respond to questions and stay on site during the audit – don't start the scans and then walk away, as systems might crash and administrators might need help and answers.

Control 1 - Verify that encryption is being used during transmission of the log data

Test 1 (Control 1)	Result: pass
<p>Testing procedure</p> <p>This test consists of two parts, the verification of the SSH daemon and the TCP dump verification. First, we connect to port 22 (default SSH port) with nc to verify that the daemon is running:</p> <pre>\$ nc log_server 22 SSH-2.0-OpenSSH_3.9p1</pre> <p>This shows that the SSH daemon is responding on port 22. It is returning it's version string (SSH-2.0-OpenSSH_3.9p1). Unfortunately this string does not let us verify that it's running the latest version including the latest patches as linux distributions are patching their software without giving them a new version identification.</p> <p>The second test involves using Ethereal to dump the log transfer traffic over SSH. The following screen shot shows the dump in Ethereal:</p>	

Protocol	Info
TCP	56678 > ssh [SYN] Seq=0 Len=0 MSS=1460 TSV=10381454 TSER=0 WS=2
TCP	ssh > 56678 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 TSV=2362312219 TSER=10381454 WS=2
TCP	56678 > ssh [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSV=10381454 TSER=2362312219
SSH	Server Protocol: SSH-2.0-OpenSSH_3.9p1
TCP	56678 > ssh [ACK] Seq=1 Ack=23 Win=5840 Len=0 TSV=10381460 TSER=2362312226
SSH	Client Protocol: SSH-2.0-OpenSSH_4.2p1 Debian-8
TCP	ssh > 56678 [ACK] Seq=23 Ack=32 Win=5792 Len=0 TSV=2362312241 TSER=10381476
SSHv2	Client: Key Exchange Init
TCP	ssh > 56678 [ACK] Seq=23 Ack=744 Win=7216 Len=0 TSV=2362312242 TSER=10381477
SSHv2	Server: Key Exchange Init
SSHv2	Client: Diffie-Hellman GEX Request
SSHv2	Server: Diffie-Hellman Key Exchange Reply
SSHv2	Client: Diffie-Hellman GEX Init
SSHv2	Server: Diffie-Hellman GEX Reply
SSHv2	Client: New Keys
TCP	ssh > 56678 [ACK] Seq=1279 Ack=928 Win=8640 Len=0 TSV=2362312346 TSER=10381541
SSHv2	Encrypted request packet len=48
TCP	ssh > 56678 [ACK] Seq=1279 Ack=976 Win=8640 Len=0 TSV=2362312347 TSER=10381581
SSHv2	Encrypted response packet len=48
SSHv2	Encrypted request packet len=64
TCP	ssh > 56678 [ACK] Seq=1327 Ack=1040 Win=8640 Len=0 TSV=2362312388 TSER=10381582
SSHv2	Encrypted response packet len=1264
SSHv2	Encrypted request packet len=240
TCP	ssh > 56678 [ACK] Seq=2591 Ack=1280 Win=8640 Len=0 TSV=2362312398 TSER=10381632
SSHv2	Encrypted response packet len=192
SSHv2	Encrypted request packet len=384
TCP	ssh > 56678 [ACK] Seq=2783 Ack=1664 Win=10064 Len=0 TSV=2362312455 TSER=10381690
SSHv2	Encrypted response packet len=32
SSHv2	Encrypted request packet len=64
SSHv2	Encrypted response packet len=48
SSHv2	Encrypted request packet len=448
SSHv2	Encrypted response packet len=48

Illustration 6: Ethereal screen shot showing session initiation and transfer

The screen shot shows us the three-way TCP connection handshake and the initial key exchange. Eventually, all traffic is sent encrypted.

Findings

SSH daemon is running and only encrypted packets are seen in traffic dump.

Recommendation

None

Test 2 (Control 1)	Result: fail
<p>Testing procedure</p> <p>This test checks if bogus syslog entries can be inserted on the log server.</p> <p>After downloading the source code of <code>syslog_deluxe</code>, we have to compile it:</p> <pre>\$ gcc -o syslog_deluxe syslog_deluxe.c</pre> <p>This created the binary executable <code>syslog_deluxe</code>. Let's now execute it as root with the following parameters:</p> <pre># syslog_deluxe 192.168.1.99 log_server</pre> <p>This sends one syslog entry as specified in the source code (facility: <code>LOG_DAEMON</code>; priority: <code>LOG_INFO</code>) with the source IP address of <code>192.168.1.99</code> to the log server. Let's check if the log server just got a new syslog message. On the log server, execute the</p>	

Test 2 (Control 1)	Result: fail
<p>following command to print the last line of the default syslog log file:</p> <pre>\$ tail -1 /var/log/messages Jun 7 13:37:48 192.168.1.99 telnetd[4489]: connection from ...</pre> <p>We just got a new syslog entry from the bogus IP address 192.168.1.99.</p>	
<p>Findings</p> <p>The syslog daemon running on the log server accepts any kind of syslog messages. This means, any kind of entries can be inserted and may create a Denial of Service (DoS) attack through filling the disk space or just making the log analysis a very difficult task.</p>	
<p>Recommendation</p> <p>Protecting a syslog server using UDP for the transmission is very hard due to the nature of the protocol which does not provide reliability. One may restrict the source IPs of the syslog packets on the server. This will limit most fake entries but will still allow to enter bogus entries if an attacker fakes the source IP to one of the allowed IP addresses.</p> <p>The only secure way to transport syslog is to use a reliable syslog server like syslog-ng³⁸ which uses TCP. Additionally, traffic should be encrypted to provide integrity and confidentiality. This can be achieved for example with stunnel³⁹, which allows you to encrypt TCP connections inside SSL.</p>	

Control 2 - Verify that the log server does not support weak encryption algorithms for the web access.

