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.

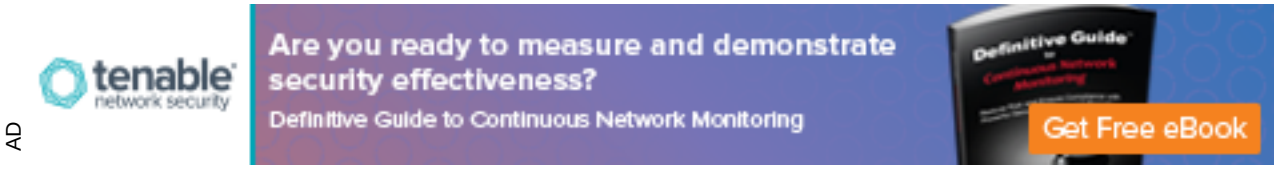
Assessing Vendor Application Security A Practical Way to Begin

Many companies are adopting a preference toward buying vendor software versus building software in-house to meet business needs. Some of the drivers for this preference are integration, scalability, outsourcing, support, speed-to market, process savings, and reducing the cost of information technology (IT). In adopting a preference for purchased software, it becomes critical that companies have an assessment methodology for determining how well each proposed vendor package will meet established business and technical r...

Copyright SANS Institute

Author Retains Full Rights

AD



SANS GSEC Practical Assignment
Version 1.4b Option 1
Administrivia version 2.7
Pre-Approval Submission Date: January 18, 2004
Topic Approval Date: January 24, 2004
Practical Submission Date: February 17, 2004

Assessing Vendor Application Security – A Practical Way to Begin

By: Bart L. Hubbs

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Table of Contents

Abstract....3
Part One: Where to Begin....4
Part Two: Application Architecture....6
2.1: The Tiers....6
2.2: Operating Systems (O/S)....7
2.3: Applications by Tier....7
2.4: Basic Communication....9
Part Three: Network Communications....10
3.1: Ports of Communication....10
3.2: Source Connections....13
3.3: Services....14
3.4: Quick Analysis....16
Part Four: Directory and File Review....18
4.1: Application Install Structure....18
4.2: Classifying the Directories and Files....19
4.3: File Interrogation....20
4.4: Quick Analysis....22
Part Five: Authentication....23
5.1: User Authentication....23
5.2: Tier Authentication....24
5.3: Managing/Controlling Authentication....25
5.4: Quick Analysis....28
Part Six: Authorization....30
6.1: Authorization Decision Locations....30
6.2: Granularity, Consistency, and Simplicity....30
6.3: Dependence on Trust – Know the Gaps....31
6.4: Quick Analysis....32
Part Seven: Auditing....33
7.1: Log Locations and Formats....33
7.2: Logging Configurations, Settings, and Content....33
7.3: Log Protection....33
7.4: Quick Analysis....34
Part Eight: Overall Analysis....35
Summary....36
Bibliography....37
Appendix A....38

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Abstract

Many companies are adopting a preference toward buying vendor software versus building software in-house to meet business needs. Some of the drivers for this preference are integration, scalability, outsourcing, support, speed-to-market, process savings, and reducing the cost of information technology (IT). In adopting a preference for purchased software, it becomes critical that companies have an assessment methodology for determining how well each proposed vendor package will meet established business and technical requirements.

Therefore, the purpose of this paper is to establish a guide for targeting areas of potential concern to the business regarding the security of vendor developed applications that will be deployed in an enterprise environment. This paper is not intended to be a complete guide to assessing vendor applications, but will give the reader a roadmap for gathering relevant information about the proposed application, formulating directed questions to ask the vendor, determining where potential pitfalls may exist, and giving management feedback on security concerns that may influence the final purchasing decision.

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Part One: Where to Begin

The critical parts of beginning any IT related project supporting a business function are deciding where to begin, how to structure the project based upon time and financial constraints, and what the customers of your project expect the output to contain. Answering these questions and then validating your direction is important to achieving success.

Most businesses establish policies, procedures, and/or standards for business operations, financial reporting, IT, etc. Therefore, the recommendation is to first review IT related policies, procedures and/or, standards which may be referred to as governance documents or management directives.

The governance documents may contain high-level concepts and directives at the policy level, but are brought into specifics through the procedures and/or standards. Often business leadership will agree on the policy and depend on IT to develop the procedures and/or high-level standards which support the policy. Some IT organizations will either limit or eliminate the procedures documentation set due to obsolescence issues and instead implement high-level standards. Furthermore, mature and developed IT organizations will issue technical platform standards giving required settings to the detail level. The relationship of these documents can be seen in Diagram 1.

High-Level Governance Documents

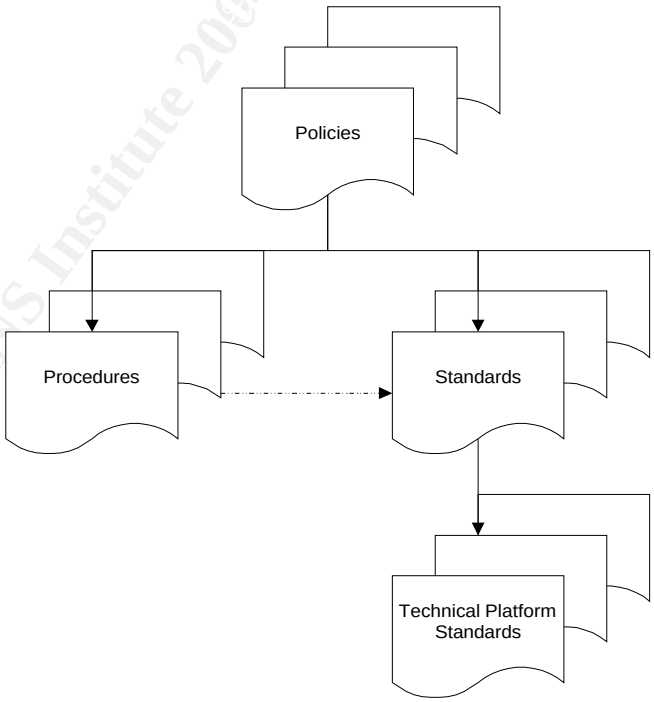


Diagram 1

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

This paper will illustrate analysis methods using a hypothetical company (Great Company) and its implementation of an Enterprise Resource Planning (ERP) system. Great Company is a large company in a highly regulated industry and, therefore, must implement and follow many policies and standards to comply with Federal, State, and Local regulations.

Sample Scenario

The concepts documented in this paper will center on Great Company and its implementation of a vendor developed Enterprise Resource Planning (ERP) system that will be used initially for administration of Human Resources (HR) and Payroll (PR) for the entire company. Furthermore, Great Company has a long-term goal of completing a full implementation of the following additional components:

- General Ledger (GL);
- Accounts Payable (AP);
- Supply Chain (SC); and
- Enterprise Budgeting (EB)

The high-level business goals for this implementation are to reduce administration cost, streamline and improve operations, enable improved reporting and analysis, and enable future flexibility.

