Global Information Assurance Certification Paper

Operating Environment Minimisation for Security

Research Paper for SANS GIAC (GSEC) certification

Based on SANS Security Essentials GSEC Practical Assignment Version 1.3

(Amended December 12, 2001)

Prepared by Jeffrey Bailey February-March 2002

1	Trademarks	1
2	Overview	2
3	What is Operating Environment Minimisation for Security?	3
4	Why Use Operating Environment Minimisation for Security?	5
5	Where Does Operating Environment Minimisation Fit In?	7
6	Choosing the Right Operating Environment Profile	10
6.1	What is an Operating Environment Profile	10
6.2	Reference Sites for Known Good Base Profiles	10
6.3	Creating New Profiles by Research and Testing	13
7	Performing Operating Environment Minimisation for Security	15
7.1	General	15
7.1.1	Implementing During the Install	15
7.1.2	Implementing with Automated Installations	15
7.1.3	Implementing After a Basic Installation	16
7.2	Solaris in Detail	17
7.2.1	Implementing During the Install	18
7.2.2	Implementing with JumpStart	18
7.2.3	Implementing After a Basic Cluster Install	19
8	References	22

1 Trademarks

Sun, Sun Microsystems, Java, UNIX, Solaris, Java JumpStart, JumpStart, Solaris JumpStart, Solstice JumpStart, WebStartFlash, Solaris Security Toolkit, Java BluePrints, Sun BluePrints, SunScreen EFS and the SUN logo are trademarks or registered trademarks of Sun Microsystems, Inc.

SSH is a registered trademark of SSH Communications Security in the United States and in certain other jurisdictions.

All other company, brand and product names maybe registered trademarks or trademarks or their respective companies and are hereby recognised.

2 Overview

A very common and effective security measure is the minimisation of operating environments, or as they are more commonly known, operating systems[1]. Effective operating environment minimisation is an important part of overall host security.

This paper covers: -

- The concept of “Operating Environment Minimisation for Security”
- How “Operating Environment Minimisation for Security” relates to other host based security measures that are in use today
- Some guidelines on how to achieve “Operating Environment Minimisation for Security”
- The application of “Operating Environment Minimisation for Security”, to the Sun Microsystems Solaris 8 operating environment

[1] Both operating environments and operating systems will hereinafter be referred to as operating environments

3 What is Operating Environment Minimisation for Security?

"Operating Environment Minimisation for Security" is the process of removing (or better still not installing in the first place) all non-essential components of the operating environment. This will leave the bare minimum installation footprint that is required to support the applications or functions that must be performed by a computer.

"Operating Environment Minimisation for Security" is a key step in the hardening[2] of a host for enhanced security and is particularly important for any server in a hostile environment. The less operating system software that is present on a server, the less software has to be inspected for vulnerability's, patched, and otherwise defended from attack.

Almost all environments are to some extent hostile, and ideally all hosts should be hardened. Due to limited time and resources, most of the effort is normally focused on very vulnerable servers (those with many known security weaknesses), and servers that face the most dangerous environments, such as servers placed on what are known as demilitarised zones[3] (DMZs), or servers placed directly on the public Internet.

"Operating Environment Minimisation for Security" is normally achieved either during the installation process, by choosing not to install unnecessary operating environment components, or after the install process, by removing unnecessary operating environment components. The ideal situation is often to use a combination of the two, installing a minimum base operating environment, and then doing more fine grained removal and addition of operating environment components to suit the requirements of the particular application(s) that must run on a computer.

Some automated installation methods, such as the Sun Microsystems JumpStart technology for Solaris, are capable of achieving full minimisation during the installation phase, because options are available using these technologies that are not present in the normal interactive installation procedures.

Many vendors do not consider security when creating operating environments, system software or application software; even if security has been addressed to some extent, in most cases vendors do not document the minimum installation level of the operating environment as part of their system requirements. In many cases the system administrator must take care of operating environment minimisation. Furthermore operating environment hardening of any sort can create serious vendor support issues, which may impact your ability to get vendor support or assistance

[2] Hardening is a term commonly used to describe the process of making adjustments to an operating environment or application to make it more secure or "hardened" to resist attacks.

[3] Demilitarised zones in this context are semi-protected (normally firewall controlled) network segments, often used to allow tightly controlled external access from business partners, or the public Internet, to servers that do not store sensitive data, but that have very controlled access to obtain this data from more protected network segments.

with software problems that may arise, and in some cases even with hardware.

It is recommended that you maintain a very open dialogue with a vendor, to establish how your proposed operating environment hardening (including minimisation) may affect their support position. Inquiring about any vendor-supported recommendations in this regard, such as toolkits or white papers, can save much heartache, downtime, lost revenue and perhaps even your job.

4 Why Use Operating Environment Minimisation for Security?

Due to the complexity of possible support issues, the difficulty of obtaining good information from vendors on system requirements, the faith that people have in their other host security hardening measures, and the unbounding faith that people have in their firewalls, many people feel that operating environment minimisation for security is just too hard. Why would one go to so much trouble to put such a security measure in place?

The primary reason to deploy any computer security measure is to reduce risk. The only real claim that any computer security measure can make is that it reduces the likelihood of events that constitute risks, or events that could lead to other events that constitute risks. Thus once a measure is in place the real result is that the event constituted to be a risk is very unlikely to take place. The sad converse of this is that should a particular very unlikely event take place; then the event that constitutes a risk can then take place.