Test 1 (Control 2)	Result: fail
<p>Testing procedure</p> <p>This test is inspecting the Apache SSL configuration. We're going to see what cyphers are configured in the Apache configuration files.</p> <p>The main configuration file is <code>/etc/httpd/conf/httpd.conf</code>. The SSL configurations are set in the following file, though: <code>/etc/httpd/conf.d/ssl.conf</code>. Here are the relevant entries:</p> <pre>SSL Engine on SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+EXP</pre> <p>This line allows the all cipher suites except ADH and EXPORT56. There are no other SSL settings in the <code>ssl.conf</code> file.</p>	
<p>Findings</p> <p>The configuration showed that all cipher suites except ADH and EXPORT56 are allowed. This is not a very strong setting. There are also some configurations missing like the allowed SSL protocols.</p>	
<p>Recommendation</p> <p>It is recommended that the configuration is adjusted to include the following settings:</p> <p>Only allow TLSv1 and SSLv3 protocols:</p> <pre>SSLProtocol -all +TLSv1 +SSLv3</pre>	

³⁸ syslog-ng - flexible and scalable audit trail processing tool: http://www.balabit.com/products/syslog_ng/

³⁹ Stunnel - Universal SSL Wrapper: <http://www.stunnel.org/>

Test 1 (Control 2)	Result: fail
<p>Use only HIGH (>168 bits) and MEDIUM (128 bits) cipher suites; exclude aNULL (no encryption):</p> <pre>SSLCipherSuite HIGH:MEDIUM:!aNULL:+SHA1:+MD5:+HIGH:+MEDIUM</pre> <p>Force the usage of a SSL/TLS connection:</p> <pre>SSL Engine on SSL Options +StrictRequire <Directory /> SSLRequireSSL </Directory></pre>	

Test 2 (Control 2)	Result: fail
<p>Testing procedure</p> <p>Now we're testing the SSL connection manually with the openssl tool from the OpenSSL package. Our goal is to find out if weak cyphers are supported.</p> <p>The following command tries to open a connection with the SSLv2 protocol and either a LOW, NULL, aNULL or EXP cipher suite:</p> <pre>\$ openssl s_client -connect log_server:443 -ssl2 -cipher 'LOW:NULL:aNULL:EXP' CONNECTED(00000003) ...</pre> <p>The connection works – with a weak cipher suite. Now let's try the same ciphers with SSLv3:</p> <pre>\$ openssl s_client -connect log_server:443 -ssl3 -cipher 'LOW:NULL:aNULL:EXP' CONNECTED(00000003) ...</pre> <p>Again, the connection works. To verify that not any ciphers are accepted, we'll try to connect with the SSLv2 protocol and NULL or aNULL ciphers:</p> <pre>\$ openssl s_client -connect log_server:443 -ssl2 -cipher 'NULL:aNULL' error setting cipher list 16781:error:1410D0B9:SSL routines:SSL_CTX_set_cipher_list:no cipher match:ssl_lib.c:1176:</pre> <p>This time the connection doesn't work anymore. The server doesn't support those ciphers.</p>	
<p>Findings</p> <p>The log server supports weak cipher suites, including LOW (<56 bits) and the protocol SSLv2.</p>	
<p>Recommendation</p> <p>It is recommended to switch to strong ciphers and only allow SSLv3 and TLSv1 protocols.</p>	

Test 3 (Control 2)	Result: pass
<p>Testing procedure</p> <p>With the help of SSLDigger we're going to assess the strength of the SSL configuration of the log server.</p>	

Test 3 (Control 2)**Result: pass**

Download the tool from the Foundstone website⁴⁰. Unzip the archive and install it by double clicking on the SSLDigger.msi file. Once installed start the tool and enter the URL of the log server in the "Address" location bar and hit "Go". You'll see something like this:

The screenshot shows the 'Foundstone SSLDigger v1.0 - SSL Test Results' window. It displays a table of supported ciphers with columns for 'OpenSSL Cipher Name', 'Cipher Description', 'Cipher Strength', and 'Exportable?'. The results are as follows:

OpenSSL Cipher Name	Cipher Description	Cipher Strength	Exportable?
NULL-MD5	Key Exchange: None; Authentication: None; Encryption: None; MAC: MD5	No Security	<input checked="" type="checkbox"/>
NULL-SHA	Key Exchange: None; Authentication: None; Encryption: None; MAC: SHA1	No Security	<input checked="" type="checkbox"/>
EXP-DES-CBC-SHA	Key Exchange: RSA(512); Authentication: RSA; Encryption: DES(40); MAC: SHA1	Weak Security	<input checked="" type="checkbox"/>
EXP-RC2-CBC-MD5	Key Exchange: RSA(512); Authentication: RSA; Encryption: RC2(40); MAC: MD5	Weak Security	<input checked="" type="checkbox"/>
EXP-RC4-MD5	Key Exchange: RSA(512); Authentication: RSA; Encryption: RC4(40); MAC: MD5	Weak Security	<input checked="" type="checkbox"/>
EXP1024-DHE-DSS-DES-CBC-SHA	Key Exchange: EDH (EXPORT - 1024); Authentication: DSS; Encryption: DES(56); MAC: SHA1	Weak Security	<input checked="" type="checkbox"/>
EXP1024-DHE-DSS-RC4-SHA	Key Exchange: EDH (EXPORT - 1024); Authentication: DSS; Encryption: RC4(56); MAC: SHA1	Weak Security	<input checked="" type="checkbox"/>
EXP1024-DES-CBC-SHA	Key Exchange: RSA (EXPORT - 1024); Authentication: RSA; Encryption: DES(56); MAC: SHA1	Weak Security	<input checked="" type="checkbox"/>
EXP1024-RC4-SHA	Key Exchange: RSA (EXPORT - 1024); Authentication: RSA; Encryption: RC4(56); MAC: MD5	Weak Security	<input checked="" type="checkbox"/>
DES-CBC-SHA	Key Exchange: RSA; Authentication: RSA; Encryption: DES(56); MAC: SHA1	Weak Security	<input type="checkbox"/>
ADH-AES128-SHA	Key Exchange: ADH; Authentication: RSA; Encryption: AES(128); MAC: SHA1	Weak Security	<input type="checkbox"/>
ADH-AES256-SHA	Key Exchange: ADH; Authentication: RSA; Encryption: AES(256); MAC: SHA1	Weak Security	<input type="checkbox"/>
DH-DSS-AES128-SHA	Key Exchange: DH; Authentication: DSS; Encryption: AES(128); MAC: SHA1	Strong Security	<input type="checkbox"/>
DH-RSA-AES128-SHA	Key Exchange: DH; Authentication: RSA; Encryption: AES(128); MAC: SHA1	Strong Security	<input type="checkbox"/>
DHE-DSS-RC4-SHA	Key Exchange: EDH; Authentication: DSS; Encryption: RC4(128); MAC: SHA1	Strong Security	<input type="checkbox"/>
DHE-DSS-AES128-SHA	Key Exchange: EDH; Authentication: DSS; Encryption: AES(128); MAC: SHA1	Strong Security	<input type="checkbox"/>
DHE-RSA-AES128-SHA	Key Exchange: EDH; Authentication: RSA; Encryption: AES(128); MAC: SHA1	Strong Security	<input type="checkbox"/>
RC4-MD5	Key Exchange: RSA; Authentication: RSA; Encryption: RC4(128); MAC: MD5	Strong Security	<input type="checkbox"/>
RC4-SHA	Key Exchange: RSA; Authentication: RSA; Encryption: RC4(128); MAC: SHA1	Strong Security	<input type="checkbox"/>
AES128-SHA	Key Exchange: RSA; Authentication: RSA; Encryption: AES(128); MAC: SHA1	Strong Security	<input type="checkbox"/>
DES-CBC3-SHA	Key Exchange: RSA; Authentication: RSA; Encryption: 3DES(128); MAC: SHA1	Strong Security	<input type="checkbox"/>
DH-DSS-AES256-SHA	Key Exchange: DH; Authentication: DSS; Encryption: AES(256); MAC: SHA1	Excellent Security	<input type="checkbox"/>
DH-RSA-AES256-SHA	Key Exchange: DH; Authentication: RSA; Encryption: AES(256); MAC: SHA1	Excellent Security	<input type="checkbox"/>
DHE-DSS-AES256-SHA	Key Exchange: EDH; Authentication: DSS; Encryption: AES(256); MAC: SHA1	Excellent Security	<input type="checkbox"/>
DHE-RSA-AES256-SHA	Key Exchange: EDH; Authentication: RSA; Encryption: AES(256); MAC: SHA1	Excellent Security	<input type="checkbox"/>
AES256-SHA	Key Exchange: RSA; Authentication: RSA; Encryption: AES(256); MAC: SHA1	Excellent Security	<input type="checkbox"/>