Therefore, the first step in this evaluation is to complete an analysis of the company's governance documents to ensure the evaluation and subsequent analysis is foundationally based upon the expectations of management. The first stage of the analysis includes the governance document reference with the specific applicable directive. The analysis document used for Great Company is documented in Appendix A. The specific directives will be used as a measurement standard for this analysis and will referred to as "functional requirements" throughout this paper.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Part Two: Application Architecture

2.1: The Tiers

After analysis of the governance documents, the next step is to determine the general architecture of the application. Basically, this entails identifying the components of the application and how they work together. These application components are often referred to as "tiers" quoted by Chartier as "any number of levels arranged above another, each serving distinct and separate tasks." The tiers may also be referred to as data, business/application, and presentation tiers or layers. Additionally, each tier can be specified further by physical device or logical implementation. Additional explanation of tier definition can be viewed in Chartier's article "Application Architecture: An N-Tier Approach - Part 1."

The tiers that will be presented in our ERP implementation scenario are presented in Diagram 2 below.

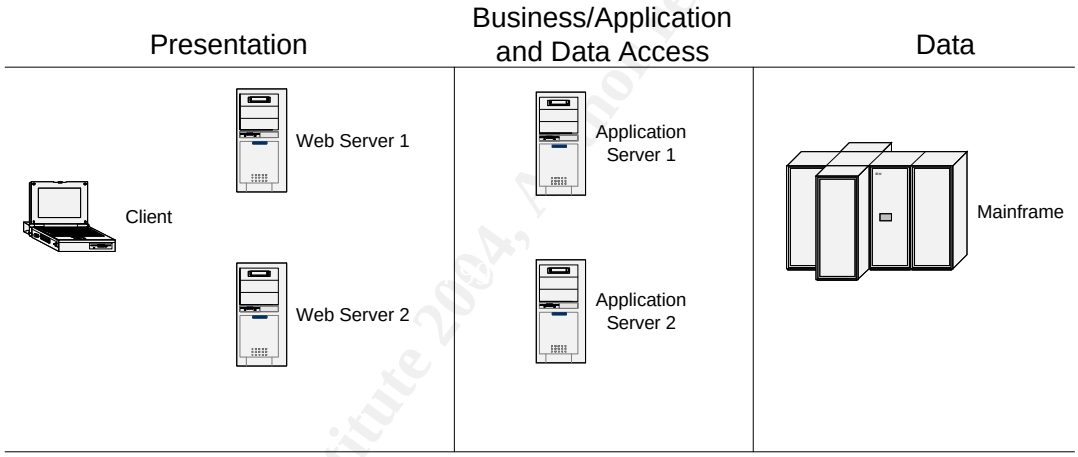


Diagram 2

The presentation tier in the diagram is implemented on either the user's computer (client) and/or on the web server(s). Only one web server is active at any time. The second server is established as a hot backup with automatic failover (active/passive).

The business/application tier is implemented on two servers configured in active/passive mode similar to the web servers. The business/application tier can receive connections directly from the client and/or from the web servers. Additionally, the business/application tier contains multiple drivers which enable connection to different relational database management systems (RDBMS). The connection component will be referred to as the "data access" as detailed in the Chartier article.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

The data tier is implemented on a mainframe; however, most of the routine data management is performed from the application tier through an interface provided by the ERP vendor.

The vendor application account representative should be able to provide the information illustrated in diagram 1. Additionally, the account representative should be able to facilitate discussions with advanced engineers if necessary. The terminology used by the application vendor to describe the tiers may vary; however, carefully crafted questions directed to the account representative and/or the vendor's engineers can help make the translation for analysis purposes.

2.2: Operating Systems (O/S)

The next step will utilize the information gained from establishment of the tiers and define the O/S' that will support the application layers. Many vendor applications are written to allow installation on various O/S'. However, vendors tend to prefer or test one or twp primary O/S configurations. Asking the account representative which top O/S' they use or install most often will help the analyst start to evaluate the vendor's preference. Furthermore, if application performance data is provided by the vendor ask which O/S' were used in the test. This information may give valuable insight on how well the vendor can support the application for the O/S chosen by the company.

The O/S choices in this sample scenario are as follows:

Presentation: Client – Microsoft © Windows 2000; Web Servers – IBM© S80 running AIX 5.1.

Business/Application and Data Access: Application Servers – IBM© S85 running AIX 5.1.

Data: Database Server – IBM© Z series Mainframe running OS/390.

The preference was given to IBM for the O/S and hardware to help ensure compatibility and support. Additionally, a key observation that was made is all "on-line" user type activity on the server-side is provided by the AIX/Unix platform and the main data component was served by the mainframe. This fact combined with the client using Microsoft © Windows 2000 means that three different operating system platforms were utilized.

2.3: Applications by Tier

<u>Presentation Tier</u>
The end-user can access the ERP application using the web browser or a vendor developed Win32 client. The main applications on the server-side presentation

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

tier were IBM® HTTP Server (IHS) and IBM® WebSphere. Additionally, the vendor provided custom JavaScript, and Common Gateway Interface (CGI) scripts which help make the application flexible and cross-platform.

Application Tier

The application servers utilize primarily vendor developed code written in the following languages

- Common business oriented language (Cobol)
- C
- CGI;
- Java; and
- Perl.

Additionally, each development language may require various compilers, editors, libraries, etc. to make each perform correctly. The data access components will require Db2 connection for the final connection. The DB2 connect component "provides the application enablement and robust, highly scalable communication infrastructure for connecting Web, Windows, UNIX, Linux, OS/2 and mobile applications to S/390 and AS/400 data" (IBM). Lastly, vendor source code, binaries, and configuration files will be needed to complete the tier.

Data Tier

The data components primarily resided in the DB2 Relational Database Management System (RDBMS) on a mainframe. However, multiple smaller data and metadata files were installed and accessed from the application-tier. Many of the non-DB2 files were Indexed Sequential Access Method (ISAM) files or regular text files.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

2.4: Basic Communication

After defining each tier, the next step is to determine how the tiers communicate. The easiest method for accomplishing this task is to ask the vendor. Many times the vendor will provide this information in the installation guide. Many of the communication protocols will be standard protocols (i.e. telnet, ssh, ftp, sftp, etc). However, some of the protocols that allow the tiers to communicate will be developed by the vendor to enable or "extend" special features of the application.

The standard protocol analysis will be a fairly straight-forward process that has many reference materials. However, the "vendor proprietary" protocols (vpp) will be significantly more difficult to assess. The analyst should pay special attention to the vpp's during the assessment. The diagram below illustrates the original application tiers with lines that depict how the tiers communicate, often referred to as a data flow diagram.