So what we are basing our perception of security on is that fact that an unlikely but not impossible event will not take place. The computer security community, and the general security and military communities, realised long ago that basing one's whole defence on a single measure left one very vulnerable to the possible failure of that measure. The principal of security that is employed to manage this situation is called "Defence in Depth".

The idea of "Defence in Depth" is rather simple at the high level. Simply place multiple layers of security measures is place so should one unlikely event occur, and your outermost security measures come tumbling down (and perhaps even become weapons to be used against you), the attacker must then penetrate another defence, and then another, and so on. "Defence in Depth" will slow down attackers, so that there is a better chance of you detecting them before they have access to the very sensitive material you are trying to protect. It also means that now we basing our perception of security on the possibility that a number (the larger the better) of very unlikely but not impossible, and hopefully unrelated events will all take place. We learn from this principal that an effective security solution should place as many difficult to circumvent obstacles in the path of an attacker as possible, thus reducing the chances that all the measures can be compromised.

The specific problems that exist with relying on hardening measures that do not employ minimisation as part of the hardening approach, is that these normally disable a large number of dangerous services that may have been exploited by attackers, but the service software itself remains installed on the computer.

Disabling but not removing potentially dangerous services (read "any services"), leaves the way open for an attacker to re-activate these services easily should they comprise the host[4], thus leaving them installed saved the attacker time and effort, and perhaps placing the compromise of our security within the reach of an attacker

[4] Even when computer security "best practices" are applied, any system that serves any purpose must be running programs. Any computer running programs, in particular programs that are accessible is some way over a network is potentially vulnerable to attack. So even a web server with all ports closed and firewalled off except TCP 80 and TCP 443, with only a supposedly "secure" e-commerce

who would have otherwise failed in their efforts[5], rather than placing as many difficulties in their path as possible as the principal of “Defence is Depth” advocates”.

The specific problems that exist with trusting firewalls to “cover a multitude of sins” are that firewalls can be misconfigured and by their very nature they pass some traffic. Sometimes the traffic allowed to pass firewalls can either be cleverly engineered to achieve convert ends, or more often attackers compromise the service that the firewall allows access to. As firewalls like everything else made by man are imperfect it is possible that the firewall software itself may crash, this can have unpredictable results, and perhaps even leave a previously reasonably secure gateway wide open, allowing all traffic. As with other software firewall software itself may also have known or unknown vulnerabilities that may be exploitable.

The specific advantages of operating environment minimisation for security on a particular computer are that: -

- Operating environment components that are not installed cannot be exploited
- Monitoring for new vulnerabilities and the corresponding work around fixes and/or solutions is not necessary for operating environment components that are not installed
- Operating environment components that are not installed do not need to be patched
- Security analysis and hardening need not be performed for operating environment components that are not installed

application’s web server component running is still a potential vulnerability.

[5] In some cases simply activating a service that is already installed is very easy, and much less likely to be detected than uploading the software and installing it. If the nature of the attacker’s access to a system is limited the upload and install steps may perhaps place the attack beyond an attacker’s abilities, and save your sensitive material from compromise at least on this occasion.

5 Where Does Operating Environment Minimisation Fit In?

"Operating Environment Minimisation for Security" is an essential part of host based security. Ideally this should be the first security measure implemented on a computer because it effects decisions made during the operating environment installation process. Once the operating environment is installed a certain level of security should already have bee achieved.

"Operating Environment Minimisation for Security" lays the groundwork for all other security measures at the host level, and often reduces the work that needs to be done in other areas by creating a more streamlined and simplified system.

Creating a simple system without unnecessary operating environment components is in harmony with one of the basic security principles… "Know Thy System". The less complex the system is, the more achievable it is to "Know Thy System". To "Know Thy System", one needs to keep up with vulnerabilities, patches and the many other moving targets that are involved in administration of a security sensitive system for all the installed system and application software, so the less software installed on your system, the better.

While "Operating Environment Minimisation for Security" is a foundation level host based security measure it in no way replaces the many other valuable host based, network based, and administrative practice measures that should be in place to enhance the security of a system, network and by extension an enterprise.

At the host level, in addition to "Operating Environment Minimisation for Security", one should consider the following measures as important steps toward computer security best practice: -

- Operating Environment Hardening
  This is the process of configuring an operating environment for enhanced security. Hardening often involves modifying configuration files/directories to disable or enhance the security of system daemons/services, enabling security related operating environment features, setting up stronger user access control, and modifying file permissions to more secure values. In recent years a large number of tools and papers have been produced to assist with these tasks. Some of the most notable work on the Solaris platform has been the "Armouring Solaris" papers and scripts by Lance Spitzner, The "Titan" tool by Brad Powell, and the "Solaris Security Toolkit" (formerly known as JASS) by Alex Noordergraaf. Links to relevant web addresses for these are listed below: -

  - http://www.enteract.com/~lspitz/armoring.html
    Armouring Solaris by Lance Spitzner
  - http://www.enteract.com/~lspitz/armoring2.html
    Armouring Solaris II by Lance Spitzner
  - http://www.fish.com/titan/
    Titan Hardening Tool by Brad M. Powell, Dan Farmer, and Matthew Archibald

- http://www.sun.com/security/jass/
  The Solaris Security Toolkit by Alex Noordergraaf

- Host Based Intrusion Detection Systems (IDS)
  This measure involves running a daemon on the host which monitor important aspects of the system, such as logs, system files, and network interfaces, trying to detect suspicious events that may indicate that an attack is occurring or may have already occurred. When events are detected IDS systems normally send alerts in various ways.