Illustration 7: SSLDigger scanning for supported ciphers

Once the scan is complete, it will display the results. Here is what SSLDigger found:

Ciphers supported:

No security:	0
Weak security:	4
Strong security:	5
Excellent security:	2

OpenSSL Name	Display Name	Strength
EXP-DES-CBC-SHA	Key Exchange: RSA(512); Authentication: RSA; Encryption: DES(40); MAC: SHA1	Weak Security
EXP-RC2-CBC-MD5	Key Exchange: RSA(512); Authentication: RSA; Encryption: RC2(40); MAC: MD5	Weak Security
EXP-RC4-MD5	Key Exchange: RSA(512); Authentication: RSA; Encryption: RC4(40); MAC: MD5	Weak Security
DES-CBC-SHA	Key Exchange: RSA; Authentication: RSA; Encryption: DES(56); MAC: SHA1	Weak Security
DHE-RSA-AES128-SHA	Key Exchange: EDH; Authentication: RSA; Encryption: AES(128); MAC: SHA1	Strong Security
RC4-MD5	Key Exchange: RSA; Authentication: RSA; Encryption: RC4(128); MAC: MD5	Strong Security

⁴⁰ SSLDigger™ : <http://www.foundstone.com/resources/proddesc/ssldigger.htm>

Test 3 (Control 2)		Result: pass
OpenSSL Name	Display Name	Strength
RC4-SHA	Key Exchange: RSA; Authentication: RSA; Encryption: RC4(128); MAC: SHA1	Strong Security
AES128-SHA	Key Exchange: RSA; Authentication: RSA; Encryption: AES(128); MAC: SHA1	Strong Security
DES-CBC3-SHA	Key Exchange: RSA; Authentication: RSA; Encryption: 3DES(128); MAC: SHA1	Strong Security
DHE-RSA-AES256-SHA	Key Exchange: EDH; Authentication: RSA; Encryption: AES(256); MAC: SHA1	Excellent Security
AES256-SHA	Key Exchange: RSA; Authentication: RSA; Encryption: AES(256); MAC: SHA1	Excellent Security

The total number of ciphers used in testing was 26. SSLDigger grades the overall security with a B.

Findings
SSLDigger found 4 weak supported ciphers.

Recommendation
As in the previous tests, it is recommended to only support strong ciphers and disable all weak cipher suites.

Control 3 - Verify that no unneeded services and daemons are running

Test 1 (Control 3)	Result: pass
Testing procedure	
This test verifies what services are running. First, we're checking which runlevel is set as default:	
<pre># grep initdefault /etc/inittab id:3:initdefault:</pre>	
This means that by default, the server is booting into runlevel 3. We need this information to see what services are running in this specific runlevel. Now let's see which services are running:	
<pre># chkconfig --list syslog 0:off 1:off 2:on 3:on 4:on 5:on 6:off mdmonitor 0:off 1:off 2:on 3:on 4:on 5:on 6:off readahead_early 0:off 1:off 2:off 3:off 4:off 5:on 6:off atd 0:off 1:off 2:off 3:off 4:off 5:off 6:off gpm 0:off 1:off 2:off 3:off 4:off 5:off 6:off iptables 0:off 1:off 2:on 3:on 4:on 5:on 6:off cpuspeed 0:off 1:on 2:on 3:on 4:on 5:on 6:off dc_server 0:off 1:off 2:off 3:off 4:off 5:off 6:off apmd 0:off 1:off 2:on 3:on 4:on 5:on 6:off cups 0:off 1:off 2:off 3:off 4:off 5:off 6:off anacron 0:off 1:off 2:on 3:on 4:on 5:on 6:off httpd 0:off 1:off 2:on 3:on 4:on 5:on 6:off netdump 0:off 1:off 2:off 3:off 4:off 5:off 6:off crond 0:off 1:off 2:on 3:on 4:on 5:on 6:off acpid 0:off 1:off 2:off 3:on 4:on 5:on 6:off NetworkManager 0:off 1:off 2:off 3:off 4:off 5:off 6:off ntpd 0:off 1:off 2:on 3:on 4:on 5:on 6:off</pre>	

Test 1 (Control 3)							Result: pass	
readahead	0:off	1:off	2:off	3:off	4:off	5:on	6:off	
snmptrapd	0:off	1:off	2:off	3:off	4:off	5:off	6:off	
haldaemon	0:off	1:off	2:off	3:off	4:off	5:off	6:off	
arpwatch	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
messagebus	0:off	1:off	2:off	3:on	4:on	5:on	6:off	
xinetd	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
saslauthd	0:off	1:off	2:off	3:off	4:off	5:off	6:off	
irqbalance	0:off	1:off	2:off	3:on	4:on	5:on	6:off	
psacct	0:off	1:off	2:off	3:off	4:off	5:off	6:off	
autofs	0:off	1:off	2:off	3:on	4:on	5:on	6:off	
dc_client	0:off	1:off	2:off	3:off	4:off	5:off	6:off	
smartd	0:off	1:off	2:off	3:off	4:off	5:off	6:off	
sshd	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
kudzu	0:off	1:off	2:off	3:on	4:on	5:on	6:off	
rhnsd	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
netplugd	0:off	1:off	2:off	3:off	4:off	5:off	6:off	
diskdump	0:off	1:off	2:off	3:off	4:off	5:off	6:off	
ipmi	0:off	1:off	2:off	3:off	4:off	5:off	6:off	
snmpd	0:off	1:off	2:off	3:off	4:off	5:off	6:off	
network	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
rawdevices	0:off	1:off	2:off	3:on	4:on	5:on	6:off	

Findings

Only the necessary services are running in runlevel 3.