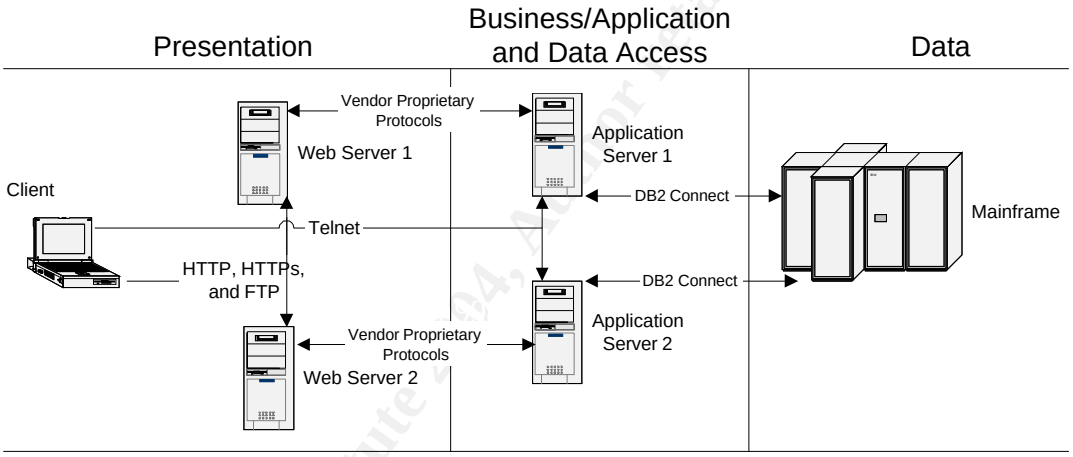


Diagram 3

Most of the protocols are standard protocols such as telnet, http, https, and ftp. However, the connections from the web server(s) to the application server(s) are noted as "vendor proprietary" protocols. Note that three of the four standard protocols noted in the Diagram 3 do not include encryption capability as a part of the protocol. Therefore, telnet, http, and ftp should be included as areas of concern in our analysis. Furthermore, the focus of the analysis should center on the unknown vpp's because protocols developed by vendors are usually not well documented and may provide some "features" that should be assessed.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Part Three: Network Communications

3.1: Ports of Communication

The high-level overview analysis that was performed in the previous sections will help provide background for analysis of the network communication components of the application. The primary concept that should be defined in network communication is the meaning of a "port" as referenced in the concept of a "listening" port. A port as defined by Enders and Hayes "is a 16-bit number, used by the host-to-host protocol to identify to which higher-level protocol or application program (process) it must deliver incoming messages."

The next step is to determine the listening ports on each tier and work to match the listening ports to the diagram. Different methods can be utilized in this process. The two primary methods will be referred to as local and remote methods. The local method utilizes commands on the local machine to report the listening ports while the remote method utilizes the nmap scanning tool which can be downloaded from www.insecure.org. We will highlight the vpp's in the listing for reference.

Application Tier – Local

app01/ $ netstat -af inet|grep -i listen

Protocol	Local Addr	Foreign Addr	State
tcp4	*.ftp	*.*	LISTEN
tcp4	*.sshd	*.*	LISTEN
tcp	*.telnet	*.*	LISTEN
tcp4	*.smux	*.*	LISTEN
tcp	*.shell	*.*	LISTEN
tcp4	*.523	*.*	LISTEN
tcp4	*.db2cdb2i	*.*	LISTEN
tcp4	*.MQSeries	*.*	LISTEN
tcp4	*.1622	*.*	LISTEN
tcp4	*.2620	*.*	LISTEN
tcp4	*.3181	*.*	LISTEN
tcp4	*.vpp01	*.*	LISTEN
tcp4	*.53700	*.*	LISTEN
tcp4	*.5401	*.*	LISTEN
tcp4	*.vpp02	*.*	LISTEN
tcp4	*.nrpe	*.*	LISTEN
tcp4	*.nsca	*.*	LISTEN
tcp4	*.5912	*.*	LISTEN
tcp4	*.clm_lkm	*.*	LISTEN
tcp4	*.clm_smux	*.*	LISTEN
tcp4	*.godm	*.*	LISTEN
tcp4	*.clver	*.*	LISTEN
tcp4	*.clsmuxpd	*.*	LISTEN

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
tcp4        *.32768        *.*        LISTEN
tcp4        *.vpp03        *.*        LISTEN
tcp4        *.vpp04        *.*        LISTEN
tcp4        *.vpp05        *.*        LISTEN
```

The illustration above shows all listening ports on the application-tier servers. Both application servers are configured the same from a network footprint perspective. Clarifying notes on the comments and output displayed above are as follows:

1. The term "network footprint" is often used to describe those services that are visible and available from the network.

2. The service listed with a port number only (i.e. *.32768) indicates that the service was not defined in the /etc/services file.

3. The columns recv-Q and send-Q were removed to allow space and help simplify the netstat listing.

<u>Web Tier - Remote</u>

```
app01/ $ /usr/local/bin/nmap –sT web01
Port      State      Service
22/tcp    open       ssh
23/tcp    open       telnet
80/tcp    open       http
199/tcp   open       smux
443/tcp   open       https
513/tcp   open       login
514/tcp   open       shell
900/tcp   open       unknown
```

The scan illustration on the Web-Tier shows all listening ports the scanning application was able to establish a connection to on the web server. Both web servers are configured identically from a network footprint perspective.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Using the scan listed above, it appears the proprietary protocols are only listening on the application tier. Therefore, the next step is to determine the binary that is holding the port open and how it is started. The process will be illustrated using the application server and will focus on the VPP01, but should be performed on all the vpp's. The tool used to match listening ports to binaries is called "lsof". The tool "displays information about files open to Unix processes" and "runs on many Unix dialects" (Abell).

VPP01

app01 / $ lsof -i|grep -i vpp01

```
su_vpp01 1589276     root    3u  IPv4 0xf1000089c9758b40       0t0  TCP *:vpp01 (LISTEN)
```

This entry indicates the listening binary is running as the root user and it was initiated with the command su_vpp01. The command su_vpp01 was the initial command, but it may not be the actual binary listening on the server. Therefore, a few steps should be followed to verify.

The first step is to use the process ID 1589276 and search the process list for more information. This can be accomplished by using the "ps" command. "The ps command displays a list of the processes currently running on the machine that you are logged into. If no arguments are entered with the ps command, only the "important" processes, that you own (i.e. that you are running) are displayed" (Aspinall). The "grep" command will enable the search of the process list to filter only the entries needed. "The grep command searches one or more files, specified by file, for the text string specified by pattern" (Aspinall).

app01 / $ ps -ef |grep 1589276

```
root 1589276       1   0 01:23:46      -  0:00 /erpapp/ bin/vppServer
```

The output of this command shows the process is actually running vppServer. The next step is to determine the file type of vppServer and su_vpp01. This task can be completed using the file command as follows:

app01 / $ file /erpapp/ bin/vppServer
/erpapp/ bin/vppServer: executable (RISC System/6000) or object module

app01 / $ file /erpapp/ bin/su_vpp01
/erpapp/ bin/su_vpp01: executable (RISC System/6000) or object module

The next step is to determine the file permissions of both su_vpp01 and vppServer.

```
-rwxrwxrwx   1 erp1   users      3260690 Feb 25 2003  su_vpp01
-rwsrwxrwx   1 root   users      32 20410 Feb 23 2001  vppServer
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Both binary files shown have very “open” file permissions meaning that all file permissions are given to all users of the system. The vppServer binary permits anyone to run it as “root” (i.e. the super user) while also providing the opportunity to modify the file. This is a red flag issue and should be discussed with the application vendor to determine if the permissions can be reduced and by evaluating if the process can be run as a user with a lower privilege.

3.2: Source Connections

Now that the listening services are defined and the vpp’s are mapped, the next step is to determine the devices connecting to each listening service. This process may also provide a good blueprint to help better secure the server (i.e. server hardening) later.

The ideal place to begin this process is to discuss the data flow diagram and the list of listening services with the application vendor. The main purpose of the vendor discussion is to ensure each service is needed and to determine the purpose of each vpp. The vendor should be able to communicate the purpose of each vpp, the server or tier that should be expected as the source connection. Additionally, the analyst can utilize the “lsof” command while the system is being utilized to determine the devices that are actually connecting to each “listening” port by piping the output of “lsof” into the “grep” command filtering for established connections as follows: “app01 / $ lsof –i|grep –i established”. The purpose of gathering this information is to help the analyst evaluate the possibility of controlling the source connections to the vpp’s and other listening binaries to minimize the opportunity of exploit.

The example situation shows five vpp’s listening on the application tier. Three of the vpp’s are started dynamically and two are started via “inetd”. Some useful tools that can run on the application tier and can help control source connections to the vpp’s are tcpwrappers and iptables. These tools can help control source connections and are good mitigating controls for all listening services. They are even more important to consider on vpp’s that must run with elevated privileges (i.e. root).

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

3.3: Services

The goal of this paper is not to be code review guide, but a quick method to do a calculated "sniff test" on a listening service/binary to check for potential buffer overrun opportunities. According to Howard and LeBlanc, "buffer overruns have been a know security problem for quite some time" (23) Howard and LeBlanc site the "strcpy function as inherently unsafe" (81) and state that the "sprintf function is right up there with strcpy in terms of mischief it can cause" (84). Furthermore, Howard and Leblanc state "this function is just disaster waiting to happen" (88) writing about the "gets" function.

Therefore, a quick method to determine if a binary may be susceptible to a buffer overrun is to determine if certain functions are used in the code and analyze the potential for exploit. However, often the vendor does not provide the source code for the binaries used to enable proprietary protocols. Therefore, a command called "strings" can be utilized to provide some possible clues about some of the functions used in the vendor provided listening binaries.

In section 3.1, a process was used to locate the binary holding a port open on the server to provide a remote service. The quick analysis can be used on the listening binary vppServer to evaluate if the logic contains one of the inherently unsafe functions. This process is not a bullet proof method for code analysis, but it is quick check method that can help the analyst locate potential areas of concern needing additional investigation. The process is illustrated below with commands and related output.

```
app01 / $ strings vppServer|grep -i strcpy
app01 / $  strings vppServer|grep -i sprintf
sprintf
app01 / $  strings vppServer|grep -i gets
```

Based upon the strings test, the listening binary has at least one instance of the "sprintf" function used. No positives were received back on the "strcpy" or the "gets" function. Howard and LeBlanc indicate that "there is almost no way to use this function safely" (84) regarding the "sprintf" function.

Another "sniff" test that can be used to evaluate vpp's is to use scanning technologies like nmap, Nessus (downloadable at http://www.nessus.org/download.html), SPIKE, or sharefuzz (downloadable at http://www.atstake.com/research/tools/vulnerability_scanning/) to detect potential problems via brute-force. This example utilized nmap to scan all ports on the application-tier servers and it was noted that the user application stopped processing during the scan. Additionally, after restarting the nmap scan the service consistently stopped processing after the port opened by vppServer was connected to by the tool.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Therefore, a cause for concern exists that should be discussed with the vendor and should become part of the analysis. The vppServer binary runs as the “root” user, has open file permissions, has the potential for a buffer overrun, and stops functioning after a simple port scan.

The process followed for the VPP01 service should be performed on all vendor listening services. This process may seem time consuming, but it can be easily scripted and additional “checks” can be implemented in the script to search for other unsafe functions.

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

3.4: Quick Analysis

.

Governance Document Analysis and Review Program		
Policy, Procedure, Standard	*Functional Requirement*	*Technical Implementation*
Security Policy 2 Encryption Standard	Encryption for confidential network communications • *112-bit key length for symmetric encryption (such as Triple 56 bit DES);* • *2048-bit key length for asymmetric encryption; and* • *160 bits for Elliptical Curve systems.*	HTTPs is the only service used by the end user implementing encryption at the level required. Telnet, http, and ftp are all cleat-text protocols utilized by user applications. <u>Failed</u>
Security Policy 1 Encryption Standard	Password Transmission • *Passwords may be sent over the network only when encrypted or hashed.*	The telnet, http, and ftp applications, protocols enable the user to connect to various components of the application and require authentication. The users' password will be sent over the network without encryption or a hash using these protocols. <u>Failed</u>
Security Policy 1 Server Administration Standard	Services • *Services, which are not required for the role of the server, must be disabled.* • *All "listening" services must be documented and patched regularly.* • *Appropriate care should be taken to limit source connections as possible.*	Services that are installed and running by default may need to be disabled before implementation.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Policy, Procedure, Standard	*Functional Requirement*	*Technical Implementation*
	• *All "listening" services should be able to withstand a routine port scan.*	Vendor Proprietary Protocols were not able to withstand a routine "port scan" without shutting down abnormally. Failed

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Part Four: Directory and File Review

4.1: Application Install Structure

The purpose of this phase is to establish the file system footprint of the vendor application. It is difficult and sometimes impossible to successfully secure an application that is not understood. Therefore, it is important to understand the location of critical binaries, configuration files, scripts, databases, logs, and metadata components to enable development of a security strategy. Furthermore, it is critical to have a good understanding of the application file structure to begin and successfully complete most troubleshooting activities.

Most application vendors will utilize industry standard RDBMS for the data tier. Therefore, the vendor will probably reference the corresponding RDBMS documentation for any database installation structure questions. For this review focus will be on the presentation and application tiers residing on the servers. The database analysis and security are extremely important, but will not be covered in any depth in this paper

The focus of this step is to establish diagram of the server application installation and any add-on components needed. In the sample scenario, the installation structure is visualized in diagram 4. Furthermore, a good idea is to ask the vendor to review the diagram for accuracy if it was not already provided in the documentation.

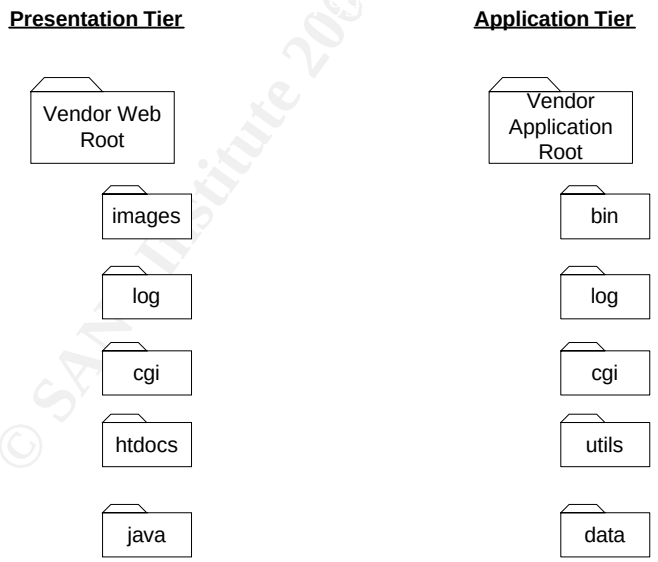


Diagram 4

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

4.2: Classifying the Directories and Files

After establishing the installation directory structure for the vendor application, it is necessary to classify the content of each. Frequently, the directories are self-explanatory, but often the vendor utilizes terms that are not industry standard and require additional explanation. The application directory structure for the sample scenario is detailed in the table below.