- Host Based System Integrity Monitors
  Specific integrity monitors, such as the "Tripwire" family of products, monitor important files or other modifiable items on a system and report on and/or send alerts about changes as appropriate. The monitors will normally be configured in such a way that they only send alerts when such changes may indicate malicious activity. Integrity monitors are often a component of the "Host Based IDS" systems referred to above.

- Host Based Firewall software
  These products insert themselves into the networking software of the computer and allow the enforcement of a security policy that specifies which incoming and outgoing network connections will be allowed or disallowed; in some cases users are interactively prompted with questions that allow the product to create these policies on the fly. The metrics on which policies can evaluate traffic vary, but normally include such things as: the source IP address, destination IP address, service (often denoted by a TCP or UDP port number), and is some cases the program initiating the connection or listening on the port to which the connection is directed. Many host-based firewalls also include logging functions.

- Host Based Virus Detection/Repair Software
  For many years virus scanners have been available to check a local computer for known viruses or other malicious code. In recent years a number of more advanced features have been added to allow files to be scanned as they are accessed, to allow web/email traffic to be scanned as it is transmitted from or received by a computer, and to detect. Some products can also and offer protection against malicious and/or potentially dangerous code such as Java or ActiveX that is included as content in web traffic. (re read the last two sentences of this paragraph

- Host Based Security Auditing Tools
  These tools are used to check a single host for known vulnerability's security-related misconfigurations and any other security-related issues of which the tool is aware. Some of these tools also can optionally correct some or all of the problems they detect, thus making the tool a hardening tool as well as an auditing tool. Some of the more famous tools of this type are Titan, COPS, Tiger and TARA. Information on these tools which happen to be designed for Unix style systems can be found at the following web address: -

  - http://wwwinfo.cern.ch/dis/security/general/tools/detect.html
    List of "Local Host Security Tools"

- System Administration Security Tools – Unix systems, and some other types of operating environment, have a very low administrative privilege granularity: that is, an all or nothing approach to administrative privileges on the computer. This poses a problem for delegation of the more mundane tasks to staff in positions of lower responsibility, because allowing them to do mundane tasks also allows them unlimited access to all other aspects of the system. The all or nothing approach also creates problems when trying to track changes and events in logs because often all staff with administrative access log in as the same user, so actions by a one staff member are indistinguishable from those of another in the system logs. A number of tools exist in the Unix world that allow administrative access to users logged in under their own credentials, only giving administrative access to specific functions that are appropriate for that user, and specifically logging all administrative activity performed by that user. For example, the "sudo" program provides this functionality.

6 Choosing the Right Operating Environment Profile

6.1 What is an Operating Environment Profile

An “Operating Environment Profile” is simply a description of which components of an operating environment to install, expressed in a way that makes sense for that operating environment. When performing operating environment minimisation for security, the aim is to establish what is the most minimal portion (number of components/sub-systems) of operating environment that can be installed, without preventing the computer from performing its assigned task(s).

Once a working, minimised operating environment is established, by research, experimentation, and testing this is the environment that should be used for that type of server (a server running the same application, or suite of applications) from now on, until the application requirements or the operating environment changes.

Because the process of establishing the correct minimised operating environment can be lengthy and difficult at times, it makes sense that the results of successful efforts in this regard should be documented. The documented minimised operating environment for that particular application or suite of applications becomes the “Operating Environment Profile” for that type of server and should be followed consistently while it remains valid.

6.2 Reference Sites for Known Good Base Profiles

To establish the correct minimised operating environment for a particular type of server check the relevant operating environment and application vendor web sites and other published vendor documentation. Establish if the applications that you need to run have details of the minimum operating environment components, as part of the published system requirements information, and if the operating environment vendor publishes guidelines on minimisation for security.

Often application vendors do not publish minimum operating environment component requirements. Some vendors of security products do provide this information, as they are aware that their customers are trying to create secure environments, and thus will often be making efforts to enhance the security of the underlying operating environment.

One example of a vendor providing a good level of detail to assist the process of operating environment minimisation for security is Sun Microsystems, and the detail they provide for their SunScreen firewall product. They provide the main software group (or installation type) to support the product, and then provide fine grain package-by-package information on any packages required in addition to the software group that they recommend[6]. This sort of information, while very valuable,

---

[6] The Solaris Operating System software is organised into Software Groups (Core, End User System Support, Developer System Support Entire Distribution and Entire Distribution Plus OEM Support). The groups are composed of clusters, which are logical groupings of software packages. Each software package is a functional group of files and directories.

does not mean that the process of operating environment minimisation for security for a particular product begins and ends at the vendor's documentation. While the information provides a good level of detail about what is needed, it will also normally include many components that are not essential and are potentially exploitable. What this vendor data does do is provide an excellent base profile as a starting point for fine grain minimisation. For the documentation of the vendor recommended minimum operating environment components for SunScreen please see the web address below: -

- http://docs.sun.com/ab2/coll.557.2/SSCRNINSTALL/@Ab2PageView/425?DwebQuery=sunscreen&oqt=sunscreen&Ab2Lang=C&Ab2Enc=iso-8859-1
  SunScreen 3.1 Installation Guide published by Sun Microsystems.

In some cases people or organisations that use a vendor's products will do research on minimum operating environment component requirements and publish this on their web sites to assist others in their efforts in regard to the same products. Often, even if a vendor publishes minimum operating environment component requirement information, the detail level of industry specialists and enthusiasts will be higher and more practical, because often they are more experienced in adapting a product to diverse real world situations than a vendor development or engineering team.

An example of an industry specialist who shares his knowledge and experience with a vendor's product with the world is Mr Lance Spitzner. Lance has published a number of papers that offer very practical, useful information and advice about the Firewall-1/VPN-1 product range from Checkpoint Software Technologies. Lance has also published very detailed papers on operating hardening, including minimisation for a computer that will be running the Firewall-1/VPN-1 products. For the Solaris operating environment his profiles include the software group (or installation type), additional packages required, and a list of unneeded packages that can be removed from the chosen software group. This is a high enough quality effort at operating environment minimisation for security on the Solaris platform that Lance's firewall profiles can often serve as a base profile or starting point for other similar gateway based network appliances and even Solaris based servers in general.

Links to the minimised operating environment profiles published by Lance are provided below: -

- Lance's "Armoring Solaris" paper updated 19 August, 2001
  http://www.enteract.com/~lspitz/armoring.html
  - http://www.enteract.com/~lspitz/core6.txt
    Core Installation of Solaris 2.6 for FW 4.0
  - http://www.enteract.com/~lspitz/core7.txt
    Minimum installation of Solaris 2.7 for Checkpoint FW-1 4.1
- http://www.enteract.com/~lspitz/armoring2.html
  Lance's "Armoring Solaris: II" paper updated 05 November, 2001 focused on minimisation of Solaris 8 (64-bit), running Firewall-1/VPN-1 NG

At the outset of this section it was suggested that you should check to see if the operating environment vendor publishes guidelines on minimisation for security.

Operating environment vendors may publish this type of information either in their documentation for the operating environment, or more likely in frequently asked question lists (FAQs), or as security articles in best practice guides for their operating environment.

A good example of an operating environment best practice guide series is the "Sun BluePrints" program from Sun Microsystems. The "Sun BluePrints" program provides (according to it's home page) "in-depth technical information on Best Practices using Sun Solutions"[7]. Included in this program are many articles related to security, and one that is particularly relevant to the topic of this paper is a guide to operating environment minimisation for security on the Solaris platform. This paper provides a methodology that can be followed to achieve a minimised Solaris operating environment, and as a case study applies the methodology to a Solaris computer running the "iPlanet Web Server" also from Sun Microsystems. The paper contains a good base profile for Solaris 2.6, Solaris 7 and Solaris 8 running "iPlanet Web Server". This article is of high enough quality that in additional to providing a profile for running "iPlanet Web Server" on Solaris, it could also serve as a base profile or starting point for other similar applications and even for Solaris based servers in general. Links to the "Sun BluePrints" program home page, and the specific "Solaris™ Operating Environment Minimization for Security: A Simple, Reproducible and Secure Application Installation Methodology" paper are shown below: -

- http://www.sun.com/blueprints/
  Sun BluePrints™ programs home page
  - http://www.sun.com/blueprints/1299/minimization.pdf
    Solaris™ Operating Environment Minimization for Security: A Simple, Reproducible and Secure Application Installation Methodology (December 1999) -by Keith Watson and Alex Noordergraaf
  - http://www.sun.com/blueprints/1100/minimize-updt1.pdf
    Updated Solaris™ Operating Environment Minimization for Security: A Simple, Reproducible and Secure Application Installation Methodology - Updated for Solaris 8 Operating Environment (November 2000) -by Alex Noordergraaf

6.3 Creating New Profiles by Research and Testing

A good base profile should be selected, as described above, either using reference sites such as those given previously; or in cases were the application bears no resemblance to any application types for which known good profiles have been created, by selecting the appropriate software group (or installation type). If just a software group is to be selected, the "core" group is the preferred starting point, if possible, to achieve the maximum minimisation.

Once the base profile or starting point is established the hardware and software

[7] Copyright 1994-2002 Sun Microsystems, Inc
Sun BluePrints Program Home Page
http://www.sun.com/blueprints/
(accessed 24th March 2002)

requirements for the computer and the applications it must run must be considered. Work out what types of hardware will be used (for example, only PCI network cards), what network and internal services are needed by the application (for example, does the application use FTP but not supply a daemon). Understanding thoroughly the hardware and software support requirements for the computer and the application will help you make educated guesses about which hardware support software and services software will need to added to, or can be taken away from, your base profile.

Establishing the hardware and software support requirements as described above can take some research effort searching vendor sites, mailing lists and using search engines. Often, even after thorough research, extra support requirements will only become evident during the testing stage, which is discussed next.

Once a candidate profile has been established (including removing all components not required, and adding additional required components to the base profile) the operating environment profile should be installed on the computer, then the application(s) should also be installed, and testing should then be performed.

During the testing stage error messages on the console, and in system and application logs should be analysed to determine if the minimised environment is causing them. The assistance of an expert in the application(s) will be needed to ensure that testing is thorough, and to assist in the interpretation of error messages and aberrant application behaviour. Access to application vendor support during the testing phase is a very valuable, because testing on a minimised operating environment may produce behaviour in applications that even the vendor has never seen before, and the vendor may be the only one that can help you resolve the issues.

As problems are identified, additional software components may have to be added to the profile. If your application expert(s) have expertise in the “hardening” (configuring for enhanced security) of the application(s), you may also be able to identify other operating environment components that are not used by the application(s) after “hardening”, and which can safely be removed.

As a result of all this research and testing, in cooperation with your application expert(s) and perhaps even vendors, you should now have a working, minimised operating environment profile.

[8] The Unix “grep” utility is a global regular expression matching tool that in this case is used to search a file for a particular string of characters.

7 TIP: When working on a Solaris system, if you are missing a library or other file that your application requires, you should have a copy of the "/var/sadm/install/contents" file from a system of the same version with the full distribution installed. Running a "grep"[8] for the file name you are missing against the contents file will yield the name of the Solaris package to which the file belongs, if the file is part of Solaris.

Performing Operating Environment Minimisation for Security

7.1 General

When installing almost any operating environment, there are three main ways to implement our minimised profile; in many cases a combination of at least two of these is required to completely achieve our goal.

7.1.1 Implementing During the Install

The first and most basic method of installation is the standard interactive install, with minimal automation. This is the normal method, in which you boot a machine from an operating environment installation floppy or CD and then answer the installation questions interactively to produce a system that matches your requirements as closely as possible. Depending on your operating environment you will be able to perform a greater or lesser amount of minimisation at this stage: some operating environments offer very little in the way of granular choices, while others allow a high level of control. Almost all interactive installs will place some restrictions on the installer's choices, which will prevent complete implementation of our minimised profile, or at least make it quite tedious and time consuming to implement.

The advantage with this method is that it is easy, because most people can boot off a CD or floppy and answer the installation questions without too much trouble.

The disadvantage is that, because this method is designed to take the difficult decisions out of the installer's hands, it is often difficult or impossible to specify to the fine grain level required exactly what we want our minimised system to be include and exclude.

7.1.2 Implementing with Automated Installations

The second method is the automated installation. Many modern operating environment vendors or third party utility vendors provide automated install methods; many of these offer both a "hands-off" fully automated method to execute the actual installation procedure, and various types of "cloning" methods. The "cloning" method simply allows you to create a model system manually, and "clone" copies of that system as required, either manually or automatically setting the new system's individual characteristics.

Some examples of the fully automated installation methods are the "JumpStart"

[8] The Unix "grep" utility is a global regular expression matching tool that in this case is used to search a file for a particular string of characters.

installation technique from Sun Microsystems which is used to automate Solaris installations, and the "Kickstart" installation method used to automate installations for some types of Linux systems. Web links are given below for these two installation technologies.

- http://www.redhat.com/docs/manuals/linux/RHL-7.1-Manual/custom-guide/ch-kickstart2.html
  Red Hat Linux 7.1: The Official Red Hat Linux Customization Guide - Chapter 2. Kickstart Installations

- http://docs.sun.com/ab2/coll.214.7/SPARCINSTALL/@Ab2TocView?Ab2Lang=C&Ab2Enc=iso-8859-1
  Solaris 8 Advanced Installation Guide (JumpStart)

Some examples of the "cloning" methods are the "Norton Ghost" tool from Symantec, which allows the "cloning" of Intel compatible computers at the disk or partition level, and the "Web Start Flash" tool from Sun Microsystems to allow the "cloning" of Solaris based computers. Web links are given below for these two "cloning" technologies.

- http://www.sun.com/solaris/webstartflash/
  Web Start Flash Complete Systems Replication by Sun Microsystems

- http://www.symantec.com/region/au_nz/product/ghost/ghost_corp/
  Symantec Ghost Corporate Edition 7.5
  for Windows 98, Windows NT 4.0, Windows 2000, Windows XP

The advantage with fully automated installation or "cloned" installations is that you can normally fully specify the details of the system that you require, thus producing your minimised operating environment in one pass. Because once you create an automated installation profile or a "cloning" image you can re-use it over and over, this method is ideal for large organisations or engineering firms that need to create a large number of consistent, minimised computers.

The disadvantage of fully automated installation or "cloned" installations is that it is often difficult and time consuming to set up, and maintain. In some cases, to create an automated installation, system staff will need expensive training or external engineers/consultants will have to be hired, because creating these systems can require specialised skills. For a small organisation that does not need to install a large number of minimised systems a business/judgement call needs to be made as to whether the expense, is justified by the time saved performing automated installations.

Note: I have only considered the value of the fully automated installation or "cloned" installation systems in regard to installing systems with minimised operating environments. Obviously these types of systems can increase the speed and consistency of operating environment (and sometimes application) installations of all types, and thus may be well worth the effort, even if their value for minimisation for security does not justify them.

7.1.3 Implementing After a Basic Installation

The third installation method is a natural extension of the first method, given in

section 7.1.1. It is often difficult to achieve complete implementation of our chosen minimised profile during the interactive install, because restrictions in interactive installation procedures often make it very difficult or impossible to choose each and every component of the operating environment that we wish to include or exclude.

The obvious solution to our problem is to find a simple path through the choices offered by the installation procedure, to produce a system that is as close as is easily achievable to the exact system we want, and to just “fix” the rest after the installation is completed. This involves performing a basic install, then removing all the components we do not want in our profile manually, and adding any components that we require that were not included in the basic install manually.

The advantage with this method is that it is easy and very achievable, and no advanced skills required. With some simple scripting, or perhaps using tools included with some operating environments for applying security policy, a lot of the tedious work of adding and removing can be eliminated or at least reduced. This method is ideal for a small organisation with only occasional “one off” requirements to install computers with minimised operating environments.

One other major advantage to using this method, at least at first, is that it’s very visible “hands-on” approach gives one a very good idea of what is happening during the minimised install process. Knowledge gained using this method may be very useful should advanced automated methods, where the processes are invisible to the user, be adopted later.

The disadvantage of this method is that it does tend to be time consuming and a little tedious, especially the first few times you do it and the post installation tasks by default are very manual. This method would be an impractical choice for a large organisation that needed to install many computers with similar minimised operating environments, as the install time for a single computer is excessive.

7.2 Solaris in Detail

As the reader may have gathered by now, my special area of interest is the Solaris operating environment from Sun Microsystems. I have decided in conclusion to devote some attention to the specifics of implementing a minimised operating environment on Solaris. Up until this point I have tried to stay **reasonably** platform independent in my comments and observations, but I am sure my preferences did sometimes shine through.

A word on packages verses files in regard to Solaris minimisation for security. I choose to make the package the smallest functional unit that I will use for minimisation mainly because of the way patches and packages work in Solaris.

Many people rightly point out that it is more secure to just install required library files, and other individual file level required components by copying them from the CD to their correct path. This is true, but creates a problem with patching, and sometimes with package installs.

Solaris patches are created to patch a specific package: if the package they are created to patch does not exist then the patch will not apply.

Most Solaris software ships in package format; on installation the “pkgadd” program

will check for other prerequisite packages: if the whole prerequisite package is not properly installed you will have a little difficulty installing the package (although you should be able to get around it).

The major problem all this produces is that if you choose to install individual files, rather than the whole packages they belong to, these files will not be patched by the normal installation of Solaris patches, so any bugs and vulnerabilities will not be fixed.

You could work around the patching problems by noting the packages that the files would normally belong to and manually checking new patches for references to these packages, then checking the ones that are applicable for updates to your individual files and applying them manually. I feel that in all but the most extreme cases this approach would be an unjustified effort compared to the increase in security it brings.

7.2.1 Implementing During the Install

To install a minimised Solaris operating environment (OE) during the install you will need to select the software group that you require (often core). Choose the "customise" option, for all software groups including "core" add additional required packages, if you choose any software group other than "core" you should then remove all unnecessary packages (you cannot remove any packages from the "core" software group).

You should have already identified the packages to add and remove using the profile creation methods discussed earlier in this paper (sections 6.2 and 6.3 provide details on this). Once all your package choices are made continue as normal with the install.

If you chose the core software group and were unable to remove packages that you had decided should be removed from you minimised profile you will still need to remove these post installation with the "pkgrm" command. You may encounter some difficulty with package dependencies while you are testing your profile; this may require you to adjust your profile or perhaps just remove packages in a different order (once you ensure you have solved any important dependency issues) if you wish to avoid the error messages.

7.2.2 Implementing with JumpStart

By far my preferred way to create a minimised Solaris OE is using the JumpStart automated installation technology. This technology allows one to specify the exact minimised profile desired, and to roll out many minimised Solaris systems quickly, consistently and in one pass. I would suggest the use of the "/etc/bootparams", "/etc/ethers" and "/etc/hosts" files in preference to using NIS or NIS+ maps for JumpStart especially if you are new to JumpStart.

A number of very good manuals, books and papers exist that discuss JumpStart in detail. One book in particular "JumpStart™ Technology: Effective Use in the Solaris ™ Operating Environment" by John S. Howard and Alex Noordergraaf also discusses Solaris OE minimisation for security, and Solaris OE hardening using the "Solaris Security Toolkit (formerly JASS)", within the JumpStart installation

architecture.

Links to information about the manuals, books and papers mentioned above are provided below: -

- http://docs.sun.com/ab2/coll.214.7/SPARCINSTALL/@Ab2TocView?Ab2Lang=C&Ab2Enc=iso-8859-1
  Solaris 8 Advanced Installation Guide (JumpStart)

- http://www.sun.com/blueprints/pubs.html
  "JumpStart™ Technology: Effective Use in the Solaris™ Operating Environment" By John S. Howard and Alex Noordergraaf

- http://www.sun.com/blueprints/0401/BuildInf.pdf
  Building a JumpStart™ Infrastructure (April 2001) - by Alex Noordergraaf

- http://www.sun.com/blueprints/1100/jssec-updt1.pdf
  JumpStart™ Architecture and Security Scripts for the Solaris™ Operating Environment - Part 1 - by Alex Noordergraaf and Glenn Brunette

- http://www.sun.com/blueprints/1100/jssec2-updt1.pdf
  JumpStart™ Architecture and Security Scripts for the Solaris™ Operating Environment - Part 2 - by Alex Noordergraaf and Glenn Brunette

- http://www.sun.com/blueprints/1100/jssec3-updt1.pdf
  JumpStart™ Architecture and Security Scripts for the Solaris™ Operating Environment - Part 3 - by Alex Noordergraaf and Glenn Brunette

7.2.3 Implementing After a Basic Cluster Install

As was described in section 7.2.1 it may not be possible, and is almost always quite tedious, to fully implement our minimised Solaris OE completely during the normal interactive installation process. As described in the platform independent information in section 7.1.3, often a compromise is needed where a basic install is performed and then some post install adjustments are made.

In Solaris a basic install of the chosen software group closest to your minimised profile should be performed (often core). No packages should be added or removed at this time, simplifying the install process significantly.

Once the system is installed use "pkgrm" to remove all unnecessary packages and "pkgadd" to add any additional required packages not included in the chosen software group, the minimised profile is now implemented.

As this process can be somewhat labour intensive and boring, I would suggest creating a small script that performs the needed package removal commands, then prompts for the correct Solaris CD(s) as required, mounts them, and performs the needed package add commands.

To further simplify this process it is possible to prevent the "pkgadd" and "pkgrm" commands from prompting the user for confirmation while removing or adding packages, thus avoiding the need to constantly interact with your script while the post-install adjustments are taking place.

The syntax that is required to prevent the “pkgrm” and “pkgadd” commands from prompting is shown below. The response file (called “response.file” in the example) referred to must exist but should be an empty file. The admin file (called “admin.file” in the example) referred to should match the example admin file shown.

- `pkgrm -n -a /PATH_TO_ADMIN_FILE/admin.file PACKAGEname`

- `pkgadd -n -a /PATH_TO_ADMIN_FILE/admin.file -r /PATH_TO_EMPTY_RESPONSE_FILE/response.file -d /PACKAGE_PARENT_DIRECTORY PACKAGEname`

- `The correct contents of the admin file are shown below:`

```
basedir=default
mail=
runlevel=quit
conflict=nocheck
setuid=nocheck
action=nocheck
partial=nocheck
instance=unique
idepend=nocheck
rdepend=nocheck
space=quit
```

More advanced information concerning automation of the package adding and package removal processes can found in the man pages, and the “Application Packaging Developer's Guide”, at the following web addresses: -

- http://docs.sun.com/ab2/coll.40.6/REFMAN1M/@Ab2PageView/191687?Ab2Lang=C&Ab2Enc=iso-8859-1
  “pkgrm” manual page for Solaris 8 by Sun Microsystems

- http://docs.sun.com/ab2/coll.40.6/REFMAN1M/@Ab2PageView/190290?Ab2Lang=C&Ab2Enc=iso-8859-1
  “pkgadd” manual page for Solaris 8 by Sun Microsystems

- http://docs.sun.com/ab2/coll.45.13/PACKINSTALL/@Ab2TocView?Ab2Lang=C&Ab2Enc=iso-8859-1
  Application Packaging Developer's Guide by Sun Microsystems

8 References

- Turolla, Stefano
  Detect Host Vulnerability - Local Host Security Tools
  Last Modified on 25/4/2000, 15:53:48
  http://wwwinfo.cern.ch/dis/security/general/tools/detect.html

- Copyright 1994-2002 Sun Microsystems Inc., 901 San Antonio Road, Palo Alto CA 94303 USA
  SunScreen 3.1 Installation Guide
  http://docs.sun.com/ab2/coll.557.2/SSCRNINSTALL/@Ab2PageView/425?DwebQuery=sunscreen&oqt=sunscreen&Ab2Lang=C&Ab2Enc=iso-8859-1
  (accessed 24th March 2002)

- Watson, Keith and Noordergraaf, Alex
  Solaris™ Operating Environment Minimization for Security: A Simple,Reproducible and Secure Application Installation Methodology (December 1999)
  http://www.sun.com/blueprints/1299/minimization.pdf
  (accessed 24th March 2002)

- Noordergraaf, Alex
  Updated Solaris™ Operating Environment Minimization for Security: A Simple, Reproducible and Secure Application Installation Methodology - Updated for Solaris 8 Operating Environment (November 2000)
  http://www.sun.com/blueprints/1100/minimize-updt1.pdf
  (accessed 24th March 2002)

- Galvin, Peter Baer
  Solaris Security FAQ Unix Insider 1/1/01
  http://www.itworld.com/Comp/2377/security-faq/index.html
  (accessed 24th March 2002)

- Spitzner, Lance
  Armouring Solaris by Last Modified: 19 August, 2001
  http://www.enteract.com/~lspitz/armoring.html
  (accessed 24th March 2002)

- Spitzner, Lance
  Armouring Solaris II by Last Modified: 05 November, 2001
  http://www.enteract.com/~lspitz/armoring2.html
  (accessed 24th March 2002)

- Powell, Brad M., Farmer, Dan, and Archibald, Matthew
  Titan Hardening Tool by 24 March 2002
  http://www.fish.com/titan/
  (accessed 24th March 2002)

- Noordergraaf, Alex (Copyright 1994-2002 Sun Microsystems Inc.)
  The Solaris Security Toolkit
  http://www.sun.com/security/jass/
  (accessed 24th March 2002)

- Red Hat Linux 7.1: The Official Red Hat Linux Customization Guide - Chapter 2. Kickstart Installations
http://www.redhat.com/docs/manuals/linux/RHL-7.1-Manual/custom-guide/ch-kickstart2.html
(accessed 24th March 2002)

- Copyright 1994-2002 Sun Microsystems Inc., 901 San Antonio Road, Palo Alto CA 94303 USA.
Solaris 8 Advanced Installation Guide (JumpStart)
http://docs.sun.com/ab2/coll.214.7/SPARCINSTALL/@Ab2TocView?Ab2Lang=C&Ab2Enc=iso-8859-1
(accessed 24th March 2002)

- Copyright 1994-2002 Sun Microsystems, Inc.
Web Start Flash Complete Systems Replication by Sun Microsystems
http://www.sun.com/solaris/webstartflash/
(accessed 24th March 2002)

- Howard, John S. and Noordergraaf, Alex (Copyright 1994-2002 Sun Microsystems, Inc.)
JumpStart™ Technology: Effective Use in the Solaris™ Operating Environment
http://www.sun.com/blueprints/pubs.html
(accessed 24th March 2002)

- Noordergraaf, Alex
Building a JumpStart™ Infrastructure (April 2001)
http://www.sun.com/blueprints/0401/BuildInf.pdf
(accessed 24th March 2002)

- Noordergraaf, Alex and Brunette, Glenn
JumpStart™ Architecture and Security Scripts for the Solaris™ Operating Environment - Part 1 (November 2000)
http://www.sun.com/blueprints/1100/jssec-updt1.pdf
(accessed 24th March 2002)

- Noordergraaf, Alex and Brunette, Glenn
JumpStart™ Architecture and Security Scripts for the Solaris™ Operating Environment - Part 2 (November 2000)
http://www.sun.com/blueprints/1100/jssec2-updt1.pdf
(accessed 24th March 2002)

- Noordergraaf, Alex and Brunette, Glenn
JumpStart™ Architecture and Security Scripts for the Solaris™ Operating Environment - Part 3 (November 2000)
http://www.sun.com/blueprints/1100/jssec3-updt1.pdf
(accessed 24th March 2002)

- © 1995-2002 Symantec Corporation.
Symantec Ghost Corporate Edition 7.5
http://www.symantec.com/region/au_nz/product/ghost/ghost_corp/
(accessed 24th March 2002)

- Sun Microsystems Copyright 1994-2002 Sun Microsystems Inc., 901 San Antonio Road, Palo Alto CA 94303 USA.
  "pkgrm" manual page for Solaris 8
  Solaris 8 Reference Manual Collection
  man pages section 1M: System Administration Commands
  http://docs.sun.com/ab2/coll.40.6/REFMAN1M/@Ab2PageView/191687?Ab2Lang=C&Ab2Enc=iso-8859-1
  (accessed 24th March 2002)

- Sun Microsystems Copyright 1994-2002 Sun Microsystems Inc., 901 San Antonio Road, Palo Alto CA 94303 USA.
  "pkgadd" manual page for Solaris 8
  Solaris 8 Reference Manual Collection
  man pages section 1M: System Administration Commands
  http://docs.sun.com/ab2/coll.40.6/REFMAN1M/@Ab2PageView/190290?Ab2Lang=C&Ab2Enc=iso-8859-1
  (accessed 24th March 2002)

- Copyright 1994-2002 Sun Microsystems Inc., 901 San Antonio Road, Palo Alto CA 94303 USA.
  Application Packaging Developer's Guide
  http://docs.sun.com/ab2/coll.45.13/PACKINSTALL/@Ab2TocView?Ab2Lang=C&Ab2Enc=iso-8859-1
  (accessed 24th March 2002)

- Sun Microsystems Inc., Palo Alto, CA, 2000
  Solaris 8 Advanced Installation Guide (JumpStart)
  http://docs.sun.com/ab2/coll.214.7/SPARCINSTALL/@Ab2TocView?Ab2Lang=C&Ab2Enc=iso-8859-1
  (accessed 24th March 2002)

- Copyright 1994-2002 Sun Microsystems, Inc
  Sun BluePrints Program Home Page
  http://www.sun.com/blueprints/
  (accessed 24th March 2002)

Last Updated: May 29th, 2017

Upcoming Training

SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS Rocky Mountain 2017 - SEC401: Security Essentials Bootcamp Style	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
Community SANS Portland SEC401	Portland, OR	Jun 12, 2017 - Jun 17, 2017	Community SANS
SANS Secure Europe 2017	Amsterdam, Netherlands	Jun 12, 2017 - Jun 20, 2017	Live Event
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Minneapolis SEC401	Minneapolis, MN	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Phoenix SEC401	Phoenix, AZ	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Munich Summer 2017	Munich, Germany	Jul 10, 2017 - Jul 15, 2017	Live Event
Mentor Session - SEC401	Macon, GA	Jul 12, 2017 - Aug 23, 2017	Mentor
Mentor Session - SEC401	Ventura, CA	Jul 12, 2017 - Sep 13, 2017	Mentor
Community SANS Atlanta SEC401	Atlanta, GA	Jul 17, 2017 - Jul 22, 2017	Community SANS
Community SANS Colorado Springs SEC401	Colorado Springs, CO	Jul 17, 2017 - Jul 22, 2017	Community SANS
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
SANSFIRE 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
Community SANS Charleston SEC401	Charleston, SC	Jul 24, 2017 - Jul 29, 2017	Community SANS
Community SANS Fort Lauderdale SEC401	Fort Lauderdale, FL	Jul 31, 2017 - Aug 05, 2017	Community SANS
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event