Recommendation

There are some services like atd, gpm and cups which are installed but not running, as they are disabled in all runlevels. It is recommended that only services which are needed to run are installed. Those services should be uninstalled.

Test 2 (Control 3)		Result: pass	
Testing procedure			
Test 2 checks the <code>xinet.d</code> configuration. There are two ways to verify the running <code>xinet.d</code> services. The command <code>'chkconfig --list'</code> also lists services started by <code>xinet.d</code> . The other method is the manual verification of the configuration scripts the directory <code>/etc/xinet.d/</code> for each service. Each service has a directive called <code>'disable'</code> .			
<pre># chkconfig -list xinetd based services: vnetd: on rsync: off cups-lpd: off time-udp: off echo-udp: off krb5-telnet: off daytime-udp: off kshell: off bpjava-msvc: on time: off bpcd: on echo: off chargen: off klogin: off</pre>			

Test 2 (Control 3)	Result: pass
<pre>vopied: on eklogin: off daytime: off chargen-udp: off</pre>	
<p>Let's double check this result with the manual verification.</p> <pre># grep -H disable /etc/xinetd.d/* /etc/xinetd.d/bpcd: disable = no /etc/xinetd.d/bpjava-msvc: disable = no /etc/xinetd.d/chargen: disable = yes /etc/xinetd.d/chargen-udp: disable = yes /etc/xinetd.d/cups-lpd: disable = yes /etc/xinetd.d/daytime: disable = yes /etc/xinetd.d/daytime-udp: disable = yes /etc/xinetd.d/echo: disable = yes /etc/xinetd.d/echo-udp: disable = yes /etc/xinetd.d/eklogin: disable = yes /etc/xinetd.d/klogin: disable = yes /etc/xinetd.d/krb5-telnet: disable = yes /etc/xinetd.d/kshell: disable = yes /etc/xinetd.d/rsync: disable = yes /etc/xinetd.d/time: disable = yes /etc/xinetd.d/time-udp: disable = yes /etc/xinetd.d/vnetd: disable = no /etc/xinetd.d/vopied: disable = no</pre>	
<p>Findings The two tests showed the same results and only the allowed services are running.</p>	
<p>Recommendation None</p>	

Test 3 (Control 3)	Result: pass
<p>Testing procedure This test verifies in yet another way what processes/services are running.</p>	
<pre># ps aux USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND root 1 0.0 0.0 2676 512 ? S Mar22 0:26 init [3] root 2 0.0 0.0 0 0 ? S Mar22 1:06 [migration/0] root 3 0.0 0.0 0 0 ? SN Mar22 0:04 [ksoftirqd/0] root 4 0.0 0.0 0 0 ? S Mar22 1:27 [migration/1] root 5 0.0 0.0 0 0 ? SN Mar22 0:03 [ksoftirqd/1] root 6 0.0 0.0 0 0 ? S Mar22 1:02 [migration/2] root 7 0.0 0.0 0 0 ? SN Mar22 0:03 [ksoftirqd/2] root 8 0.0 0.0 0 0 ? S Mar22 1:25 [migration/3] root 9 0.0 0.0 0 0 ? SN Mar22 0:03 [ksoftirqd/3] root 10 0.0 0.0 0 0 ? S< Mar22 0:02 [events/0] root 11 0.0 0.0 0 0 ? S< Mar22 0:00 [events/1] root 12 0.0 0.0 0 0 ? S< Mar22 0:00 [events/2] root 13 0.0 0.0 0 0 ? S< Mar22 0:00 [events/3] root 14 0.0 0.0 0 0 ? S< Mar22 0:00 [khelper] root 15 0.0 0.0 0 0 ? S< Mar22 0:00 [kacpid] root 45 0.0 0.0 0 0 ? S< Mar22 0:00 [kblockd/0] root 46 0.0 0.0 0 0 ? S< Mar22 0:00 [kblockd/1]</pre>	

Test 3 (Control 3)										Result: pass	
root	47	0.0	0.0	0	0	?	S<	Mar22	0:00	[kblockd/2]	
root	48	0.0	0.0	0	0	?	S<	Mar22	0:00	[kblockd/3]	
root	61	0.0	0.0	0	0	?	S<	Mar22	0:00	[aio/0]	
root	62	0.0	0.0	0	0	?	S<	Mar22	0:00	[aio/1]	
root	63	0.0	0.0	0	0	?	S<	Mar22	0:00	[aio/2]	
root	64	0.0	0.0	0	0	?	S<	Mar22	0:00	[aio/3]	
root	49	0.0	0.0	0	0	?	S	Mar22	0:00	[khubd]	
root	60	0.0	0.0	0	0	?	S	Mar22	11:59	[kswapd0]	
root	137	0.0	0.0	0	0	?	S	Mar22	0:00	[kseriod]	
root	215	0.0	0.0	0	0	?	S<	Mar22	0:00	[kmirror]	
root	216	0.0	0.0	0	0	?	S<	Mar22	0:00	[kmir_mon]	
root	224	0.0	0.0	0	0	?	S	Mar22	23:59	[kjournald]	
root	1469	0.0	0.0	0	0	?	S<	Mar22	0:00	[kauditd]	
root	1515	0.0	0.0	0	0	?	S	Mar22	0:00	[kjournald]	
root	2192	0.0	0.0	3048	728	?	Ss	Mar22	1:45	syslogd -m 0 -r	
root	2196	0.0	0.0	2840	468	?	Ss	Mar22	0:00	klogd -x	
root	2206	0.0	0.0	1712	476	?	Ss	Mar22	0:00	irqbalance	
root	2297	0.0	0.0	2528	548	?	Ss	Mar22	0:00	/usr/sbin/acpid	
root	2348	0.0	0.0	2772	872	?	Ss	Mar22	0:00	xinetd -stayalive	
-pidfile /var/run/xinetd.pid											
root	2506	0.0	0.7	27316	16556	?	Ss	Mar22	0:04	/usr/sbin/httpd	
root	2515	0.0	0.0	6612	1120	?	Ss	Mar22	0:01	cron	
root	2831	0.1	0.0	114684	1048	?	Ssl	Mar22	114:38	hpsm	
dbus	3045	0.0	0.0	3620	1068	?	Ss	Mar22	0:00	dbus-daemon-1 -system	
root	3056	0.0	0.0	1512	416	tty1	Ss+	Mar22	0:00	/sbin/mingetty tty1	
root	3058	0.0	0.0	2160	416	tty2	Ss+	Mar22	0:00	/sbin/mingetty tty2	
root	3059	0.0	0.0	1632	416	tty3	Ss+	Mar22	0:00	/sbin/mingetty tty3	
root	3061	0.0	0.0	3488	416	tty4	Ss+	Mar22	0:00	/sbin/mingetty tty4	
root	3062	0.0	0.0	2032	416	tty5	Ss+	Mar22	0:00	/sbin/mingetty tty5	
root	3064	0.0	0.0	2760	416	tty6	Ss+	Mar22	0:00	/sbin/mingetty tty6	
root	17894	0.0	0.0	7192	1404	?	S	Mar22	0:00	cron	
root	556	0.0	0.0	5164	564	?	Ss	Apr18	0:00	rhnsd --interval 240	
root	16038	0.0	0.0	2612	468	?	S<s	Apr18	0:00	udev	
root	17703	0.0	0.0	5144	1692	?	Ss	Apr18	0:00	/usr/sbin/sshd	
ntp	22869	0.0	0.2	5148	5148	?	SLs	Apr18	0:00	ntpd -u ntp:ntp -p	
/var/run/ntpd.pid -g											
apache	19651	0.0	1.2	35292	25424	?	S	Jun04	0:06	/usr/sbin/httpd	
apache	19652	0.0	1.2	35508	25564	?	S	Jun04	0:07	/usr/sbin/httpd	
apache	19653	0.0	1.2	35504	25632	?	S	Jun04	0:07	/usr/sbin/httpd	
apache	19654	0.0	1.2	35372	25480	?	S	Jun04	0:08	/usr/sbin/httpd	
apache	19655	0.0	1.2	35320	25416	?	S	Jun04	0:03	/usr/sbin/httpd	
apache	19656	0.0	1.2	35676	25692	?	S	Jun04	0:04	/usr/sbin/httpd	
apache	19657	0.0	1.2	35568	25636	?	S	Jun04	0:10	/usr/sbin/httpd	
apache	19658	0.0	1.2	35164	25272	?	S	Jun04	0:03	/usr/sbin/httpd	
apache	9241	0.0	1.2	35088	25204	?	S	Jun06	0:08	/usr/sbin/httpd	
root	5773	0.0	0.1	7492	2236	?	Ss	11:20	0:00	sshd: user01 [priv]	
user01	5775	0.0	0.1	7492	2292	?	S	11:20	0:00	sshd: user01@pts/0	
user01	5776	0.0	0.0	5700	1392	pts/0	Ss	11:20	0:00	-bash	
root	6967	0.0	0.0	4572	1184	pts/0	S	11:25	0:00	su - root	
root	6972	0.0	0.0	5940	1476	pts/0	S	11:25	0:00	-bash	
root	2541	0.0	0.0	2724	748	pts/0	R+	12:56	0:00	ps aux	
# netstat -lnp --tcp -udp											
Active Internet connections (only servers)											
Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program name					
tcp	0	0	0.0.0.0:13782	0.0.0.0:*	LISTEN	2348/xinetd					
tcp	0	0	0.0.0.0:13783	0.0.0.0:*	LISTEN	2348/xinetd					
tcp	0	0	0.0.0.0:13722	0.0.0.0:*	LISTEN	2348/xinetd					
tcp	0	0	0.0.0.0:13724	0.0.0.0:*	LISTEN	2348/xinetd					
tcp	0	0	:::22	:::*	LISTEN	17703/sshd					
tcp	0	0	:::443	:::*	LISTEN	2506/httpd					
udp	0	0	0.0.0.0:514	0.0.0.0:*		2192/syslogd					
udp	0	0	192.169.2.49:123	0.0.0.0:*		22869/ntpd					
udp	0	0	127.0.0.1:123	0.0.0.0:*		22869/ntpd					
udp	0	0	0.0.0.0:123	0.0.0.0:*		22869/ntpd					
udp	0	0	:::123	:::*		22869/ntpd					

Test 3 (Control 3)	Result: pass
Findings	
Both commands show only approved processes and open ports, i.e. xinetd, ssh, httpd, syslog and ntpd.	
Recommendation	
None	

Test 4 (Control 3)	Result: pass																												
Testing procedure																													
We're going to verify the open ports from yet another perspective, from the remote side.																													
<pre># nmap -sS -P0 -sV -p1-65535 log_server</pre> <p>Starting Nmap 4.03 (http://www.insecure.org/nmap/) at 2006-06-09 13:08 CEST Interesting ports on log_server.corp.com (192.168.2.49): (The 65526 ports scanned but not shown below are in state: closed)</p> <table border="1"> <thead> <tr> <th>PORT</th> <th>STATE</th> <th>SERVICE</th> <th>VERSION</th> </tr> </thead> <tbody> <tr> <td>22/tcp</td> <td>open</td> <td>ssh</td> <td>OpenSSH 3.9p1 (protocol 2.0)</td> </tr> <tr> <td>443/tcp</td> <td>open</td> <td>ssl/http</td> <td>Apache httpd 2.0.52 ((Red Hat))</td> </tr> <tr> <td>13722/tcp</td> <td>open</td> <td>netbackup</td> <td>Veritas Netbackup javalistener</td> </tr> <tr> <td>13724/tcp</td> <td>open</td> <td>vnetd</td> <td>Veritas Netbackup Network Utility</td> </tr> <tr> <td>13782/tcp</td> <td>open</td> <td>bpcd</td> <td>Veritas Netbackup (refused)</td> </tr> <tr> <td>13783/tcp</td> <td>open</td> <td>arcserve</td> <td>ARCserve Discovery</td> </tr> </tbody> </table> <p>Service Info: OS: Unix</p> <p>Nmap finished: 1 IP address (1 host up) scanned in 31.028 seconds</p>		PORT	STATE	SERVICE	VERSION	22/tcp	open	ssh	OpenSSH 3.9p1 (protocol 2.0)	443/tcp	open	ssl/http	Apache httpd 2.0.52 ((Red Hat))	13722/tcp	open	netbackup	Veritas Netbackup javalistener	13724/tcp	open	vnetd	Veritas Netbackup Network Utility	13782/tcp	open	bpcd	Veritas Netbackup (refused)	13783/tcp	open	arcserve	ARCserve Discovery
PORT	STATE	SERVICE	VERSION																										
22/tcp	open	ssh	OpenSSH 3.9p1 (protocol 2.0)																										
443/tcp	open	ssl/http	Apache httpd 2.0.52 ((Red Hat))																										
13722/tcp	open	netbackup	Veritas Netbackup javalistener																										
13724/tcp	open	vnetd	Veritas Netbackup Network Utility																										
13782/tcp	open	bpcd	Veritas Netbackup (refused)																										
13783/tcp	open	arcserve	ARCserve Discovery																										
After scanning the TCP ports, we're scanning all UDP ports:																													
<pre># nmap -sU -P0 -p1-65535 log_server</pre> <p>Starting Nmap 4.03 (http://www.insecure.org/nmap/) at 2006-06-12 14:44 CEST Interesting ports on log_server.corp.com (192.168.2.49): (The 65526 ports scanned but not shown below are in state: closed)</p> <table border="1"> <thead> <tr> <th>PORT</th> <th>STATE</th> <th>SERVICE</th> </tr> </thead> <tbody> <tr> <td>123/udp</td> <td>open filtered</td> <td>ntp</td> </tr> <tr> <td>514/udp</td> <td>open filtered</td> <td>syslog</td> </tr> </tbody> </table> <p>Nmap finished: 1 IP address (1 host up) scanned in 1476.177 seconds</p>		PORT	STATE	SERVICE	123/udp	open filtered	ntp	514/udp	open filtered	syslog																			
PORT	STATE	SERVICE																											
123/udp	open filtered	ntp																											
514/udp	open filtered	syslog																											