Tier	Directory	Description
Presentation	images	Image directory for all gif and jpg files used on the web-based presentation tier.
Presentation	log	Log file directory for web server and application transactions.
Presentation	cgi	Common gateway interface scripts used by the presentation-tier to process transactions sent to the application-tier.
Presentation	htdocs	Web content directory for holding html and xml files
Presentation	java	Directory that holds java and JavaScript code.
Application	bin	Compiled binary files that make-up the core application.
Application	log	Log file directory with flat-text record based user access and database request transactions.
Application	cgi	Common gateway interface scripts used by the application-tier to receive and process transactions sent from the presentation-tier.
Application	utils	Directory that holds various utilities for database and application maintenance.
Application	data	Security data and metadata directory

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

4.3: File Interrogation

Using the information established in the previous section, it is appropriate to determine the general data contained in each area of the file system and evaluate the controls. It is important to start with a general "sweep" of the installation structure and then concentrate on areas of the file system for additional focus based upon risk.

The general "sweep" of the installation structure will be performed using the "find" command using techniques proposed in the Wreski article titled, "Monitoring Files with Special Permissions." As quoted in the Wreski article, "setuid and setgid files on your system are a potential security risk, and should be monitored closely. Because these programs grant special privileges to the user who is executing them, it is necessary to ensure that insecure programs are not installed." To illustrate the process documented in the Wreski article, the sample scenario will be utilized. Starting at the root of the application (/erpapp), we will execute the commands as listed below with output. Note: Some of the columns of output generated by the "-ls" option (which usually generates output formatted as if the command 'ls –dlis' we executed on the files returned) were removed to allow space and help simplify the listing.

```
 app01 / $  find . -type f -perm +6000 -ls
-rwsrwxrwx   1 root   users      3220410 Feb 23 2001  ./bin/vppServer
-rwsrwxrwx   1 root   users      851471  Feb 23 2001  ./bin/otherServer
```

The command with output above shows that two binaries set with the SUID permission. Additionally, the binaries allow anyone to execute or change them. This permission set implemented on these files is not secure especially since one creates a listening service. The function of vppServer has been established. However, the otherServer binary should be reviewed with the vendor to analyze the function and potential risk.

The next general "sweep" to conduct is to find world-writable files. As noted in the article above, "World-writable files, particularly system files, can be a security hole if a cracker gains access to your system and modifies them. Additionally, world-writable directories are dangerous, since they allow a cracker to add or delete files as he wishes." To illustrate this process, the following commands are issued from the root of the example application.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
app01 / $  find / -perm -2 ! -type l -ls
drwxrwxrwx  2 root   users      4096 Feb 23 2001  ./bin
-rwsrwxrwx   1 root   users      3220410 Feb 23 2001 ./bin/vppServer
-rwsrwxrwx   1 root   users      851471  Feb 23 2001  ./bin/otherServer
drwxrwxrwx   2 erp1     users       4096 Feb 23 2001 ./log
drwxrwxrwx   2 erp1     users       4096 Feb 23 2001 ./cgi
-rwxrwxrwx   1 erp1     users       6981 Feb 23 2001 ./cgi/cgi3.cgi
-rwxrwxrwx   1 erp1     users      15674 Feb 23 2001 ./cgi/cgi2.cgi
-rwxrwxrwx   1 erp1     users       5621 Feb 23 2001 ./cgi/cgi1.cgi
drwxrwxrwx   2 erp1     users       4096 Feb 23 2001 ./utils
drwxrwxrwx   2 erp1     users       4096 Feb 23 2001 ./data
-rwxrwxrwx   1 erp1     users       5238 Feb 23 2001 ./data/data3.dat
-rwxrwxrwx   1 erp1     users        342 Feb 23 2001 ./data/data1.i
-rwxrwxrwx   1 erp1     users        414 Feb 23 2001 ./data/data2.i
-rwxrwxrwx   1 erp1     users        198 Feb 23 2001 ./data/data3.i
-rwxrwxrwx   1 erp1     users       9180 Feb 23 2001 ./data/data1.dat
-rwxrwxrwx   1 erp1     users       6624 Feb 23 2001 ./data/data2.dat
```

The output generated shows a significant number of important files that are writable by anyone on the system. All of the files shown in this output should be reviewed with the vendor to determine the content and risk associated with each file.

Note: Some of the columns of output generated by the "-ls" option (which usually generates output formatted as if the command 'ls –dlis' were executed on the files returned) were removed to allow space and help simplify the listing. Additionally, the third test performed in the Wreski article is not illustrated in this paper because no output was returned.

Using the data collected and presented in section 4.1, concern should be shown regarding the open file permissions on very important files such as logs, security data files, binaries, and cgi files. The information noted should be communicated in the analysis for this section.

Permissions & Files	Directory	Description
-rwsrwxrwx ./bin/vppServer -rwsrwxrwx ./bin/otherServer	bin drwxrwxrwx	Compiled binary files that comprise the core application.
None generated, but the permission will be set to 777 by default.	log drwxrwxrwx	Log file directory for user accesses and database request transactions.
-rwxrwxrwx ./cgi/cgi3.cgi -rwxrwxrwx ./cgi/cgi2.cgi -rwxrwxrwx ./cgi/cgi1.cgi	cgi drwxrwxrwx	Common gateway interface scripts used by the application-tier to receive and process transactions sent from the presentation-tier.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Permissions & Files	Directory	Description
-rwxrwxrwx ./data/data3.dat -rwxrwxrwx ./data/data1.i -rwxrwxrwx ./data/data2.i -rwxrwxrwx ./data/data3.i -rwxrwxrwx ./data/data1.dat -rwxrwxrwx ./data/data2.dat	data drwxrwxrwx	Security data and metadata directory

4.4: Quick Analysis

Governance Document Analysis and Review Program		
Policy, Procedure, Standard	*Functional Requirement*	*Technical Implementation*
Security Policy 2 Server Administration Standard	File, directory, and resource permissions • *Permissions should be set to the least access that still enables the system to function appropriately.* • *Includes mounted file systems.* • *Files that are set with permissions that escalate user privileges (i.e. suid, sgid, etc.) or contain access control settings should be documented, approved, monitored, and controlled as possible.* • *Files that contain information which is critical to "after the fact" investigations should be secured and managed to provide integrity*.	File permissions on critical files (i.e. application, security data, and log files) are not protected and are subject to change by any system user. Two files are executable and writable by any system user and enable privilege escalation. Additionally, multiple data files that control security are writable by any system user. Application log files, when created, will allow all users the "write" privilege and therefore are subject to modification or deletion by any user of the system. Failed

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Part Five: Authentication

5.1: User Authentication

User identification and authentication are key components of any system modern business system. "Identification and authentication are the processes of recognizing and verifying valid users or processes" (Krause and Tipton 89). Authentication is the most visible area of security for the end-user and often the most time consuming to administer depending on the number of points of user administration and the control mechanisms given.

Using the information gathered in part two, the fact was noted that a user can authenticate to both the application tier and the web tier. The best place to begin the analysis of the authentication system used is to ask the vendor: In the sample scenario, the following information was gathered from the questions presented to the vendor and further analysis of the application:

How does the user authenticate? (Question to the vendor)

> The user will authenticate on the web-tier using a standard web browser over http (or optionally installed https) using the basic authentication function built into the IBM® HTTP Server. On the application tier, the user will authenticate using the vendor supplied win32 client to the standard UNIX login function. On the data-tier, one generic user ID and password will be used globally. The application does not have the functionality built in to integrate with any centralized user authentication system. (Vendor Response)

Where is the user authentication data stored on the system? (Question to the vendor)

> The user authentication data is stored in the "htpasswd" file on the web-tier, the "passwd" file on the application tier, and the dbauth file stored on the application tier used to authenticate using the Resource Access Control Facility (RACF) on the mainframe. (Vendor Response)

Based upon the information provided by the vendor and additional research, it appears there will be two points of user administration, one on the web-tier and one on the application tier. Additionally, one generic user ID and password will be utilized for DB2 database access. Therefore, the next step is to gather additional detail concerning tier-based authentication.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

5.2: Tier Authentication

In the tier authentication, the goal is to determine how the user actually authenticates on each tier and where the authentication information is stored and maintained. According to the vendor, the application cannot integrate with any centralized user store. Therefore, one must depend on the authentication systems built into the web, application, and data tiers.

The next step is to determine if the application provides any legal or warning banner facilities and to obtain additional detail on the tier-authentication systems. The vendor is the best place to start for questions concerning the banner facility. The O/S and the web server documentation can be consulted for the authentication detail. Therefore, a demonstration of the process will be completed using the sample scenario and asking the vendor the about the banner facility available in the application.

Are facilities available for displaying company standard banners or special information to the user during the authentication process?

> The application does not provide any specific facilities for displaying special banners during the authentication process. The application depends on the facilities supplied by the web server application on the web-tier (i.e. IIS, HIS, Apache, etc.) and the O/S on the application tier.

Based upon the answer from the vendor, it appears the O/S and web server application will need to provide this functionality. After consulting the documentation on the O/S and the web server application the following information was noted.

Functional Requirement	O/S or Web Server Feature
User Authentication	The IBM® AIX operating system System Management Interface Tool (SMIT) for user management and the IHS authentication system using htpasswd.
Warning Banner Facility	Application Tier - A warning banner can be documented in the “motd” file on the Unix operating system and most login services can be configured to use this file for the banner displayed during login. Web Tier – The warning banner can be displayed on the IHS login page.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

The next step is to determine the control features that can be utilized to manage users within each of the tiers.

5.3: Managing/Controlling Authentication

The control and management of user ID's and passwords is critical in any online business application. The specific controls provided by each authentication system will vary from virtually none to every control imaginable. Therefore, it is important for the analyst to understand the standards required by the company in order to perform a thorough analysis. To demonstrate the process of mapping the functional requirement to the authentication system, the sample scenario will be utilized. Due to the specific requirements noted via the standards, we will reference this analysis directly in the quick analysis section.

Functional Requirement	Web Tier	Application Tier	Database Tier
Passwords should expire every 90 days	No facility is provided by IHS to automatically expire passwords after 90 days.	The SMIT utility provides the functionality to automatically expire the password as needed.	The RACF utility provides the functionality to automatically expire the password as needed.
The user will be notified of the password's coming expiration and be granted up to four (4) grace logins prior to the old password's expiration.	No facility is provided by IHS to enable this functionality.	The SMIT utility provides the functionality to provide this feature.	The RACF utility provides the functionality to provide this feature.
Users should be restricted to a minimum number of weeks before a password can be changed.	No facility is provided by IHS to enable this functionality.	The SMIT utility provides the functionality to provide this feature.	The RACF utility provides the functionality to provide this feature.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Functional Requirement	Web Tier	Application Tier	Database Tier
Users should be restricted to a maximum number of weeks beyond expiration that an expired password can be changed by the user.	No facility is provided by IHS to enable this functionality.	The SMIT utility provides the functionality to provide this feature.	The RACF utility provides the functionality to provide this feature.
The user should not be able to reuse the same password from the past 12 months.	No facility is provided by IHS to enable this functionality.	The SMIT utility provides the functionality to provide this feature.	The RACF utility provides the functionality to provide this feature.
Passwords should be a minimum of seven characters for regular users and eight characters for administrators and service type accounts.	No facility is provided by IHS to enable this functionality.	The SMIT utility provides the functionality to provide this feature.	The RACF utility provides the functionality to provide this feature.
Access should be denied after three incorrect login attempts.	No facility is provided by IHS to enable this functionality.	The SMIT utility provides the functionality to provide this feature.	The RACF utility provides the functionality to provide this feature.
The user account must be disabled after termination, transfer, or forty-five days of inactivity.	No facility is provided by IHS to enable this functionality.	No specific feature is provided by SMIT, but this process can be scripted using O/S log files and command-line SMIT utilities.	N/A

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Functional Requirement	Web Tier	Application Tier	Database Tier
The user account must be deleted after 90 days in a disabled state.	No facility is provided by IHS to enable this functionality.	No specific feature is provided by SMIT, but this process can be scripted using O/S log files and command-line SMIT utilities.	N/A
All user-chosen passwords must contain at least one alphabetic, one numeric and one special character.	No facility is provided by IHS to enable this functionality.	The SMIT utility provides the functionality to provide this feature.	N/A

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

5.4: Quick Analysis

Governance Document Analysis and Review Program		
Policy, Procedure, Standard	*Functional Requirement*	*Technical Implementation*
Security Policy 2 Legal Standard Server Administration Standard	Warning Banner • *Must be displayed at every interactive user login where identification and authentication is needed to access information.*	The requirement is implemented using the O/S on the application tier and IHS on the Web-tier. <u>Passed</u>
Security Policy 1 System Account Standard	Password Characteristics • *Passwords should expire every 90 days* • *The user will be notified of the password's coming expiration and be granted up to four (4) grace logins prior to the old password's expiration.* • *Users should be restricted to a minimum number of weeks before a password can be changed.* • *Users should be restricted to a maximum number of weeks beyond expiration that an expired password can be changed by the user.* • *The user should not be able to reuse the same password from the past 12 months.*	See detailed analysis on previous two pages. <u>Failed</u>

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Policy, Procedure, Standard	*Functional Requirement*	*Technical Implementation*
	• *Passwords should be a minimum of seven characters for regular users and eight characters for administrators and service type accounts.*	
Security Policy 1 System Account Standard	User ID and Password Management • *Access should be denied after three incorrect login attempts.* • *The user account must be disabled after termination, transfer, or forty-five days of inactivity.* • *The user account must be deleted after 90 days in a disabled state.* • *All user-chosen passwords must contain at least one alphabetic, one numeric and one special character.*	See detailed analysis on previous two pages. <u>Failed</u>

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Part Six: Authorization

6.1: Authorization Decision Locations

Authorization and authentication are two distinct functions. However, it is important to recognize that implementing authorization for each user is only possible if each user can be identified. If the user is not individually authenticated during the process, then user-based authorization cannot happen. The key point for this section is to determine how the application stores and uses the access control list (ACL) for each user.

In the sample scenario, security related data was stored in the "data" directory as detailed in section 4.1. Additionally, it was noted in section 5.1 that the application did not perform the authentication function. Therefore, we gathered that the "security data" stored in the "data" directory was actually authorization data for each user. The confirmation of this assessment was obtained in discussions with the vendor voicing our concerns about the open file permissions by location. The vendor is the best information source to obtain detail about authorization information and how it operates. It is important to use the vendor for information at every opportunity.

6.2: Granularity, Consistency, and Simplicity

The purpose of this section is to determine the features implemented in the application to grant and revoke user access to application resources such as screens, data, reports, jobs, metadata, parameters, etc. These user access control facilities vary widely by application vendor. It is impossible to cover every aspect of this often gray area, but three questions should be kept in mind as you analyze this area:

How granular can user access to resources be defined?

In the sample scenario, the user access control can be very granular down to the resource and the action permitted to be taken on each area (i.e. inquire, add, delete, etc.). However, the access control model the vendor implemented was a subtractive security model and not and additive security model. Therefore, each user of the system was granted full access to all system resources by default and each function the user was not intended to access must be removed. An additive access control model would have given the user no access by default and each function the user was intended to access would be added to the access control list. The vendor did point out that a report could be generated showing the access permissions of each user.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

How consistent and effective are the security features?

In our sample scenario, the security application appeared to be very consistent. The vendor explained how the security system worked via user groups and the users were added to a group to obtain access to resources. However, the component the vendor neglected to explain was access to the system through the various system "utilities" and other "non-standard" methods was not controlled by the security application and must be maintained via a separate model.

How simple is the security model?

The easier and more straightforward a security model is implemented the better. The implementation of a complicated security model usually will result in security holes due to incorrect configuration or non-use. Mark Curphey writes that "often the most effective security is the simplest security" and "if the steps to properly secure a function or module of the application are too complex, the odds that the steps will not be properly followed increase greatly."

6.3: Dependence on Trust – Know the Gaps

It is important to note that in a multi-tier application there are many different methods to implement a trust model between the tiers. In our sample scenario, we noted that the vendor was using VPP01 to pass commands from the web tier through the application tier that formulated a query to the data tier. However, the application tier did not require specific user authentication. The application tier received the queries from the web tier and the trusted that the web tier authenticated the user. Furthermore, all user transactions were passed to the database using a single user ID and password.

It was noted during testing that only user ID information was passed from the web tier to the application tier in the query. The application tier accepted the user ID in the query as valid, completed the normal authorization process, and passed the results back to the web tier. However, there were no application level controls to ensure that the user ID passed by the web server via the query was valid. Therefore, another device could pass a similar query to the application tier and receive data without authentication. Understanding the application in total is critical to determining the trust relationships inherent within the architecture. Furthermore, using a consistent holistic approach and utilizing the vendor will help make it obvious where security gaps exist in the application and where compensating controls are needed.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

6.4: Quick Analysis

Governance Document Analysis and Review Program		
Policy, Procedure, Standard	*Functional Requirement*	*Technical Implementation*
Security Policy 2 User Access Standard	Data Access Controls • *Users will be granted access to the minimum resources needed to complete assigned duties.* • *Business Owners must approve all user access to data resources* • *Automated processes/tools must be in place to produce the access permissions each user has on the system.* • *Processes/procedures must be in place to compare the access approved by the business owners to the actual settings.*	This requirement can be met with application security; however, additional controls may need to be implemented to fill the gaps created by application architecture. The application can produce a report that shows the access permissions of each user. This requirement cannot be provide by the application. Additional tools can be built to provide this functionality. <u>Passed</u>

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Part Seven: Auditing

7.1: Log Locations and Formats

In section 4.2 it was noted that the application logs were maintained in the log directory and kept in flat text record format. Other non application logs (i.e. O/S logs) will be important in providing all the auditing requirements, but for this analysis focus will be applied to the logs generated by the application.

Application audit logs are important because they are the primary source for review to determine the transactions the user is performing inside the application. Therefore, maintaining good application audit logs is critical to troubleshooting, tracing user activities, and evaluating how a specific outcome (i.e. generation of a duplicate payroll check) was generated,

7.2: Logging Configurations, Settings, and Content

An application logging facility is only as good as its consistency and flexibility. Consistency is important because the business should have faith that the logging system will track activities as configured. Flexibility is important because logs can quickly become overbearing in size and scope. The logging facility should be flexible enough to enable the business to detail log the important transactions while minimizing the logging on transactions not deemed important.

The logging facility of the application in the sample scenario was consistent for most user access, but was not flexible. The facility provided the feature to turn logging on or off, but would only log failed user attempts. Additionally, the facility did not log "utility" based access as described in section 6.2. Therefore, the business could not obtain logs of all user activities because of the architecture. The information obtained in section 6.3 shows the web or application tiers are the only places where user level logs could be obtained because a generic user ID and password are used to access the database and "utility" based access is not logged.

7.3: Log Protection

A log is only as dependable as the consistency of the logging facility and the integrity controls placed on the log output. In our sample scenario, we noted in section 4.3 noted the log file default permissions were very open and permitted any user of the system to edit or delete the logs. Additionally, in section 7.2 it was noted that the logging facility was consistent but not flexible. Therefore, the logging system is not well protected due to the file permissions provided by the vendor.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

7.4: Quick Analysis

Governance Document Analysis and Review Program		
Policy, Procedure, Standard	*Functional Requirement*	*Technical Implementation*
Security Policy 2 Audit Trail Standard	Audit Logs • *Must capture and contain information needed to determine possible fraudulent use of the system. Activities that must be recorded:* ○ *unsuccessful logon attempts* ○ *Successful and unsuccessful resource access attempts* ○ *Use of special utilities that allow non-standard transactions.*	This functional requirement cannot be completely met due to the logging facility not recording successful access attempts and "utility" based access.
	• *Minimum audit trail elements:* ○ *date and time of the event* ○ *user ID* ○ *type of event* ○ *success or failure of event* ○ *source of event* ○ *before and after data state (if applicable)*	The logging facility cannot provide this functionality because successful and "utility" based attempts are not logged.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Part Eight: Overall Analysis

The primary purpose of this section is to provide a scoring of each section based on the function requirements established through policy, procedure, and standard. The format of this analysis will be in report card format with each functional requirement assigned one point. The analysis is below.

Application Security Analysis Report (Based upon Great Company Policy, Procedure, and Standard)		
Part. Title	*Total High-Level Functional Requirements*	*Functional Requirements Met*
2. Application Architecture	*0*	*0*
3. Network Communications	*3*	*0*
4. File and Directory Review	*1*	*0*
5. Authentication	*3*	*1*
6. Authorization	*1*	*1*
7. Auditing	*1*	*0*
Total Score (22.2%)	*9*	*2*

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Summary

The process of assessing vendor application security can be very complex, but the information obtained may be critical in making a good purchasing decision. It is important to be thorough in the assessment, but the key is to define the review criteria and goals before beginning the process. Additionally, utilizing policies, procedures, and standards as the guide will help ensure that the results are applicable.

The assessment program should be organized into logical parts and summarized accordingly. This format will allow the analyst to present the overall findings at a high-level, but enable reference back to the detail if necessary. The overall goal of the assessment process is not to find every obscure security issue within the application, but to help ensure that the security of the system is in-line with the business requirements.

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Bibliography

Abell, Vic. Vic Abell's Home Page. 16 February 2004 <http://www-rcd.cc.purdue.edu/~abe/>

Aspinall, Jeff. Common UNIX Commands. 16 February 2004 <http://web.nwe.ufl.edu/writing/tools/unix/commands.html>

Chartier, Robert. "Application Architecture: An N-Tier Approach - Part 1." 15seconds.com. 23 October 2001 <http://www.15seconds.com/issue/011023.htm>.

Curphey, Mark, et al. A Guide to Building Secure Web Applications. 22 September 2002. The Open Web Application Security Project. <http://prdownloads.sourceforge.net/owasp/OWASPGuideV1.1.1.pdf?download>

Enders, Matthew, and Steve Hayes. TCP/IP Tutorial and Technical Overview. June 1995, <http://www.auggy.mlnet.com/ibm/3376c210.html#sokapi>.

Howard, Michael, and David LeBlanc. Writing Secure Code. Redmond, WA: Microsoft Press, 2002.

International Business Machines (IBM). DB2 Connect. 1 February 2004 <http://www-306.ibm.com/software/data/db2/db2connect/>

Krause, Micki, and Harold F. Tipton. Handbook of Information Security Management 1999. Boca Raton, FL: Auerbach, 1999.

Wreski, Dave. "Monitoring Files with Special Permissions." Linuxsecurity.com. 17 July 2000 <http://www.linuxsecurity.com/tips/tip-4.html>.

"win32, “javascript”, “common gateway interface”, “common business oriented language”, “c”, “java”, “perl”, “Indexed Sequential Access Method”, "inetd”, “Resource Access Control Facility.” Hyperdictionary.com. Online. Internet. 1 February 2004. Available HTTP: www.hyperdictionary.com/computer/

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Appendix A

Governance Document Analysis and Review Program		
Policy, Procedure, Standard	*Functional Requirement*	*Technical Implementation*
Security Policy 2 Encryption Standard	Encryption for confidential network communications • *112-bit key length for symmetric encryption (such as Triple 56 bit DES);* • *2048-bit key length for asymmetric encryption; and* • *160 bits for Elliptical Curve systems.*	
Security Policy 1 Encryption Standard	Password Transmission • *Passwords may be sent over the network only when encrypted or hashed.*	
Security Policy 1 Server Administration Standard	Services • *Services, which are not required for the role of the server, must be disabled.* • *All "listening" services must be documented and patched regularly.* • *Appropriate care should be taken to limit source connections as possible.* • *All "listening" services should be able to withstand a routine port scan.*	

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Policy, Procedure, Standard	*Functional Requirement*	*Technical Implementation*
Security Policy 2 Server Administration Standard	File, directory, and resource permissions • *Permissions should be set to the least access that still enables the system to function appropriately.* • *Includes mounted file systems.* • *Files that are set with permissions that escalate user privileges (i.e. suid, sgid, etc.) should be documented, approved, and monitored.* • *Files that contain information which is critical to "after the fact" investigations should be secured and managed to provide integrity*.	
Security Policy 2 Legal Standard Server Administration Standard	Warning Banner • *Must be displayed at every interactive user login where identification and authentication is needed to access information.*	

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Policy, Procedure, Standard	*Functional Requirement*	*Technical Implementation*
Security Policy 1 System Account Standard	Password Characteristics • *Passwords should expire every 90 days* • *The user will be notified of the password's coming expiration and be granted up to four (4) grace logins prior to the old password's expiration.* • *Users should be restricted to a minimum number of weeks before a password can be changed.* • *Users should be restricted to a maximum number of weeks beyond expiration that an expired password can be changed by the user* • *The user should not be able to reuse the same password from the past 12 months.* • *Passwords should be a minimum of seven characters for regular users and eight characters for administrators and service type accounts.*	

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Policy, Procedure, Standard	*Functional Requirement*	*Technical Implementation*
Security Policy 1 System Account Standard	User ID and Password Management • *Access should be denied after three incorrect login attempts.* • *The user account must be disabled after termination, transfer, or forty-five days of inactivity.* • *The user account must be deleted after 90 days in a disabled state.* • *All user-chosen passwords must contain at least one alphabetic, one numeric and one special character.*	
Security Policy 2 User Access Standard	Data Access Controls • *Users will be granted access to the minimum resources needed to complete assigned duties.* • *Business Owners must approve all user access to data resources* • *Automated processes/tools must be in place to produce the access permissions each user has on the system.*	

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Policy, Procedure, Standard	*Functional Requirement*	*Technical Implementation*
	• *Processes/procedures must be in place to compare the access approved by the business owners to the actual settings.*	

Security Policy 2
Audit Trail Standard
	Audit Logs
	• *Must capture and contain information needed to determine possible fraudulent use of the system. Activities that must be recorded:*
	  ○ *unsuccessful logon attempts*
	  ○ *Successful and unsuccessful resource access attempts*
	  ○ *Use of special utilities that allow non-standard transactions.*
	• *Minimum audit trail elements:*
	  ○ *date and time of the event*
	  ○ *user ID*
	  ○ *type of event*
	  ○ *success or failure of event*
	  ○ *source of event*
	  ○ *before and after data state (if applicable)*	

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Last Updated: September 3rd, 2015

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS Crystal City 2015	Crystal City, VAUS	Sep 08, 2015 - Sep 13, 2015	Live Event
SANS Network Security 2015	Las Vegas, NVUS	Sep 12, 2015 - Sep 21, 2015	Live Event
SANS Seoul 2015	Seoul, KR	Sep 14, 2015 - Sep 19, 2015	Live Event
Data Breach Investigation Summit & Training	Dallas, TXUS	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS Baltimore 2015	Baltimore, MDUS	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS Perth 2015	Perth, AU	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS Tallinn 2015	Tallinn, EE	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS ICS Amsterdam 2015	Amsterdam, NL	Sep 22, 2015 - Sep 28, 2015	Live Event
SANS Bangalore 2015	Bangalore, IN	Sep 28, 2015 - Oct 17, 2015	Live Event
SANS DFIR Prague 2015	Prague, CZ	Oct 05, 2015 - Oct 17, 2015	Live Event
SANS Seattle 2015	Seattle, WAUS	Oct 05, 2015 - Oct 10, 2015	Live Event
SANS Tysons Corner 2015	Tysons Corner, VAUS	Oct 12, 2015 - Oct 17, 2015	Live Event
SOS: SANS October Singapore 2015	Singapore, SG	Oct 12, 2015 - Oct 24, 2015	Live Event
SANS Gulf Region 2015	Dubai, AE	Oct 17, 2015 - Oct 29, 2015	Live Event
SANS Cyber Defense San Diego 2015	San Diego, CAUS	Oct 19, 2015 - Oct 24, 2015	Live Event
SANS Tokyo Autumn 2015	Tokyo, JP	Oct 19, 2015 - Oct 31, 2015	Live Event
SANS South Florida 2015	Fort Lauderdale, FLUS	Nov 09, 2015 - Nov 14, 2015	Live Event
SANS Sydney 2015	Sydney, AU	Nov 09, 2015 - Nov 21, 2015	Live Event
SANS London 2015	London, GB	Nov 14, 2015 - Nov 23, 2015	Live Event
Pen Test Hackfest Summit & Training	Alexandria, VAUS	Nov 16, 2015 - Nov 23, 2015	Live Event
SANS Hyderabad 2015	Hyderabad, IN	Nov 24, 2015 - Dec 04, 2015	Live Event
SANS Cape Town 2015	Cape Town, ZA	Nov 30, 2015 - Dec 05, 2015	Live Event
SANS San Francisco 2015	San Francisco, CAUS	Nov 30, 2015 - Dec 05, 2015	Live Event
SANS Milan 2015	OnlineIT	Sep 07, 2015 - Sep 12, 2015	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced