Global Information Assurance Certification Paper

Hardening Oracle in a Linux (Unix) Environment
You have to make it unbreakable!

SANS GIAC Security Essentials Certification
(GSEC) Practical
V.1.4c Option 1

Robert W. Persick
November 22, 2004

Abstract

Worms, and hackers, and spyware… Oh my! Not to mention viruses, spam, ID theft, denial of service attacks and random security flaws. You would think that at 35 years old the Internet would be a more responsible place to do business. Just like real life, it's not and next to people the next most valuable asset a company has is usually it's data. Are you concerned with physical data security? It should come as no surprise to anyone that a rental car was stolen in St. Louis on 26 Feb. 2004, one of the most dangerous places to live in the U.S. But in the trunk of that stolen rental car was a laptop that maintained account and private information of thousands of Wells Fargo mortgage customers. [1 Thompson] Viruses and worm outages, caused by the likes of Code Red, Melissa and Blaster cost companies billions in down time, system restoration, data corruption and good will. These viruses and worms, like SQL Slammer, cost consumers an estimated $1.2 billion in lost productivity in its first five days alone. [1 Thompson] Last year (2003), an FBI survey of mid to large companies indicated that the average cost to be $1.4 million for security breaches and denial of services attacks from external hackers.

Data represents one of the most valuable assets of an organization; some may say the most valuable. Most organizations store their valuable data in databases. [2 Husain] This tutorial attempts to highlight the steps of procedure and security considerations in securing that database, specifically on a Red Hat Linux Operating System (OS) using an Oracle 9i database since this is an Enterprise Class product with almost a "hacker challenge" tied to it's marketing. Thirty-four vulnerabilities -- the majority of them critical -- have been identified in multiple versions (10g, 9i and 8i) of Oracle's database server. [3 McAlearney] These vulnerabilities released during the Las Vegas Black Hat Briefings highlight the need to understand that if Oracle is going to be unbreakable, the I.T. department is going to have to make it that way with vendor support. This is accomplished through an in-depth defense posture, three-tiered architecture, OS and application minimalization, regular patch installation and user account administration. While I will mention layers of security and architecture considerations, these issues are generally better understood and there appears to be a fair amount of literature surrounding these topics. Since my experience is mostly with Unix (Solaris flavor) and due to the sheer volume of information available, while framing the topic to Red Hat Linux and Oracle, I am going to constrain these steps further and focus on OS and application minimalization, regular patch installation and user account administration.

Layers of Defense

Oracle installations should have the highest level of security applied to them because they contain your most valuable non-human asset. Defense In-depth has to do with a layered approach to security. This is always advisable. So in addition to your Virtual Private Network (VPN) between your router and service provider, Access Control Lists (ACL) on the switches, a Firewall blocking your Demilitarized Zone (DMZ), your web-server only having port 80 open to your application server on the other side of another firewall within a VLAN set to the other side of another Firewall to your database server that is only open on port 1521 to the application server with each of these devices having minimal Operating Systems (OS) on them you still have to consider user access and Privileged user escalation.

Access-control lists in routers and switches can be a great complement to your firewalls. At the perimeter, ACLs can filter out unwanted traffic before it's processed by your intrusion-detection systems and firewalls, thereby dramatically cutting down the size of your logs, as well as the amount of labor and storage needed to manage them. ACLs are a series of rules, similar to firewall rules, which define a pattern match for a packet and associate an action with the packet. [4 Morrissey] Like Firewall Rule sets they are best set to deny everything except that which you specially permit.

Virtual LANs (VLANs), and VPNs are ways of reconciling where someone is located and what they're system is called. A network user has an address by which that user is known and a place where that user is connected. For traffic to reach the user, network devices called nodes (which include routers and switches, bridges are rarely used any more) have to convert the user's address to the user's location. This is called routing. If every user had a single, consistent address, and if all addresses were organized logically, routing would be easy. Unfortunately, this is not the case.

Organizing addresses means being sure that all users in a given location have addresses in a common range (an IP subnet, for example), so that a single routing instruction can send all users their packets. If addresses are assigned without regard for location, routing tables must steer each user's packets individually, which requires a routing entry for every user. That's impractical: the size of the routing tables would grow from today's 50,000 or so entries to tens of millions of entries. [5 Nolle]

Other problems arise when a user changes locations (moving his or her office, for example). The network has to learn of this, or the address that represents the user would continue to be routed to the old site. Giving the user a new address (one that fits in the range of addresses being routed to the new location) means changing how that user is known through the network, which may disconnect the

user from previous applications and partners. Letting the user have the same address in the new location means taking a step down the ugly road toward routing every user individually. [5 Nolle]

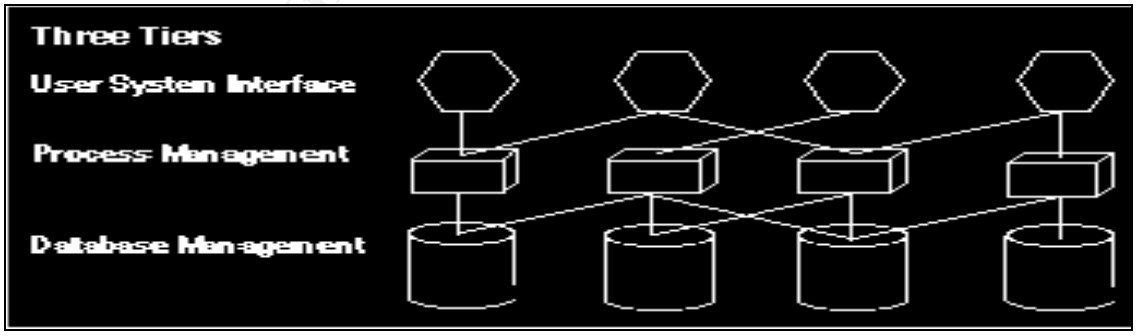
LAN switch vendors came up with the virtual LAN to alleviate the moving mess. Every user has a level-2 address (the MAC address of the user's NIC). A VLAN makes a user's MAC address appear to stay where it is when the user moves, and all the LAN software (including TCP/IP software) never even knows the user has moved. [5 Nolle]

Though we are not going to address mobile users this is good to know. Wow, this is a lot and I can't afford that many Firewall's you say… of course you can if you are using Red Hat with IPtables or IPchains each server can be it's own Firewall or with Solaris you can use ipfilter [6 Ciolek]. These are stateful packet filters. So we should use what we have and make every server a Firewall.

Three-Tier Systems

The World-wide Web (Globally) or the Internet (USA) has dramatically increased the need for three-tiered systems. The three tier architecture is used when an effective distributed client/server design is needed that provides (when compared to the two tier) increased performance, flexibility, maintainability, reusability, and scalability, while hiding the complexity of distributed processing from the user. These characteristics have made three layer architectures a popular choice for Internet applications and net-centric information systems. [7 Sadoski]

A three-tier distributed, client/server architecture (as shown) includes a user system interface top tier (also known as the web tier or portal tier) where user services (such as session, text input, dialog, and display management) reside. Make sure to test your Web Applications for "SQL Injections" and "Cross-sight Scripting."



Three tier distributed client/server architecture depiction [7 Sadoski]

The middle tier provides process management services that are shared by multiple applications. The middle tier server (also referred to as the application server or tier) improves performance, flexibility, maintainability, reusability, and

scalability by centralizing process logic. Centralized process logic makes administration and change management easier by localizing system functionality so that changes must only be written once and placed on the middle tier server to be available throughout the systems. With other architectural designs, a change to a function (service) would need to be written into every application. In addition, the application tier controls transactions and asynchronous queuing to ensure reliable completion of transactions to the database. The application tier manages distributed database integrity by the two-phase commit process. It provides access to resources based on names instead of locations, and thereby improves scalability and flexibility as system components are added or changed.

The third tier provides database management functionality and is dedicated to data and file services that can be optimized without using any proprietary database management system languages. Though we are primarily concerned with Oracle. The data management component ensures that the data is consistent throughout the distributed environment through the use of features such as data locking, consistency, and replication. It should be noted that connectivity between tiers can be dynamically changed depending upon the user's request for data and services. [7 Sadoski]

The primary reasons for deploying a three-tier system are: efficient resource management, improved scalability, and security. In a three-tier system, the middle tier can act as a concentrator, allowing many user devices to share a relatively few connections to the back-end system. Moreover, in a three-tier system the middle tier can focus on presentation of data to the user, allowing the back end tier to focus on management and processing of data. Since databases are optimized for efficient data management, moving responsibility for data presentation and connection management to the middle tier can improve system efficiency and scalability. Moreover, application logic in the middle tier can limit access of users, and provide another layer of isolation to sensitive data maintained in database. This improves system security. [8 Heimann]

Installation Overview

While Firewalls are one of the fastest growing technical tools in the field of information security. However, a firewall or any server is only as secure as the operating system it resides upon. This guide will take a step-by-step look at how you can best install the Linux OS on an Intel server to comply with Information Assurance Department (IA) requirements. These steps apply to most situations, however I will be using Red Hat 9, and we will continue with the installation of Oracle in a secure environment. The best place to start in tightening your system is at the beginning, OS installation. You cannot trust any previous installations. You want to start with a clean installation, where you can guarantee the system integrity.

While completing this practical I used a laptop, but have for previous and current employers I used everything from a Sun e250 to Sunfire 15K with Solaris and DL360 and Dell 2650 for Red Hat; typical installations could include:

	Sun Configuration	x86 Configuration
System Name	SunFire 280R	DL360
Manufacturer	Sun	Compaq
List Price	$17,995	$5,943
CPU type	900Mhz UltraSparc III	1000Mhz Pentium III
CPU count	2	2
RAM	2GB	2GB
Local Disk	2 @ 36GB	2 @ 36GB
System Size	4U	1U

Hardware Configurations [9}

First place your system in an isolated network or create it as a standalone and reconfigure the Network Interface Card (nic) and network components later. At no time do you want to connect your unprotected system on an active network or the Internet, exposing the system to a possible compromise. To get critical files and patches later, you will need a second box that acts as a go between. This second box will download files from the Internet, then connect to your isolated, configuration "network" to transfer critical files or be used to build the packages required to complete the installation. Or burn them to cdrom to transfer. This system could be in a lab or one of your development boxes.

Once you have placed your server in an isolated network, you are ready to begin. The first step is selecting what OS package to load. The idea is to load the minimum installation, while maintaining maximum efficiency. The less software that resides on the box, the fewer potential security exploits or holes that may exist.

So much depends on the system disk that it is worth keeping the disk only for system use. Swapping it then becomes possible without effecting users. Performing upgrades, cloning, and mirroring are all easier as well. Performance improves if the system disk is not used for other purposes as well. If this is a multi-disk system you should have a separate system disk.

During the installation process, you will be asked to partition your system. I always like to make root as big as possible and just throw everything in there, and then you do not run out of room. However, you do need several partitions to protect the root drive. If we were to fill the root partition with data, such as logging or email, we would cause a denial of service, potentially crashing the system.

Also it is an Oracle best practice to spread out the database components. So let's get started. You do have your media right…

1) Red Hat Linux Boot screen:

After configuring the system for booting from a CD, the Red Hat Linux Boot screen appears. At this point, press "Enter" for the graphical Setup.

2) Welcome to Red Hat:

The "Welcome to Red Hat" screen appears with the option of Hiding the help pane (left side) or viewing the Release Notes. Select **"**Next" when ready.

3) Language Selection:

The Language Selection screen displays all of the languages available to install Red Hat with. I chose English here, then **"**Next**"**.

4) Keyboard Configuration:

Keyboard Configuration is "Next". Highlight the best match for your particular system. Usually, the default works best. This may take some testing if you are not sure and don't know exact compatibility with drivers based on your system.

Select **"**Next**"** when complete.

5) Mouse Configuration:

Mouse Configuration is "Next". Again, highlight the best match for your particular system. This may take some testing if you are not sure and don't know exact compatibility with drivers based on your system. Select **"**Next**"** when finished.

6) Installation type:

Selection of a standard installation type is now available. The options include "Personal Desktop"**, "**Workstation"**, "**Server", or "Custom". Choices will vary depending on version. Note: Depending on your experience with the packgaes you need you can select " Custom" to aid in OS minimalization. This is similar in concept to the Solaris installation process, which requires the selection of one of five installation clusters. I selected "Server" and then **"**Next**"**.

7) Partitioning your system:

You are now faced with the option of automatically partitioning your system with the default values or selecting a more expert approach of choosing exactly what values each partition will be. I chose "Automatic", then selected **"**Next**"**. If you

choose you can “Select Manually partition with Disk Druid”. Then create these following partitions:

Partition	Type	Size
/boot	ext3	64MB
swap	swap	2x physical RAM, minimum of 1024MB
/	ext3	1024MB
/tmp	ext3	1024MB
/usr	ext3	3072MB
/var	ext3	512MB
/home	ext3	3072MB (this should be large enough to hold the three Oracle9*i* installation CDs)
/opt	ext3	6144MB (fill to maximum allowable size)

Recommended Partitions [9]

Multiple disks are allocated to the Oracle installation tree Oracle can be installed onto, and will run from, a single disk drive. With the exception of extremely small datasets, this always results in extremely poor performance, as Oracle reads and writes to many different files in many different locations quite often. In order to provide acceptable database performance for even the smallest log files, the necessary Oracle tablespaces should be distributed across at least four (4) separate physical disk drives. Use the Oracle Optimal Flexible Architecture.

8) Warning dialog box appears:

If this is a new hard drive or a hard drive that no partitions currently exist, a warning dialog box will appear. Select “Yes” to continue.

9) Automatic Partitioning:

Here are the options of deleting all Linux partitions, deleting all partitions, or keep the disk structure as it already is. If this is a new drive, any options work just fine, but if you already have partitions defined, as in a Multi-Boot environment (Don’t do this on a server), be careful as to which selection and drive volume you choose. Also, check the “Review and modify” box at the bottom to retain control over what happens to the hard drive and view the recommended configuration.

10) New hard drive structure:

If you checked the Review and modify box, the new hard drive structure is displayed. This fits my purposes, so I chose **“**Next**”** to continue

11) Boot loader options:

Boot Loader options are displayed. If this is the only OS to be installed (which it should be), I recommend install a boot loader, such as Grub (the default).
If this were part of a Multi-Boot System (NOT RECCOMMENDED on a server), I would *not* install a boot loader and use a boot floppy, created later on in the install process, instead.

Important: If you chose to NOT install a boot loader and NOT make a boot floppy, your Red Hat installation will NOT work.

12) Install a boot loader:

13) Network setup:

Network setup options are “Next”. “Enter” in the required information for your particular setup, here. If this is was standalone system or on an unprotected network without a DHCP server you will have to set up your network connection manually. For ease of installation I chose the default, DHCP setup, then selected **“**Next**”**.

14) Firewall setup:

Firewall setup is very important! If this system will be connected directly to the Internet, choose High to start out with. Remember this is the IPtables or IPchains portion. If this system is already behind a hardware firewall or router, choose Medium as a good starting point.

15) Language Selection:

Additional Language options are also available. Since I only understand English, the default was fine for me.

16) Time zone Selection:

Configure your Time Zone with this display. Being on the East Coast, that was selected here. Choose what is right for your location and then select **“**Next**”** to continue.

17) Create a root or administrator password:

Another important part of the installation process is to create a root or administrator password and a "normal" user account for everyday tasks, plus your oracle accounts.

DO NOT leave any of your passwords blank especially the root password and don’t make it easily guessable. Use an acronym from a sentence that you would

easily remember and use numbers and special characters like the $ for the “s” or the @ sign for an “a”. Also once installed make sure you change the Oracle default passwords.

18) Add users:

After acceptance of your root password, select the “Add” button to create an additional account for everyday tasks. Do not worry. If at anytime you need to use the root account to change system settings, you can. You will be prompted for the password even if you are logged in as a regular user. After creating a new account and selecting a "good" password, select the “OK” button.

19) Add as many users as you wish:

Add as many users as you wish, then select **“**Next**”** to continue.

20) Default package configuration:

Here, you have the option of accepting the default package configuration, or selecting exactly what applications and services you desire. If you wish, you can skip this step and select **“**Next**”** as the default option and your system will be configured accordingly. However, if you desire to add or subtract particular applications, choose the “Customize packages” to be installed button and select **“**Next**”**. This is where OS Minimalization can start to take place, but be sure you know what you need. This usually requires much testing based on your application and version numbers of software.

21) Selecting exactly what packages:

Here you will have the option of selecting exactly what packages and applications you wish to install. Detailed descriptions about each are also available, after selecting the check box on the left of the category, by selecting the details button. This is where OS Minimalization can start to take place, but be sure you know what you need. This usually requires much testing based on your application and version numbers of software. It is useful to know such things as if you are going to use “ssh” then “ssl” is require and if you are going to compile code on the box then “gcc” is required. Upon completing your options, select **“**Next**”** to continue.

22) Installation will now start:

Installation will now start after selecting the **“**Next**”** button.

23) Formatting of the hard drive:

Formatting of the hard drive or partitions will begin. Depending on the size of the hard drive, this may take much time.

24) Transfer of the install image:

Transfer of the install image to the hard drive now takes place.
Again, if you selected many packages, this could take some time.

25) Installation of all selected packages:

Installation of all selected packages and applications are now underway. Depending on what was selected and system configuration, this could take 15 minutes to over an hour.

26) Insert the “Next” CD soon:

Do not go far, though, as you may be required to insert the “Next” CD soon.

27) Boot floppy creation:

May not be needed if this is the only OS to be installed, I recommended earlier to install a boot loader previously, such as Grub (the default). This does NOT mean that you should forget about creating a boot floppy, you might want to consider that too.

Important: If you chose to NOT install a boot loader and NOT make a boot floppy, your Red Hat installation will NOT work.

28) Graphics card setup:

Select your graphics card setup and memory configuration here. This may take some testing if you are not sure and don’t know exact compatibility with drivers based on your system. Chose **“**Next**”** after highlighting your selection.

29) Monitor Setup:

“Enter” the proper values or highlight the model of your display here. After selecting the configuration for your system, chose “Next”. This may take some testing if you are not sure and don’t know exact compatibility with drivers based on your system.

30) Desktop resolution:

Choose your desktop resolution and bit-depth. Capabilities beyond your card should not be displayed.

31) Install complete:

Installation is completed and the system will reboot after selecting **"**Next"**.**

32) Grub boot loader:

Upon reboot, the Grub (if selected previously in the install process) is displayed with the option of booting your Linux installation. If you opted for a boot floppy, ensure that your system is configured to boot from "A:" first, then your hard drive, otherwise, your system may skip to your previous OS by default.

33) Boot Red Hat:

Keep in mind, while the OS is loading, Linux is famous for allowing the USER to choose exactly what, how and why everything happens. This is not the typical OS users at home would use. You will see all kinds of information about the internal workings of your system. If you are not used to Unix, do not be alarmed. This is what makes it so powerful.

34) Logon:

If you opted for a graphical install screen, the user name prompt is displayed. Log in as one of the users you created previously (not root) and "Enter" in your password (not blank).

35) Gnome Desktop:

The Red Hat default desktop is now displayed. If you selected KDE and not Gnome, your view will be slightly different.

Remember, anything that you wish to do, you can, including messing with system files. The old adage about measuring twice and cutting once is really important henceforth.

These installation instructions are based on my experience with Red Hat 9, Red Hat Enterprise 3 AS, ES and WS and "Black Viper's" Red Hat 8 Install Guide. [10. Viper]

If not done during installation you now must configure and update the operating system.

Because an Oracle database server is a networked service, it is critical that the network configuration for the operating system is correct. In addition, we will use Red Hat Network to update the system with the latest patches released by Red Hat. These patch releases contain bug fixes and security updates that should be applied to all production systems. The actions we will perform here must all be

performed while logged into the operating system as the root user. It is advisable to do this on your lab system first and then move the updates over to your "production system" since it is still not quite finished.

When you've finished the network configuration, you are ready to register the system with Red Hat Network and run the command "up2date" to update the system. Begin by running (as root) rhn_register to register your system: "rhn_register"

Note: Activation keys can also be used so that all of this can be done non-interactively

After the system is registered, it should automatically be entitled to receive RHN updates.

The kernel updates are disabled by default to give system administrators complete control over the kernel update process for production systems. You will need to configure the RHN entitled database server to receive kernel updates. To do this, edit /etc/sysconfig/rhn/up2date and find the line that reads:

pkgSkipList=kernel*;
and change it to:
pkgSkipList=;

Next, find the line that reads:

removeSkipList=kernel*;
and change it to:
removeSkipList=;

At this point, you can now use Red Hat Network to update your system with the latest patches available from Red Hat. To do this, you can either use the Red Hat Network Web interface or issue the following command on the system to be updated:

"up2date –u"

After the system has been successfully updated, take a look at /boot/grub/grub.conf and verify the default kernel is the latest kernel update [9].

Reboot to use the new kernel.

Oracle's Optimal Flexible Architecture (OFA)

Oracle Optimal Flexible Architecture (OFA), created by Cary Milsap in 1991, describes a standard organizational structure for Oracle databases that, if properly employed, will help improve the performance of Oracle databases. The Optimal Flexible Architecture generates some *common sense* rules for tablespace creation. [11. Piper] Those rules are:

- Segment types that are used the same way should be stored together.
- The system should be designed for its most common usage:
- Separate areas should exist for exceptions
- Contention among tablespaces should be minimized.
- The data dictionary should be isolated

Each tablespace, optimally, should be on separate disks and on separate controllers. This probably won't ever happen in the real world.

Many Oracle databases do not conform to the OFA guidelines. This section is an attempt to highlight that there is an Oracle guideline for setting up Oracle databases and to outline how to implemented OFA, together with some of the more practical benefits of using the guidelines.
One of the most important reasons for implementing the OFA guidelines is to improve the efficiency of support from Oracle. Both Oracle support and Oracle contractors will be able to find there way around your system and thus reducing both frustration and cost. [11. Piper]

Disk Partitions
Unless I know exactly how my databases is going to grow over the next 2+ years I usually set disk partitioning of data disks be kept to an absolute minimum. Thus if you have 36 Gb disks leave them as 36 Gb Partitions or if you have 72 Gb disks leave them as 72 Gb Partitions.

Mount Points
As recommended in the OFA try to use a fixed length arbitrary mount point naming convention as shown in the following two examples. The two-digit sequence number is used to identify different mount points. This helps maintain a fixed length for mount point names, it also a bonus when you want to perform some dba functions like the I/O balancing:

```
/disk01
/disk02
  OR
/d01
/d02
```

This example has used /d*nn* (where *nn* is a two digit sequence number) as the mount point naming for all examples.

Where To Put Oracle Source:
In accordance with the OFA put the Oracle source under the following directory:

/dnn/apps/oracle/product/10.1.0.2

Henceforth known as ORACLE_HOME. The important thing to notice is that the Oracle version number is part of the directory name. It is better to have only the Oracle source on this disk, as this will provide both the space and flexibility to install two versions of Oracle for when you want to upgrade.

Example:
/dnn/apps/oracle/product/9.2.0.1.0
/dnn/apps/oracle/product/10.1.0.2

NOTE: The home directory for the user Oracle must not be the same as ORACLE_HOME as you will tend to accumulate all sorts of files in the ORACLE_HOME directory, this will make it harder at upgrade time to identify what should be kept or what is part of Oracle.

Data Files

On each of the available disks create directory oradata. Under each of these oradata directories create a directory the same name as the Oracle SID:

Example:
/d02/oradata/SID
/d03/oradata/SID
/d04/oradata/SID

All datafiles, controlfiles, on-line redologs are placed in these directories.
Also by using this structure the length of the base component of the Oracle datafile will be the same, there for using a sub-string will allow you to strip off the directory structure from the data file name.

ORACLE_BASE

Oracle_base is a directory structure that provides a place for all Oracle related files and a further sub-division for all instance related files. I like this because once the directory structure is in place you and Oracle support will always know where to find the alert logs and trace files. [11. Piper]

The basic structure is as follows:
/d01/apps/oracle
/d01/apps/oracle/admin
/d01/apps/oracle/local
/d01/apps/oracle/TAR

/d01/apps/oracle/product

For each Oracle SID

/d01/apps/oracle/admin/SID/adhoc	Scripts for the given database
/d01/apps/oracle/admin/SID/adump	Audit trail trace files
/d01/apps/oracle/admin/SID/arch	Archive log files
/d01/apps/oracle/admin/SID/bdump	Background dump files
/d01/apps/oracle/admin/SID/cdump	Core dump files
/d01/apps/oracle/admin/SID/create	Scripts used to create the database
/d01/apps/oracle/admin/SID/exp	Export files
/d01/apps/oracle/admin/SID/pfile	database parameter files i.e. init.ora
/d01/apps/oracle/admin/SID/udump	User dump files

User Profiles

Have you ever undertaken an upgrade and found that ORACLE_HOME and ORACLE_SID has been hard coded into the users profile and you need to logon as root and change countless profiles?
Well, this is the one place I recommend the use of symbolic links.
In the /home directory create one or more profiles depending on the different types of users you have i.e.

profile_std
profile_dev

Link one of these profiles into each of the users home directories. Thus when you need to change a profile you can change it in one place.

Database Files

You cannot go too far wrong with database file names. The OFA recommends a relatively generic naming standard. The following reflects my preferred naming convention that does not deviate to far from the guidelines.
init.ora:

The init.ora should be in $ORACLE_HOME/dbs. Also place a copy or link a copy into the /disknn/apps/oracle/admin/SID/pfile directory.

Control files:

/d01/oradata/ORASID/control_01.ctl
/d02/oradata/ORASID/control_02.ctl
/d03/oradata/ORASID/control_03.ctl

Redo logs:

Assuming the logs are duplexed

/d01/oradata/ORASID/redo_01a.log
/d02/oradata/ORASID/redo_01b.log

/d01/oradata/ORASID/redo_02a.log
/d02/oradata/ORASID/redo_02b.log

/d01/oradata/ORASID/redo_03a.log
/d02/oradata/ORASID/redo_03b.log

Datafiles:
As per the OFA guidelines include the tablespace name as part of the datafile name. i.e.

system_01.dbf
temp_01.dbf
gl_data_01.dbf
gl_index_01.dbf

If we were using Solaris the installation process would be different based on the graphics used and requires the selection of one of five installation clusters:

Core
End User
Developer
Entire Distribution
Entire Distribution + OEM

The size of the Solaris clusters varies significantly. The size of the installation cluster also depends on which version of Solaris you were using. My experience has shown that, a secure server may require only 10 “Solaris 8” packages and use as few as 36 Mega Bytes of disk space, but for Oracle installations the minimum is Developer usually, because of the Oracle Universal Installer (OUI), Oracle Enterprise Manager (OEM) and development tools required.

During the installation process, you would be asked to partition your system. I always like to make root as big as possible and just throw everything in there, then you do not run out of room. However, we do need several partitions to protect the root drive. If we were to fill the root partition with data, such as logging or email, we would cause a denial of service, potentially crashing the system.
If a default installation was used, the Solaris installer tool divides the OS disk into several slices: / (root), /usr, /var and /export/home. A “Solaris 8” default installation creates three slices: (i) approximately 1GB root, (ii) swap (depending on memory capacity) and (iii) the remaining disk space for /export/home.

For workstations and servers, this partitioning scheme is not the optimum. The capacity of a partition cannot be expanded without data loss. In the case of a

partition running out of space, symbolic links to other partitions have to be created. It is useful, especially for servers, to reserve one partition for the /var directory, so that growing logfiles or big files stored into /var/tmp by users cannot fill up the root partition, thus causing OS problems.

Therefore, I always create a separate partition for /var; this is where all the system logging and email goes. By isolating the /var partition, you protect your root partition from overfilling. We have found 9 GB to be more then enough for /var. You may also consider making a separate partition for the /opt and /usr. If you create a separate partition for /usr, you can mount it read only, protecting the binaries from modification. If you are not mirroring the second drive and this is a firewall, make the second disk the partition for all the firewall logging. Once again, this protects all the other partitions in case the firewall logging floods the drive. With such a setup, your partitions would look as follows: A Workstation/Server OS disk should be partitioned this way:

slice 0 / 18 GB (1/4 of full disk); root-Partition, incl. /usr, /var, /opt plus /export/home if not partitioned out separately.

slice 1 swap Swap-Partition, 4 GB (or normally 2x amount of RAM) at least as large as large as physical memory as a minimum.

slice 2 backup Do not change this partition! It is always as big as the whole disk on Solaris systems.

slice 3 /usr 4 GB (mount as read only)

slice 4 /var 28 GB (rest of disk size; create home directories here in /var/users)

slice 5 /opt 18 GB (1/4 of full disk); Use /opt/local instead of the older /usr/local; this is a Sun Best Practice!

Once the system has rebooted after the installation, you must be sure to install the recommended patch cluster from Sun. Be sure to use your go between box to get the patches or burn them to a cdrom, the server should always remain on an isolated network. Patches are “CRITICAL” to maintaining a secure system and should be updated at least once a quarter.

This gives you the basics on installing a Unix Server and a Red Hat Linux server specifically. Now, let’s move on to the really important concepts of OS Minimalization and patching.

OS Mineralization

The idea here is to remove every package except for what is absolutely necessary for the functionality required by the applications that are going to be

used on the server. If your server is going to be a database server why would you need the packages for a web server or a mail server to be installed on it? That's the point you don't so remove them. The "man" (manual) pages are really helpful here giving you a list of which packages are associated with specific commands and configuration files. Make use of them.

Package Verification

Use the root terminal to make sure several key compatibility packages are installed on the Red Hat Enterprise Linux AS v.3 system prior to installing Oracle Database 10g. Confirm that the following are installed by typing:
rpm -q "package-name" for each of the following (example: rpm -q compat-gcc):
compat-gcc
compat-libstdc++
compat-gcc-c++
compat-db
make-3.79 or newer
binutils-2.11.90.0.8-12 or newer
gcc-3.2.3-2 or newer
openmotif-2.2.2-16 or a newer
setarch-1.3-1 or newer

If these packages are not present on the system use the "up2date" command to download and install them by typing:
up2date "package-name"
as the root user. [12 http://www.redhat.com/]

Here is the opportunity for you to fine-tune your OS installation; you can do rpm for Linux or pkgrm/pkgadd for Solaris.

One thing to consider after hardening is patch updates. Most patches once installed return the service back to it's "on" condition, which will subvert your OS minimalization.

Hardening with Bastille, JASS & Titan.

The Bastille Hardening System attempts to "harden" or "tighten" Unix operating systems. It currently supports the Red Hat, Debian, Mandrake, SuSE and TurboLinux Linux distributions along with HP-UX and Mac OS X. We attempt to provide the most secure, yet usable, system possible. The project is run by Jon Lasser, Lead Coordinator and Jay Beale, Lead Developer, and involves a number of developers, beta-testers and concept-creators. Bastille Linux was developed with several major goals [13. Beale]:

COMPREHENSIVENESS
Bastille Linux draws from every available major reputable source on Linux Security. The initial development integrated Jay Beale's existing O/S hardening experience for Solaris and Linux with most major points from the SANS' Securing Linux Step by Step, Kurt Seifried's Linux Administrator's Security Guide, and countless other sources.

INSTRUCTIVENESS
Bastille Linux has been designed to educate the installing administrator about the security issues involved in each of the script's tasks, thereby securing both the box and the administrator. Each step is optional and contains a description of the security issues involved.

COMMUNITY
Once the initial development was near complete, we brought the effort to the developers of the Bastille Discussion mailing list. Further, we began soliciting outside suggestions and testing. The script was GPL'd promptly and the Specification shared.

Download/Install Bastille 2.x

Bastille 2.x versions come in the following forms: RPM, HP-UX depot, Debian package and source tarball. To install Bastille 2.x on Red Hat from the RPM, is easy. It's been moved to a 1-rpm system, though you'll still need to install perl-Curses or perl-Tk through RPM or CPAN. So:

- Install the Bastille RPM, like this:

```
rpm -ivh Bastille-2.1.6-1.0.noarch.rpm
```

- Install perl-Tk (for our GUI) or perl-Curses (for console/text mode).

```
rpm -ivh perl-Tk-a.b-c.i386.rpm
```

or

```
rpm -ivh perl-Curses-d.e-f.i386.rpm
```

Note: you can also install perl-Tk/perl-Curses via CPAN.

Running

Now type: "bastille" to start the full custom hardening script.
Don't forget: Reboot the machine when you're done!

What Does Bastille Really Do?

Bastille encompasses many areas of security. The Perl scripts cover everything from firewall modules and bootup security to pluggable authentication modules (PAM) and account security. It also targets locking down inherently insecure services such as FTP, sendmail and inetd.

Titan

Titan is a collection of programs, each of which either fixes or tightens one or more potential security problems with a particular aspect in the setup or configuration of a Unix system. Conceived and created by Brad Powell, it was written in Bourne shell, and its simple modular design makes it trivial for anyone who can write a shell script or program to add to it, as well completely understand the internal workings of the system. [14 Archibald]

Titan does not replace other security tools, but when used in combination with them it can help make the transformation of a new, out of the box system into a firewall or security conscious system into a significantly easier task. In a nutshell, it attempts to help improve the security of the system it runs on. [13. Archibald}

> NOTE - Due to time and access to hardware/software resources, Titan 4.0 will only run on Solaris, versions 1.1.4 , 2.X, Solaris 8,and Solaris 9. Titan version 4.0 BETA6 works with Solaris, Linux, and Free BSD. These are minimalist modules for Linux and Free BSD. [13. Archibald]

Unix is often criticized for being a difficult system that is difficult and hard to secure. Some of the main reasons that a Unix system is un-secure include:

- It is nearly infinitely configurable.
- Unix can be very complex.
- Vendors don't ship systems secured by default (there are exceptions).
- It can require significant amounts of time, resources, and expertise to secure.

Once secure it will become less secure as time goes on through usage, patch updates and the continual flood of new security problems being discovered in the world.

Titan can help with all of these problems; its main goals are:

- After being run, the system should be more secure than when previously. Things may be broken, but it should be more secure! The truth is that most things you do to secure a system are probably not going to cause a problem. A vendor can't take that chance - but we can. In any case, we haven't run into anything that Titan has broken, but it certainly could happen.

- Security comes first, right along with functionality. If Titan has been run at its highest level of security, there will be no significant configuration security problems that I know of. The system will not be 100% secure - none are - but it will be pretty secure.
- Producing a consistent and understandably secure system.
- It can help create a programmatically defined technical aspect of a system or site's security policy. Allow the administrator to have complete control over what modules in Titan are run - with full source code and a fair bit of flexibility, it is easy to remove unwanted security fixes. After all, not everyone wants or needs all the actions that Titan does.
- Titan is easily extended. Shell scripts or other programs can be placed into Titan's framework, and they will be run alongside all the other programs. All you need do is build your scripts/code to produce output that it expects.

JASS

Solaris Security Toolkit (aka JASS) and formerly known as the JumpStart Architecture and Security Scripts

The Solaris (tm) Security Toolkit ("Toolkit") is a tool designed to assist in creation and deployment of secured Solaris Operating Environment systems. The Toolkit is comprised of a set of scripts and directories implementing the recommendations made in the Sun OnLine BluePrints program [15. Noordergraaf] (http://www.sun.com/blueprints).

These scripts can be executed on Solaris systems through the JumpStart technology or directly from the command line. The Toolkit includes scripts to harden, patch, and minimize Solaris systems. Sun does not support the Toolkit.

Amazingly, Sun Microsystems apparently now claims to be the full owner of Titan, from which Sun says it has "derived" JASS/ the Solaris Security Toolkit. Details regarding Sun's claims may be obtained by contacting local Sun representatives directly.

Install the Oracle Database Server

For the purposes of this document, we will be doing an installation of the Oracle database server under /opt/oracle, and using the General Purpose starter database provided with the Oracle installation media. The Oracle database System ID (SID) we will use is orcl. Some database administrators may choose to install Oracle using a different configuration. That said, nearly all of the steps below will still apply--merely use different values as appropriate for your installation.

The installation of Oracle under Unix or Linux can be a complicated process. It involves multiple configuration steps, with some commands issued as root, and some issued as the oracle user. It is critical that the kernel and system parameters are configured properly; otherwise, the creation of the database may fail. Accordingly, we will begin as the root user to configure the system, after that we will move on to the installation of Oracle itself.

Kernel Parameters for Red Hat: Oracle requires certain kernel parameters to be modified. As the root user, edit the file /etc/sysctl.cf to include the following:

```
# Disables packet forwarding
net.ipv4.ip_forward = 0
# Enables source route verification
net.ipv4.conf.default.rp_filter = 1
# Disables the magic-sysrq key
kernel.sysrq = 0
# Parameters for Oracle9i Release 2 (9.2.0)
kernel.sem = 250 32000 100 128
kernel.shmmax = 2147483647
kernel.shmmni = 4096
kernel.shmall = 2097152
fs.file-max = 65536
net.ipv4.ip_local_port_range = 1024 65000
```

Other kernel parameters are set to a certain limit for Oracle (check your documentation) some parameters are:

SHMMIN, SHMSEG, SEMMNS, SEMMNI, SEMMSL

After editing the /etc/sysctl.cf file, as root run the command:

"sysctl –p"

this will make the changes take effect. Or, you can just reboot your system. Sometimes, especially after making a number of changes it is best to reboot to make sure your system will come back up. It is especially frustrating to have a system hang during a scheduled system reboot to find out hours later that is not the most recent change that is causing the problem but one you applied months ago and did not reboot at that time.

Now we are ready to add the appropriate groups and other users required for the Oracle installation if not done previously. Issue the following commands (while logged in as root):

```
groupadd dba
groupadd oper
groupadd oinstall
```

useradd -g oinstall -G dba,oper orcl
passwd orcl

This will create a new user account (named orcl) whose primary group is oinstall and whose secondary groups are dba and oper.

Create a directory for the Oracle Database software :
mkdir -p /d01/app/oracle
chown -R oracle:oinstall /d01/app
chmod -R 775 /d01/app

You will need to increase the number of files the oracle user can open. Make sure PAM can read the /etc/security/limits.conf file by ensuring that the following lines are in /etc/pam.d/system-auth:

session required /lib/security/pam_limits.so
session required /lib/security/pam_unix.so
Add the following lines to /etc/security/limits.conf:
oracle soft nofile 4096
oracle hard nofile 8192

The above changes will go into effect the next time the oracle user logs into the system.

For Solaris the files edited or modified by the UNIX users root and oracle:

Kernel Parameters for Solaris: Oracle requires certain kernel parameters to be modified, and the parameters are found in the /etc/system file. As root, make a backup copy of this file before making any changes, this is always advisable before making any changes that will affect your system. Add or modify the following parameters, using any higher values if they already exist on your system. These settings can be placed at the end of the file.

```
set semsys:seminfo_semmni=100
set semsys:seminfo_semmns=1024
set semsys:seminfo_semmsl=256
set shmsys:shminfo_shmmax=4294967295
set shmsys:shminfo_shmmin=1
set shmsys:shminfo_shmmni=100
set shmsys:shminfo_shmseg=10
```

Another setting you will want to ensure you have to prevent buffer offerflows on Solaris is:
* Attempt to prevent and log stack-smashing attacks
set noexec_user_stack=1
set noexec_user_stack_log=1

You would also need to add users and groups as appropriate.

This section shows you how to use the Oracle Universal Installer (OUI) to install your Oracle database software and create a starter database. You will also learn how you can use the Database Configuration Assistant (DBCA) to create additional databases.

Installing the Software Using the Oracle Universal Installer (OUI)

You can use the Oracle Universal Installer (OUI) to install your Oracle software. The OUI is a GUI tool that enables you to view the Oracle software that is installed on your machine, install new Oracle software, and delete Oracle software that you no longer intend to use. Follow the steps below to install the Oracle software and create a database:

1. Log onto you computer as a member of the administrative group that is authorized to install Oracle software and create and run the database.

2. Insert the distribution CD for the database into your CD drive. The Autorun window will appear automatically. Select “Install/Deinstall” Products. Note: If you are downloading from Oracle’s download site, follow the instructions given on the Web site.

3. The Oracle Universal Installer Welcome window appears. Select “Next” to begin the installation of your software

4. On UNIX and Linux installations, the Specify Inventory directory and credentials window appears. Enter the full path of the directory in which to install the Oracle software or accept the default. Enter the name of an operating system group that has write permission to the directory. Click “Next”. Click “OK.”

5. A dialog page appears. Open a new terminal window, log in as “root”, and run the “orainstRoot.sh” script as instructed in the dialog page. When the script finishes, return to the Oracle Universal Installer page and click “Continue”.

6. The Specify File Locations page appears. Enter the Oracle home name and directory path in which to install the Oracle software or accept the default. Click “Next”.

7. The Select Installation Type page appears. Select Enterprise Edition or Standard Edition as appropriate for your environment. Click “Next”. I chose Standard Edition.

8. The installer will now verify that your system meets all the minimum requirements for installing and configuring the chosen products. Correct any reported issues before continuing. Click “Next”.

9. The Select Database Configuration page appears. Select the starter database type of “General Purpose, Transaction Processing, or Data Warehouse”. I chose General Purpose. Click “Next”.

10. The Specify Database Configuration Options page appears. Enter the Global Database Name and SID in the Database Naming section. Select the character set in the Database Character Set section. Select Create database with Sample Schemas to install the Sample schemas. Click “Next”.

11. The Select Database Management Option page appears. Select Use Database Control for Database Management. Click ““Next””.

12. The Specify Database File Storage Option page appears. Select File System, Automatic Storage Management, or Raw Devices as appropriate to your environment. Click “Next”.

13. The Specify Backup and Recovery Options page appears. Select Do not enable Automated backups to configure your own backup schedule. Click “Next”.

14. The Specify Database Schema Passwords page appears. Select Use different passwords for these accounts and enter passwords for the administrative users or select Use the same password for all the accounts and enter the password. Click “Next”.

15. The Summary page appears containing a list of the products to be installed. Click “Install” to begin the installation.

16. You will see the progress window.

17. The Configuration Assistants page appears. Allow the assistants to execute. They configure your network, start an Oracle Net Services listener process for connecting to the database, create the database and configure management tools.

18. A page showing a progress bar for database creation appears.

19. When the database creation is finished, a page containing information about your database appears. Review this list. You may click on “Password Management” at the bottom of the page if you want to unlock or change passwords for database accounts. Otherwise click “OK”.

20. A dialog page appears. Open a new terminal window, log in as “root”, and run the “root.sh” script. When the script finishes, return to the Oracle Universal Installer page and click “OK”.

21. The End of Installation page appears with important information about Web application port numbers.

22. Click “Yes” to exit.

23. You have now completed the installation of Oracle software and database creation.

Using the Database Configuration Assistant (DBCA) to Create a Database

If you choose to install software only and later create a database, or if you want to create additional databases using the release software that you just installed, you can do so by using the Database Configuration Assistant (DBCA). Follow the steps below to create a database:

1. Log onto your computer as a member of the administrative group that is authorized to install Oracle software and create and manage the database.

2. Launch the DBCA on a UNIX operating system by entering the following at a command prompt: “dbca”

3. The Welcome page appears. Click “Next”.

4. Select Create a Database on the Operations window to begin an interview that enables you to configure and create a database. Click “Next”.

5. On the Database Templates page, select the type of database template to be used in creating the database. You can click Show Details to see the configuration for each type of database. Choose the template suited to the type of workload your database will support. If you are not sure, select the default General Purpose template. Click “Next”.

6. On the Database Identification page, enter the Global Database Name and SID. Click “Next”.

7. The Management Options page appears. To use Enterprise Manager, select Configure the Database with Enterprise Manager. Select “Use Database Control for Database Management” to manage your database locally or select “Use Grid Control for Database Management” as appropriate to your configuration. Click “Next”.

8. Enter passwords for the administrative users. Click “Next”.

9. Select File System, Automatic Storage Management, or Raw Devices as appropriate to your environment. Click “Next”.

10. Specify the location for the creation of the datafiles. Choose one of the following: Use Database File Locations from Template, Use Common Location for All Database Files, or Use Oracle-Managed Files. Click Next.

11. Select Flash Recovery Area and specify a directory location and size. Select Enable Archiving to place your database in ARCHIVELOG mode. Click “Next”.

12. Click the Sample Schemas tab. Select Sample Schemas if you want to include the Sample Schemas (EXAMPLE) tablespace in your database. Click the Custom Scripts tab.

13. Specify one or more SQL scripts to be run after your database is created if required in your environment. Otherwise, accept the default “No scripts” to run. Click “Next”.

14. The Memory page appears. Select “Typical” and enter a percentage value. Click the Sizing tab.

15. The Sizing page appears. Specify the smallest block size and the maximum number of operating system user processes that can simultaneously connect to the database. Note: You cannot specify the block size if you are using a template. Click the “Character Set” tab.

16. The Character Set page appears. Select the character set for your database. Click the “Connection Mode” tab.

17. The Connection Mode page appears. Select Dedicated Server or Shared Server as appropriate for your environment. Click “Next”.

18. The Database Storage page appears. Accept the configuration or make changes as needed. Click “Next”.

19. Select Create Database to create your database. You can also select Save as a Database Template to save your configuration. Click “Finish”.

20. At the Confirmation window, confirm the options that will be installed and click “OK”.

21. Your database is now being created.

22. After the database is created, you can change or unlock your passwords or click “Exit”.

The two previous guides “Installing the Software Using the Oracle Universal Installer (OUI)” and “Using the Database Configuration Assistant (DBCA) to Create a Database”, are loosly based on my experience with Oracle 8i and 9i and specifically on Oracle’s two-day 10g DBA class. [16 Burbridge]

Oracle Database Checklist

The checklist provided by Peter Finnigan is comprehensive and should be used to audit an Oracle database installation. This checklist is just that “a checklist” and does not contain any specific SQL or shell commands because it is intended to be just a list. It is also important that the Oracle database is not checked in isolation and the surrounding elements such as the operating system used, the network configuration, web access, application servers and clients are considered. [17. Finnigan]

The Oracle database server leads the industry in security. However, in order to fully maximize the security features offered by any Oracle version in a business environment, it is imperative that Oracle itself is well protected. Furthermore, proper use of its security features and adherence to basic security practices will help protect against database related threats and attacks and provide a much more secure operating environment for the Oracle database product line.

1. Install only what is required.
2. Lock and expire default user accounts.
3. Change default user passwords.
4. Enable data dictionary protection.
5. Practice principle of least privilege.
6. Enforce access controls effectively.
7. Restrict network access.
8. Apply all security patches and workarounds. [18. Sinha]

Conclusion

As you have seen, there are many different tasks to perform across several applications and sometimes servers and the network. If your Oracle Database is to be unbreakable, it is up to you to take the steps necessary to make it so.

REFERENCES

1. Thompson, Herbert. "Warning: Security Storm Brewing," Better Software. October 2004. URL: http://www.stickyminds.com/BetterSoftware/magazine.asp?fn=cifea&ac=185 (November 15, 2004)

2. Husain, Abdur-Rahman. "Considerations for Securing Data in Oracle Data Bases." Apr 23, 2004. URL: http://www.giac.org/practical/GSEC/Abdur-Rahman_Husain_GSEC.pdf (September 22, 2004)

3. McAlearney, Shawna. " Multiple critical flaws identified in Oracle." SearchSecurity.com. Aug 04, 2004. URL: http://searchsecurity.techtarget.com/originalContent/0,289142,sid14_gci996801,00.html?track=NL-333&ad=488249

4. Morrissey, Peter. "Implementing Access-Control Lists: Access Control," Network Computing. March 18, 2004. URL: http://nwc.securitypipeline.com/howto/18400169 (November 15, 2004)

5. Nolle, Tom; "Location vs. Address: Tunnels, VLANs, and VPNs," Network Magazine. Sep. 01,1999. URL: http://www.networkmagazine.com/shared/article/showArticle.jhtml?articleId=8702532&classroom (November 15, 2004)

6. Ciolek, Dr T.Matthew. "The pioneer site for Asia-Pacific research and electronic publishing" The Coombsweb, Last updated: 9 Nov 2004. URL: http://coombs.anu.edu.au/~avalon/ (November 15, 2004)

7. Sadoski, Darleen (GTE} and Comella-Dorda, Santiago (SEI). ''Three Tier Software Architectures, Last updated: 16 Feb 2000. URL: http://www.sei.cmu.edu/str/descriptions/threetier.html (November 15, 2004)

8. Heimann, John. "Securing Three-Tier Systems with Oracle8i," Oracle Technology Network, Security Collateral for OOW 1999. URL: http://www.oracle.com/technology/deploy/security/oow99.htm (September 22, 2004)

9. "Deploying Oracle 9i on Red Hat Enterprise Linux AS 2.1" URL: http://www.redhat.com/solutions/info/whitepapers/ (November 21, 2004)

10. Viper, Black. "Red Hat Install Guide" Last updated: 24 Oct. 2004. URL: http://www.blackviper.com/Articles/OS/#Red_Hat_80 (November 15, 2004)

11. Piper, G & J. "Oracle and Oracle Applications Tech Tips" Last updated: June 15, 2000. URL: http://members.ozemail.com.au/~gpiper/oracle/ora_8.html (November 21, 2004)

12. "Deploying Oracle 10g on Red Hat Enterprise Linux v.3" URL: http://www.redhat.com/solutions/info/whitepapers/ (November 21, 2004)

13. Beale, Jay and Lasser, Jon. Bastille Linux Project, "Bastille Hardening System" URL: http://www.bastille-linux.org/ (November 21, 2004)

14. Archibald, Matthew; Farmer, Dan; and Powell, Brad M. "Titan Security Toolkit Release 4.1" Last updated: November 6, 2004. URL: http://www.fish.com/titan/ (November 21, 2004)

15. Noordergraaf, Alex; (Enterprise Engineering) and Brunette, Glenn; (Sun Professional Services). "The Solaris™ Security Toolkit- Installation, Configuration and Usage Guide." June 2001. URL: http://www.sun.com/blueprints/0601/jass_conf_install-v03.pdf

16. Burbridge, John. Oracle Technology Network, "Chapter 2: Installing Oracle Software and Building the Database." Last updated: 8/14/2003. URL: http://www.oracle.com/technology/obe/2day_dba/install/install.htm (November 04, 2004)

17. Finnigan, Pete. "Oracle Database Checklist." Oracle security step-by-step, A Survival Guide for Oracle Security. Sep 20, 2004. URL: http://www.sans.org/score/checklists/Oracle_Database_Checklist.pdf (September 22, 2004)

18. Sinha, Ranjan. "A Security Checklist for Oracle9*i*"; An Oracle white paper April 23, 2001. URL: http://www.cgisecurity.com/database/oracle/pdf/9i_checklist.pdf (September 22, 2004)

Last Updated: October 2nd, 2015

Upcoming Training

SANS Seattle 2015	Seattle, WA	Oct 05, 2015 - Oct 10, 2015	Live Event
Mentor Session - SEC 401	Downers Grove, IL	Oct 08, 2015 - Dec 17, 2015	Mentor
SOS: SANS October Singapore 2015	Singapore, Singapore	Oct 12, 2015 - Oct 24, 2015	Live Event
SANS Tysons Corner 2015	Tysons Corner, VA	Oct 12, 2015 - Oct 17, 2015	Live Event
Mentor Session - SEC401	Philadelphia, PA	Oct 12, 2015 - Nov 23, 2015	Mentor
SANS Gulf Region 2015	Dubai, United Arab Emirates	Oct 17, 2015 - Oct 29, 2015	Live Event
SANS Cyber Defense San Diego 2015	San Diego, CA	Oct 19, 2015 - Oct 24, 2015	Live Event
Community SANS Anaheim SEC401	Anaheim, CA	Oct 26, 2015 - Oct 31, 2015	Community SANS
Community SANS Paris SEC401 (in French)	Paris, France	Oct 26, 2015 - Oct 31, 2015	Community SANS
Mentor Session - SEC401	Des Moines, IA	Nov 02, 2015 - Jan 18, 2016	Mentor
SANS vLive - SEC401: Security Essentials Bootcamp Style	SEC401 - 201511,	Nov 02, 2015 - Dec 09, 2015	vLive
Community SANS New York SEC401	New York, NY	Nov 09, 2015 - Nov 14, 2015	Community SANS
SANS South Florida 2015	Fort Lauderdale, FL	Nov 09, 2015 - Nov 14, 2015	Live Event
SANS Sydney 2015	Sydney, Australia	Nov 09, 2015 - Nov 21, 2015	Live Event
Sydney 2015 - SEC401: Security Essentials Bootcamp Style	Sydney, Australia	Nov 09, 2015 - Nov 14, 2015	vLive
Community SANS San Antonio SEC401	San Antonio, TX	Nov 09, 2015 - Nov 14, 2015	Community SANS
Community SANS Madrid SEC401 (in Spanish)	Madrid, Spain	Nov 09, 2015 - Nov 14, 2015	Community SANS
SANS London 2015	London, United Kingdom	Nov 14, 2015 - Nov 23, 2015	Live Event
Community SANS Salt Lake City SEC401	Salt Lake City, UT	Nov 16, 2015 - Nov 21, 2015	Community SANS
Community SANS Kansas City SEC401	Kansas City, MO	Nov 16, 2015 - Nov 21, 2015	Community SANS
SANS San Francisco 2015	San Francisco, CA	Nov 30, 2015 - Dec 05, 2015	Live Event
Community SANS Chicago SEC401	Chicago, IL	Nov 30, 2015 - Dec 05, 2015	Community SANS
SANS Cape Town 2015	Cape Town, South Africa	Nov 30, 2015 - Dec 05, 2015	Live Event
Community SANS Jacksonville SEC401	Jacksonville, FL	Nov 30, 2015 - Dec 05, 2015	Community SANS
Community SANS Omaha SEC401	Omaha, NE	Nov 30, 2015 - Dec 05, 2015	Community SANS
Security Leadership Summit & Training	Dallas, TX	Dec 03, 2015 - Dec 10, 2015	Live Event
Community SANS Toronto SEC401	Toronto, ON	Dec 07, 2015 - Dec 12, 2015	Community SANS
SANS Cyber Defense Initiative 2015	Washington, DC	Dec 12, 2015 - Dec 19, 2015	Live Event
Cyber Defense Initiative 2015 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Dec 14, 2015 - Dec 19, 2015	vLive
Community SANS Portland SEC401	Portland, OR	Jan 04, 2016 - Jan 09, 2016	Community SANS
Mentor Session - SEC 401	Milwaukee, WI	Jan 06, 2016 - Mar 09, 2016	Mentor