Findings	
The portscan confirms the test done on the system, there are only approved ports open.	
Recommendation	
None	

Control 4 - Identify common vulnerabilities in the OS

Test 1 (Control 4)	Result: pass
Testing procedure	
This test involves the Nessus vulnerability scanner. Here are two screen shots of the Nessus setup and it's report :	

Test 1 (Control 4)

Result: pass

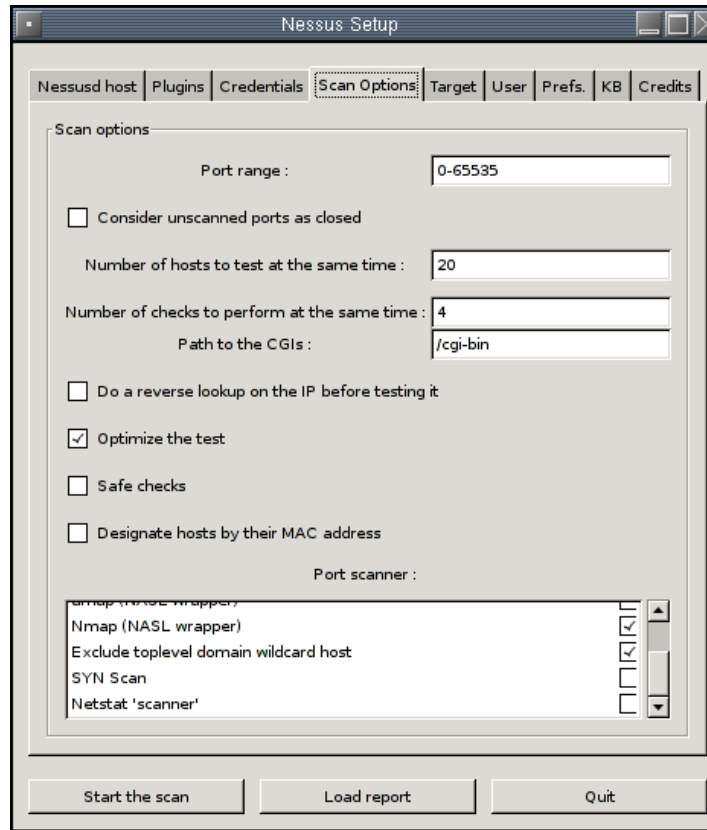


Illustration 8: Nessus Scan Options



Illustration 9: Nessus Report

Test 1 (Control 4)	Result: pass
The following was found by Nessus (list of open ports):	
<ul style="list-style-type: none"> • ssh (22/tcp) (Security notes found) • https (443/tcp) (Security notes found) • bpjava-msvc (13722/tcp) (Security notes found) • vnetd (13724/tcp) • bpcd (13782/tcp) • vopied (13783/tcp) • general/icmp (Security notes found) 	
Findings	
Nessus listed some security notes, however no security holes were found.	
Recommendation	
None	

Control 5 - Verify that the OS is up to date with security patches

Test 1 (Control 5)	Result: fail
Testing procedure	
This test verifies if all patches have been installed. The following command lists all available packages:	
<pre># up2date --dry-run Fetching Obsoletes list for channel: rhel-i386-ws-4... Fetching rpm headers... ##### Name Version Rel ----- ethereal 0.99.0 EL4.2 i386 ipsec-tools 0.3.3 6.rhel4.1 i386 kernel 2.6.9 34.0.1.EL i686 kernel-devel 2.6.9 34.0.1.EL i686 kernel-hugemem-devel 2.6.9 34.0.1.EL i686 kernel-smp 2.6.9 34.0.1.EL i686 kernel-smp-devel 2.6.9 34.0.1.EL i686 libtiff 3.6.1 10 i386 php 4.3.9 3.12 i386 php-ldap 4.3.9 3.12 i386 php-mysql 4.3.9 3.12 i386 php-pear 4.3.9 3.12 i386 ruby-libs 1.8.1 7.EL4.3 i386 Testing package set / solving RPM inter-dependencies... ##### Name Version Rel ----- ethereal 0.99.0 EL4.2 i386 ipsec-tools 0.3.3 6.rhel4.1 i386 kernel 2.6.9 34.0.1.EL i686</pre>	

Test 1 (Control 5)			Result: fail
kernel-devel	2.6.9	34.0.1.EL	i686
kernel-hugemem-devel	2.6.9	34.0.1.EL	i686
kernel-smp	2.6.9	34.0.1.EL	i686
kernel-smp-devel	2.6.9	34.0.1.EL	i686
libtiff	3.6.1	10	i386
php	4.3.9	3.12	i386
php-ldap	4.3.9	3.12	i386
php-mysql	4.3.9	3.12	i386
php-pear	4.3.9	3.12	i386
ruby-libs	1.8.1	7.EL4.3	i386
The output lists the packages which are ready to install. Those packages have not yet been installed.			
Findings			
There are some packages which haven't been updated.			
Recommendation			
It is recommended that the system is updated in a regular interval.			

Control 6 - Verify user accounts and their privileges

Test 1 (Control 6)		Result: pass
Testing procedure		
We are now verifying that the only account with UID 0 is root:		
<pre># awk -F: '(\$3==0){print \$1}' /etc/passwd root</pre>		
Findings		
There is only one root account with UID 0.		
Recommendation		
None		

Test 2 (Control 6)		Result: pass
Testing procedure		
This test verifies that no enabled user account is having a null password:		
<pre># awk -F: '(\$2==""){print \$1}' /etc/shadow</pre>		
There is no output, which means there is no account with an empty password field.		
Findings		
All account have a password set.		
Recommendation		
None		

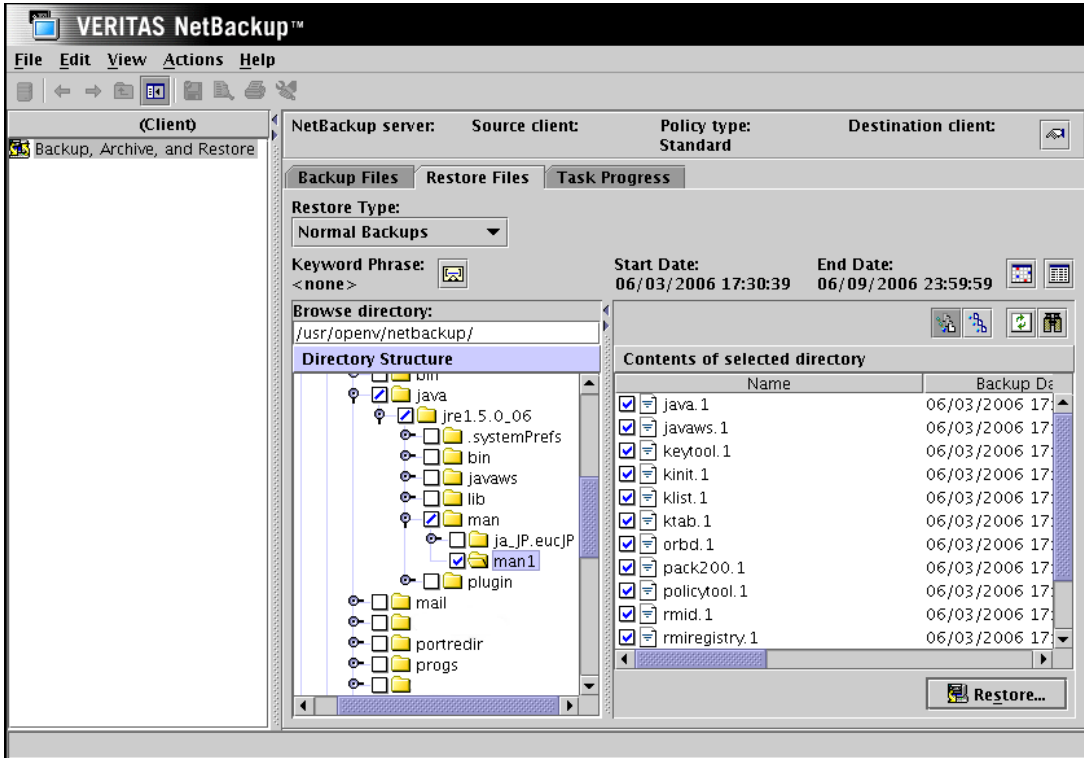
Test 3 (Control 6)	Result: pass
<p>Testing procedure By inspecting the file <code>/etc/passwd</code> we are verifying that no unnecessary user accounts are set up on the system:</p> <pre># cat /etc/passwd root:x:0:0:root:/root:/bin/bash bin:x:1:1:bin:/bin:/sbin/nologin daemon:x:2:2:daemon:/sbin:/sbin/nologin adm:x:3:4:adm:/var/adm:/sbin/nologin lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin sync:x:5:0:sync:/sbin:/bin/sync shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown halt:x:7:0:halt:/sbin:/sbin/halt nobody:x:99:99:Nobody:/:/sbin/nologin rpm:x:37:37:/:/var/lib/rpm:/sbin/nologin haldaemon:x:68:68:HAL daemon:/:/sbin/nologin sshd:x:74:74:Privilege-separated SSH:/var/empty/ssh:/sbin/nologin mailnull:x:47:47:/:/var/spool/mqueue:/sbin/nologin smmmsp:x:51:51:/:/var/spool/mqueue:/sbin/nologin pcap:x:77:77:/:/var/arpwatch:/sbin/nologin apache:x:48:48:Apache:/var/www:/sbin/nologin ntp:x:38:38:/:etc/ntp:/sbin/nologin hpsmh:x:79:79:/:opt/hp/hpsmh:/sbin/nologin user01:x:500:500:User 01:/home/user01:/bin/bash user02:x:501:501:User 02:/home/user02:/bin/bash</pre> <p>Most users are created by the system and are disabled. Only two users (user01 and user02) are “normal” user accounts.</p>	
<p>Findings There are no unused user accounts set up.</p>	
<p>Recommendation None</p>	

Test 4 (Control 6)	Result: pass
<p>Testing procedure This test is using John for a password strength verification. First, we have to get a copy of the password file.</p> <pre># unshadow /etc/passwd /etc/shadow > mypasswd</pre> <p>This created a file <code>mypasswd</code> with the password hashes. Now let's run John:</p> <pre>\$ john mypasswd Loaded 4 passwords with 4 different salts (FreeBSD MD5 [32/32]) guesses: 0 time: 0:00:02:03 (3) c/s: 4148 trying: mullan1 guesses: 0 time: 0:00:14:52 (3) c/s: 4188 trying: schOME guesses: 0 time: 0:00:33:29 (3) c/s: 4162 trying: amot29 guesses: 0 time: 0:01:16:15 (3) c/s: 4143 trying: Bageg5 guesses: 0 time: 0:01:26:23 (3) c/s: 4136 trying: smOUAQ \$ john -show mypasswd 0 passwords cracked, 4 left</pre>	

Test 4 (Control 6)	Result: pass
<p>John displays a status information when you hit any key. You see the first status message after 2 minutes and 3 seconds, and the last one after 1 hour, 26 minutes and 23 seconds, and John still hasn't been able to crack the passwords. The last column is showing the actual password which is being tried at that time.</p>	
<p>Findings No passwords have been found in one hour.</p>	
<p>Recommendation None</p>	

Control 7 - Verify that the backup procedure is in place and is working correctly

Test 1 (Control 7)	Result: pass
<p>Testing procedure</p> <p>This test performs a restore of several test files to ensure that backup can be restored. All tested files must be able to restore completely.</p> <p>First, we delete some arbitrary files, let's delete a whole directory:</p> <pre>\$ ls ja ja_JP.eucJP man1 \$ rm -r man1/ \$ ls ja ja_JP.eucJP</pre> <p>The directory <code>man1</code> has just been deleted. Now we're going to use the NetBackup graphical user interface (the NetBackup Administration Console), a Java-based, graphical-user interface to restore the directory. There is also a character-based, menu interface that is started by running the <code>bpadm</code> command.</p> <p>Start the NetBackup Administration Console:</p> <pre>/usr/opensv/netbackup/bin/jnbSA</pre> <p>Log in with your user name and password. In the 'Restore files' tab, select the directory we've just deleted in the file tree. Now click on the "Restore..." button. In the "Restore Files" dialog, verify that "Restore everything to its original location" is checked, then click "Start Restore". When the restore is finished, the status will go to "Successful". Now we are verifying that the restore completed successfully.</p> <pre>\$ ls ja ja_JP.eucJP man1 \$ ls man1/ java.1 kinit.1 orbd.1 rmid.1 tnameserv.1 javaws.1 klist.1 pack200.1 rmiregistry.1 unpack200.1 keytool.1 ktab.1 policytool.1 servertool.1</pre> <p>The <code>man1</code> directory including all files have appeared again.</p>	

Test 1 (Control 7)	Result: pass
	
<i>Illustration 10: NetBackup: Restore Files</i>	

Findings
The restore of the files was successful.

Recommendation
None

Control 8 - Prevent hardware outages

Test 1 (Control 8)	Result: pass
<p>Testing procedure Each power plug has been plugged out and the server continued running as intended.</p>	
<p>Findings Both power supplies are working correctly.</p>	
<p>Recommendation None</p>	

Test 2 (Control 8)	Result: pass
<p>Testing procedure The UPS system documentation has been checked and the system is operated as it is supposed to be. The system, including the emergency generators, is tested once a year.</p>	
<p>Findings The UPS is operated and working correctly.</p>	
<p>Recommendation None</p>	

Control 9 - Prevent physical access for unauthorized people

Test 1 (Control 9)	Result: pass
Control objective Prevent physical access for unauthorized people	
Testing procedure The entrance doors to the computer room have been inspected. There is a biometric access control in place. Only employees with a formal request are set up on the biometric system. The entrance system creates a log file with the date, time, access door and employee name.	
Findings Physical access is protected adequately.	
Recommendation None	

6.2 Summary of findings and recommendations

The audit found three issues which can be improved.

Bogus syslog entries can fill logs

The syslog daemon uses UDP to transfer log entries, which is unreliable and does not provide confidentiality and integrity. It is recommended use a reliable syslog server like syslog-ng and additionally stunnel to encrypt the logs in transfer.

Weak apache SSL configuration

Apache supports weak cipher suites, including LOW (<56 bits) and the protocol SSLv2. It is recommended to switch to strong ciphers and only allow SSLv3 and TLSv1 protocols.

Missing system patches on the OS level

There are multiple packages which haven't been updated to the most recent version. It is recommended that the system is updated in a regular interval. Additionally, unused services should be uninstalled and not only disabled, this prevents local exploits and accidentally enabled services.

7 Conclusion

Log management is an important and essential part of every corporation. There are many benefits of log data like identifying security events / fraudulent activity and forensic analysis. Several federal legislations and regulations require corporations to analyze and retain certain logs. A big challenge in the log management is the vast amount of ever increasing log data. It's a balance of keeping the right amount of data and still being able to analyze them.

The audited log server is a RHEL (Red Hat Enterprise Linux) machine. It's collecting and archiving the log files of multiple devices like firewalls and routers.

The objective of risk management is to spend the right amount of protective measures to protect the IT systems and data. Risk is a function of the likelihood of a given threat exercising a particular potential vulnerability, and the resulting impact of that adverse event on the organization. The definition of risk, threat, vulnerability and impact were described together with examples.

The following three risks have been analyzed:

- Unauthorized access of logs (loss of confidentiality)
- Unauthorized modification of logs
- Loss of logs

Multiple vulnerabilities have been listed for each risk and 9 different control mechanisms have been defined with each having multiple tests. Those tests were explained in detail and then executed. The audit found the following weaknesses:

- bogus syslog entries can fill logs
- missing system patches on the OS level
- weak apache SSL configuration

Each audit finding comes with a recommendation to mitigate the threat or to close the security hole. There were no high risk findings and in general the log server is configured and operated in a secure way.

8 References

NIST, Special Publication 800-30 (July 2002). Risk Management Guide for Information Technology Systems. <http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf>

NIST, Special Publication 800-92 (Draft) (April 2006). Guide to Computer Security Log Management <http://csrc.nist.gov/publications/drafts/DRAFT-SP800-92.pdf>

SANS Institute (2006). AUDIT 507: Auditing Networks, Perimeters & Systems.

Ziegler, Robert L. (2001). Linux Firewalls. Indianapolis: New Riders Publishing.

FBI. 2005 FBI Computer Crime Survey. FBI Publications.
http://www.fbi.gov/page2/jan06/computer_crime_survey011806.htm

NIST, Special Publication 800-42 (2001). Guideline on Network Security Testing.

Ogren, Eric (April 2006). Security Information Lifecycle: Data Retention of Event Logs for Compliance. Enterprise Strategy Group.

Northcutt, Stephen, Shenk, Jerry, Ong, Leonard, & Shackleford, Dave (2005). The Log Management Industry - An Untapped Market. SANS Analyst Program.

© SANS Institute 2006, Author retains full rights.



Upcoming SANS Training

[Click Here for a full list of all Upcoming SANS Events by Location](#)

SANS Cyber Defence Canberra 2017	Canberra, AU	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MDUS	Jun 26, 2017 - Jul 01, 2017	Live Event
SEC564:Red Team Ops	San Diego, CAUS	Jun 29, 2017 - Jun 30, 2017	Live Event
SANS London July 2017	London, GB	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, JP	Jul 05, 2017 - Jul 15, 2017	Live Event
SANS ICS & Energy-Houston 2017	Houston, TXUS	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, SG	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Los Angeles - Long Beach 2017	Long Beach, CAUS	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Munich Summer 2017	Munich, DE	Jul 10, 2017 - Jul 15, 2017	Live Event
SANSFIRE 2017	Washington, DCUS	Jul 22, 2017 - Jul 29, 2017	Live Event
Security Awareness Summit & Training 2017	Nashville, TNUS	Jul 31, 2017 - Aug 09, 2017	Live Event
SANS San Antonio 2017	San Antonio, TXUS	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Hyderabad 2017	Hyderabad, IN	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MAUS	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Prague 2017	Prague, CZ	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UTUS	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS New York City 2017	New York City, NYUS	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Chicago 2017	Chicago, ILUS	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Adelaide 2017	Adelaide, AU	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VAUS	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS San Francisco Fall 2017	San Francisco, CAUS	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Tampa - Clearwater 2017	Clearwater, FLUS	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Network Security 2017	Las Vegas, NVUS	Sep 10, 2017 - Sep 17, 2017	Live Event
SANS Dublin 2017	Dublin, IE	Sep 11, 2017 - Sep 16, 2017	Live Event
SANS Paris 2017	OnlineFR	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced