Solaris Patch Management: Recommended Strategy

A White Paper

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No. 817-0574-12
Revision 3, August 2004

Contents

Introduction

Applying patches and updating your system from one software release to another have evolved into what can be complex, time-consuming processes. Because software becomes more layered with each new patch that is applied, the strategy for updating software needs to be revisited and clarified. IT professionals also need to manage their systems so that the most appropriate software available is installed.

In addition, the number of hacker exploitations of security issues has risen alarmingly in recent years, even as the average time between a CERT advisory and the appearance of programs exploiting an issue has dropped. Knowing what software updates and patches to use and when to apply them have become critical issues for IT professionals.

More and more customers are turning to their vendors for help with selecting and applying patches. Some IT administrators have implemented a policy of attempting to keep current on all patches, while some industry experts recommend minimizing the number of patches applied.

This white paper is intended for Sun customers and discusses Sun's recommended strategy for managing patches and software updates. This paper also provides recommended practices and a process for maintaining properly patched and updated software. It discusses the risks, costs, and timing of updating your software, and provides references to Sun's patch-related tools.

Note: *Patches* are traditionally associated with bug fixes to known problems, such as software functionality, performance, reliability, data integrity, or security. Sometimes patches contain new features or enhancements to a particular software release. *Software updates* include patches for bug fixes, as well as new features or enhancements, support on new hardware, and patches that enable other patches to be applied. Much of the information in this paper is applicable to both patches and software updates.

Recommended Strategy for Updating Software

Sun's recommended strategy for updating software includes these practices:

- Analyzing the need to apply patches or update your software based on risk, cost, availability, and timing
- Minimizing change to your environment whenever possible
- Addressing Sun℠ Alert notifications and other critical issues as soon as possible
- Only making other changes to your environment to address known problems
- Maintaining your environment as current as appropriate for your business and application needs

To minimize risk and cost, and to maintain the highest level of overall availability for your systems, Sun recommends minimizing the amount of change that you introduce into your environment whenever possible. Sun also suggests that you address critical security, data loss, and availability issues as quickly as possible. To this end, Sun supports the use of its Recommended Patch Clusters and Sun Alert Packs. Beyond addressing critical issues, however, Sun's analysis of availability and patch management data shows that additional application of patches *decreases* the overall total availability of systems, counting both planned and unplanned downtime.

The *Recommended Patch Cluster* reflects the *current* version of all of the patches required to address all Sun Alert issues. Some customers have adopted a policy of periodically applying the current Recommended Patch Cluster to a system. While this strategy does address all Sun Alert issues, it also introduces more change to the system than is necessary. Similarly, reapplying the current Recommended Patch Cluster on a scheduled basis is also not necessary.

The *Sun Alert Pack* reflects the *lowest* revision of all of the patches necessary to address all Sun Alert issues, and the least possible change to your system while still addressing all Sun Alert issues.

Therefore, beyond ensuring that critical issues are addressed, Sun believes that you should apply patches only to address specific issues or needs. You should not apply patches merely to keep current. There is no benefit to applying the latest revisions of patches without understanding whether those patches provide any value.

For a new installation, Sun advises that you support your applications and business needs with the most current release of the Solaris™ Operating System (Solaris OS) that will work in your environment. Once that is done, you should only need to update your software and apply patches on an as-needed basis. Some Sun customers might have access to the Enterprise Installation Standards (EIS) methodology, which combines an installation checklist with the current Recommended Patch Set.

The following subsection describes these recommendations and provides information about how each recommendation might be implemented.

Recommended Practices

Analyze the Need to Update Your Software

Introducing change into an environment includes a certain amount of risk, however minimal. Analyze your systems' environments with regard to risk exposure, acceptable cost of maintenance, required availability, and the timing of updates. Each deployment environment, system, or application might have unique requirements.

Minimize Change to Your Environment

Once a policy has been established for each of the unique environments that you have identified, try to minimize change within each of those environments. In addition to the risk associated with implementing a change, Sun's data shows that availability drops as systems are taken offline for patches to be applied. However, you will most likely want to have the latest security patches on your Internet-facing web servers, for example, even at the expense of downtime and maintenance costs. Thus, “minimize change” is relative to the deployment environment.

Address Sun Alert Notifications

Sun Alert notifications are available to contract customers and are one way to learn about security issues.

A *Sun Alert notification* describes specific hardware and product software issues that might pose a risk to your computing environment and productivity. A Sun Alert notification informs you about a potential problem in three areas: data loss, security, and availability. Sun creates a Sun Alert notification when it believes that the alert has a wide-ranging applicability to its customer base.

Apply any applicable patches indicated by a Sun Alert notification as quickly as possible. Note that, in some cases, there is also a nonpatch resolution to the Sun Alert, for example, disabling access to a port. Such a resolution might be less intrusive and less costly than applying a patch.

Normally, Sun Alert notifications fall outside of regularly scheduled patch management processes and need to have their own patch management process applied. The process will depend on your environment and customized processes. Before you decide not to apply a patch until the next patch maintenance cycle, give careful consideration to the potential risks.

All operating environment-related patches that resolve problems indicated in Sun Alert notifications are included in Recommended Patch Clusters and Sun Alert Packs, along with any patches on which they depend. Non-operating environment-related patches that resolve problems detailed in Sun Alert notifications must be downloaded from the SunSolve Online℠ web site and applied, if appropriate for your system environment.

Check the SunSolve℠ site regularly for newly released Sun Alert notifications and relevant information about patches. Go to http://www.sun.com/sunsolve.

Appropriate technical personnel should subscribe to receive periodic Sun Alert summary reports. To subscribe to get these reports, go to http://sunsolve.sun.com/pub-cgi/show.pl?target=salert-notice.

Address Known Problems

While patches that resolve Sun Alert issues are important to many customers, most Sun patches address problems that are only of interest to a small group of customers, such as those who have a unique application suite or environment.

If a problem should occur and the solution calls for applying a patch, carefully consider the current state of your system in the context of your business environment. Ask yourself the following questions:

- Is the system currently down and inoperable?
- Is the system operating, but having errors and hindrances that are of concern?
- Is the maintenance cycle soon enough that you can risk waiting to apply the patch?

Keep Your Environment Current According to Your Business Needs

The best way to begin maintaining your environment is to install the most current operating system that is consistent with your business and hardware requirements. The benefits include the following:

- Improved overall system stability
- Latest improvements in software development, including bug fixes
- Improved system security
- Enhanced system performance
- Fewer patches to manage

If you are not using the most recent version of the Solaris OS, it is important that you run the most current release of the Solaris OS that you do use, as long as it meets your business needs. For example, if you use the Solaris 9 OS, the most current release (as of this writing) would be Solaris 9 4/04. This release would contain all of the patches available for the Solaris 9 OS at the time that the update was created. For a prior version of the Solaris OS, use the last release that exists for that version of the operating system.

Note: Software updates are released periodically only for the current version of the Solaris OS. The last release of an older Solaris OS can be obtained through your Sun sales representative.

Patches

This section discusses the following patch-related topics:

- Patch definition
- Patch types
- Patch interrelationships
- Patch delivery mechanisms

Patch Definition

A *patch* is an accumulation of fixes to a known problem or to a potential problem within the operating system or other supported software. A patch can also provide a new feature or an enhancement to a particular software release. A patch consists of files and directories that replace or update existing files and directories.

Most Solaris patches are delivered as sparse package patches. A *sparse package*

patch contains only those packages whose objects have been altered since the packages first shipped to customers. Sparse package delivery contrasts with Red Hat Package Manager (RPM) on Linux. For example, RPMs replace entire components of the operating environment. By using sparse packages, Sun can deliver patches that alter your environment the least.

Each patch is identified by a patch identification number (`patch_id`). The patch ID consists of a six-digit base identifier and a two-digit revision number of the form *xxxxxx-yy*.

Patches are also cumulative. Later revisions contain all of the functionality delivered in previous revisions. For example, patch 123456-02 contains all the functionality of patch 123456-01, plus the new bug fixes or features that have been added in Revision 02, as described in the patch README file.

Patch Types

Patches are of two general types: standard and nonstandard.

Standard Patches

Standard patches support automatic backout when necessary, and include the types of patches described in this section.

Generic Patch

A generic patch contains bug fixes or new features. While a generic patch has no special identification in the patch ID, it might be identified with a select keyword in the Keywords field of the patch `README.`*patch_id* file, for example:

- `security` - Identifies patches that were added to the security patch cluster.
- `y2000` - Identifies patches that were added to the Y2K patch cluster.
- `encryption` - Identifies patches that limit distribution through the SunSolve web site based on current export laws.
- `point_patch` - Identifies patches that limit distribution to specific customers by using a special directory on the SunSolve site.
- `kernel` - Identifies patches that ensure that this patch, and any patches on which it depends, are always added to the Recommended Patch Cluster.

Kernel Update Patch (KU Patch)

A KU patch updates the Solaris kernel and other core Solaris functionality. This patch is released on a regular schedule, not each time a new fix is introduced.

Restricted Patch (R-Patch)

A restricted patch is denoted with the prefix letter “R.” An R-patch causes any package that it modifies to be locked to prevent subsequent modification of the package by other patches.

Point Patch

A point patch is not for general use. This type of patch is accessible only to customers who have been provided with a specific point patch ID.

A point patch might contain fixes for a specific customer or even a specific system. These fixes are created on a branch of the source code tree, not folded back into the main source code tree. Typically, Sun considers a point patch to be appropriate only for the customers to whom the fix has been delivered. A point patch should be installed only after consultation with Sun support personnel and should be removed as soon as practical.

Nonstandard Patches

Nonstandard patches are not delivered in sparse package patch format.

Typical Nonstandard Patches

Many nonstandard patches are of the following types:

- Driver update patches for x86-based systems that must be released as diskette images
- Firmware/hardware patches, including OBP, controller, and disk firmware
- Flash PROM update patches that contain binary files for updating system firmware
- Some other non-operating system software patches
- Some product patches that might not be delivered in package format
  The patch README file provides the necessary installation instructions.

Temporary Patch (T-Patch)

A temporary patch is one that is built and is ready for testing, but has not yet completed that process. A T-patch might be made available to customers involved in active escalations to verify that the patch fixes the customers' problem.

A temporary patch is identified by a leading “T” in the patch ID, for example, T108528-14. The words “(Preliminary Patch - Not Yet Released)” appear after the patch ID on the first line of the patch README file.

When the patch has been verified and internal Sun patch testing has been

completed, the T-patch designation is removed from the patch ID and the patch README file, and the patch is then released on the SunSolve site.

Patch Interrelationships

The functionality delivered in a patch, be it bug fixes or new features, might have interrelationships with the functionality delivered in other patches. These interrelationships are determined by three fields in the package's `pkginfo` file:

- Patch dependencies (`SUNW_REQUIRES`)
- Patch accumulation and obsolescence (`SUNW_OBSOLETES`)
- Patch incompatibility (`SUNW_INCOMPAT`)

Patch Dependencies (`SUNW_REQUIRES`)

The functionality delivered in a patch might have a code dependency on the functionality delivered in at least one other patch. That is, one patch requires another patch to function correctly.

If the patches are otherwise unrelated, the dependency can be described as one patch being required by another patch. If a patch depends on one or more patches, the patch will specify which patches it requires in the `SUNW_REQUIRES` field in the `pkginfo` files in the patch's sparse packages.

The dependency requirement can only work one way. If Patch A requires Patch B, Patch B cannot require Patch A.

Because patches are cumulative, if Patch A requires Patch B-Revision *xx*, any revision of Patch B greater than or equal to Revision *xx* will also satisfy the requirement. That is, there is an implicit, later revision associated with any required patch revision that is specified.

The `SUNW_REQUIRES` field is used to specify straightforward, hard-coded dependencies between patches. If other types of dependencies exist, they are specified in the patch README file and can include the following:

- **Conditional dependencies:** Where a hard-coded patch dependency occurs only under specific conditions, for example, only if CDE 1.3 is installed on the target system.
- **Soft dependencies:** Where other patches are required to completely fix a particular bug, but the patch will otherwise work correctly without the other patches.

Patch Accumulation and Obsolescence (SUNW_OBSOLETES)

Sometimes bug fixes or new features cause two or more existing patches to become closely intertwined. Rather than specifying a patch dependency, it might be better to accumulate the functionality of the multiple patches into just one of the patches. The other patch or patches would then be obsolete.

The patch into which the other patch functionality is accumulated specifies one or more patches that it has obsoleted in the SUNW_OBSOLETES field in the pkginfo files in the patch's sparse packages. This declaration is called *explicit* obsolescence.

The patch accumulation can only work one way. That is, if Patch A accumulates Patch B, Patch A now contains all of Patch B's functionality. Patch B is now obsolete. No further revision of Patch B will be generated.

A later revision of a patch *implicitly* obsoletes earlier revisions of the same patch. Patches that are implicitly obsoleted are not flagged in the SUNW_OBSOLETES field. That is, there is no need for Patch A-Revision *xx* to explicitly obsolete Patch A-Revision *x*-1 with a SUNW_OBSOLETES entry in the pkginfo file in the sparse package.

Patch Incompatibility (SUNW_INCOMPAT)

On rare occasions, two patches are incompatible. For example, incompatibility might occur if one of the patches is a point patch. An incompatibility is specified in the SUNW_INCOMPAT field in the pkginfo file in the sparse package of one or both of the patches.

Patch incompatibility is two-way. If Patch A or Patch B specifies an incompatibility with the other patch, only one of the patches can be installed on the target system. For example, if Patch A is already installed on the target system and Patch B is incompatible with it, the patch install utility patchadd will not allow Patch B to be installed. If Patch B must be installed, Patch A must first be removed.

Patch Delivery Collections

Patches are often part of a set, as described in the following sections. Patches are at the SunSolve Patch Support Portal, http://www.sun.com/sunsolve/patches.

Recommended Patch Cluster

The Recommended Patch Clusters section on the SunSolve site contains a set of patches for various OS/architecture combinations. Sun recommends that you use the Recommended Patch Cluster appropriate for your needs.

Each patch in a Recommended Patch Cluster meets one or more of these criteria:

- **Addresses a Sun Alert issue:** That is, the patch addresses availability, security, or data loss issues.
- **Is required for the correct operation of the patch utilities:** Included are patches to the patch and package utilities themselves, or to utilities used by the patch and package utilities, such as `ksh`, `sh`, `csh`, `nawk`, `fgrep`, `installf`, and `removef`.
- **Is required by either of the preceding patches:** That is, the patch is specified in the `SUNW_REQUIRES` field in the `pkginfo` file of any other patch in the Recommended Patch Cluster.

Because Recommended Patch Clusters are updated frequently, check the SunSolve site regularly for updates.

Sun Alert Pack

A Recommended Patch Cluster for a given operating system *consolidates* the current revisions of all of the patches required to address all of the patch-related Sun Alert notifications. However, a Sun Alert Pack for a given operating system contains the lowest revision of all of the patches needed to address all Sun Alert issues.

The criteria for the Sun Alert Pack are basically the same as for the Recommended Patch Cluster. Each patch meets one or more of these criteria:

- **Addresses a Sun Alert issue:** That is, the patch addresses availability, security, or data loss issues.
- **Is required for the correct operation of the patch utilities:** Included are patches to the patch and package utilities themselves, or to utilities used by the patch and package utilities, such as `ksh`, `sh`, `csh`, `nawk`, `fgrep`, `installf`, and `removef`.
- **Is required by either of the preceding patches:** That is, the patch is specified in the `SUNW_REQUIRES` field in the `pkginfo` file of any other patch in the Sun Alert Pack.

Because Sun Alert Packs are updated as needed, check the SunSolve site regularly for updates.

Security T-Patches

The Security T-Patches section on the SunSolve site provides early access to patches that address security issues. The patches are still in T-patch stage, that is, they have not completed the verification and patch testing process. The installation of Security T-patches is at the user's sole discretion and risk.

Information about the issues addressed by Security T-patches and possible workarounds is available through the Free Security Sun Alert data collection, which you can access from the Security T-Patches section at http://www.sun.com/sunsolve/patches.

Software Updates

A Solaris update is a completely new release image of the current release of the Solaris OS (currently Solaris 9) and only that release. A Solaris update contains new feature packages and the latest consolidated patches available for the current release of the Solaris OS at the time the Solaris update was created. Installing or upgrading your Solaris system is similar to the initial installation of the Solaris OS.

The patches in the Solaris update are freshbitted into the Solaris OS image. That is, the patches are pre-applied to the image and cannot be backed out.

Note: The Solaris update might contain a few special script patches that are used to overcome limitations in the freshbitting process. Because such script patches have no meaning outside of the Solaris update, they are not available for download as individual patches from the SunSolve site.

Note: Check the Recommended Patch Clusters section and the Sun Alert Pack section at the SunSolve site for patches that have been released since the content of a Solaris update was finalized. A time gap exists to allow for testing and production between the date the content was finalized and the release date.

Patch and Software Update Considerations

Patches and software updates must be managed so that the most appropriate software is installed. By following effective management techniques, you gain these system benefits:

- Increased availability
- Better performance
- Increased security
- Increased stability

When determining whether to update your software, you must consider cost, risk, availability, and timing. You must strike a balance among these factors for the systems and software in question before you decide to proceed.

User and environment requirements need to be carefully thought out and planned to establish the thresholds for these factors. Note that each factor can be different for each system and network.

- **Cost:** What your business can afford
- **Risk:** Exposure to and consequences of denial of service, for example
- **Availability:** Scheduling downtime, lengths of downtime, and consequences of unplanned downtime
- **Timing:** Determining which release to run and when to apply patches

In this section, Sun tries to provide you with as much information as possible about these factors so you can make an informed decision.

Cost

Several types of costs are associated with updating software: costs for manpower, to your business, and for planned downtime. Costs associated with backout and rework for withdrawn patches are relatively low due to the backout feature in most of Sun's patches and to Sun's relatively low rate of withdrawing patches.

On the other side of the cost equation are costs associated with not applying patches. These costs typically include investigative and corrective work to restore systems, and unplanned downtime for your business.

It might appear that the cost of updating software could be compared to that of

not updating it, and a cost threshold set where a "go–no go" decision could be made. However, cost is just one factor. Risk must also be considered.

If the risk of unplanned downtime is high and the cost is not prohibitive, the software should be updated. If the risk is low, the cost too high, or both are true, the software should not be updated. The specific threshold must be evaluated in each case.

Risk

There are risks to applying a patch, and risks to not applying one. The risk of applying a patch is the possibility of introducing unexpected consequences as a result of the change. The risk of not applying a patch is the possibility of not being able to use an application that the patch fixes. The decision to update software must be based on sound data and the requirements particular to the software installation under consideration.

Security issues tend to carry the largest risk to the environment if not addressed. Unlike "generic" patches, security vulnerabilities are announced through Sun Alert notifications. Patches to prevent these vulnerabilities should be treated with greater importance and applied as quickly as possible.

Different companies have adopted various strategies for applying security-specific patches. Companies try to balance the need to avoid possible problems caused by applying a new, but potentially problematic, patch too quickly with exposure to a security flaw while waiting for the patch to demonstrate stability (aging). Note that the average time lag between the announcement of a security vulnerability and its exploitation by hackers is shrinking[i], thus putting pressure on the speed of deployment, even in highly controlled environments.

In *Timing the Application of Security Patches for Optimal Uptime*,[ii] the authors built a mathematical model based on publicly available patch information to find the ideal time to patch a system. They conclude that 10 and 30 days after a patch's release are two optimum times to apply it. Note that this study is not particular to Sun patches.

Availability

There is a perception in the IT industry that updating software prevents unplanned downtime. Sun does not have statistically significant data to speak to this conclusion. It does have data that shows that overall downtime, planned and unplanned combined, goes up with more frequent application of patches. In cases where the goal has been to reduce outages from some potential bug by proactively

applying patches, Sun's analysis shows that this actually increases the amount of downtime because updating software often requires a reboot.

Following the strategy of minimizing change, an ideal system would never need its software updated and would remain forever available. This "Ideal" is represented by the straight line in the following diagram. For every cycle of time, there is a corresponding cycle of availability.

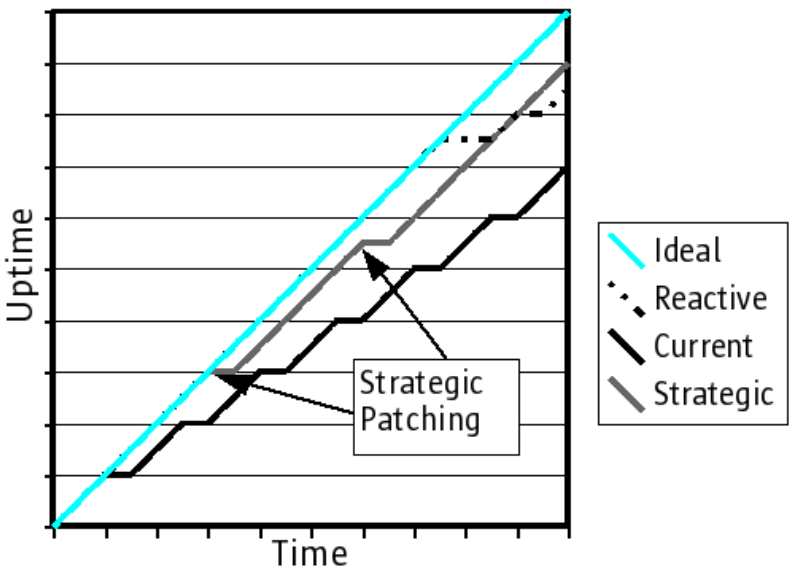

Figure 1.1 This diagram represents hypothetical strategies for applying patches and their impacts on availability.

In the event of a system outage due to a known problem, the uptime stops incrementing at the point of the outage, as shown by the short dotted line. Usually, the priority is to restore the system to operation first, then apply corrective measures later. Thus, a second, planned outage is required to resolve the issue. Note that an unplanned outage is usually longer than a planned outage because some time is needed for system recovery.

It would be tempting to avoid the unplanned outage altogether by proactively applying the patch that corrects the bug, and have just one planned outage. However, the problem is knowing beforehand which bug is likely to occur.

Many bugs for which Sun supplies patches have extremely small probabilities of causing an actual error on any given system. Some bugs will never be encountered

due to application stack, configuration of the system, or the environment in which the system runs, including data paths through the code. Attempting to inoculate a system for all possible bugs forces you to keep current with all patches as they are issued. Thus, the hypothetical "Patch to Current" approach (the long jagged line in the diagram) ends up creating a lot of planned outages to avoid having an unplanned outage. This results in the system losing more availability than it might have lost by just reacting to the bug.

By adopting a new approach in which security issues are addressed as they appear and needs are assessed periodically, labeled "Strategic Patching," it might be possible to decrease downtime over the "Patch to Current" approach, and perhaps even over the "Reactive" approach.

If you can determine that you are at risk for a given bug that will cause a critical outage, you should proactively patch it. In this scenario, the bug is targeted for proactive resolution before experiencing an outage, thus regaining a portion of potential lost availability.

For any given issue, a number of determinations must be made, for example, the probability that a given bug represents a sizable risk and that an outage caused by the bug is costly enough to attempt to avoid. Certain types of outages might involve costs beyond that of lost availability. Thus, this chart is only one axis of a multi-axis decision matrix.

Timing

Timing becomes especially important when determining which software release to run and when to apply patches to a system. More current releases of the Solaris OS provide successively higher quality. Where business needs are met, the most current software should be installed for the initial deployment environment.

Coping with withdrawn patches is another factor to consider. As noted earlier, *Timing the Application of Security Patches for Optimal Uptime* states that the best times to apply patches are 10 and 30 days after they are released. Since few Solaris patches are withdrawn (currently less than 2 percent for Solaris 8 and less than 3 percent for Solaris 9), the risk of applying these patches when issued is fairly low.

Workflow Process

This section describes how to implement the recommended strategy for updating software. The following diagram shows a high-level overview of the software update cycle.

Note that there is no defined end point to the cycle. The process is followed steadily, according to your policy, from the time that a machine is introduced into the environment until the time that the machine is decommissioned. Because Sun recommends that you complete an inventory before you change your environment, "Discover" is, in some sense, the beginning of the cycle, at least for the first round. Each of these steps in the cycle are described in the following subsections.

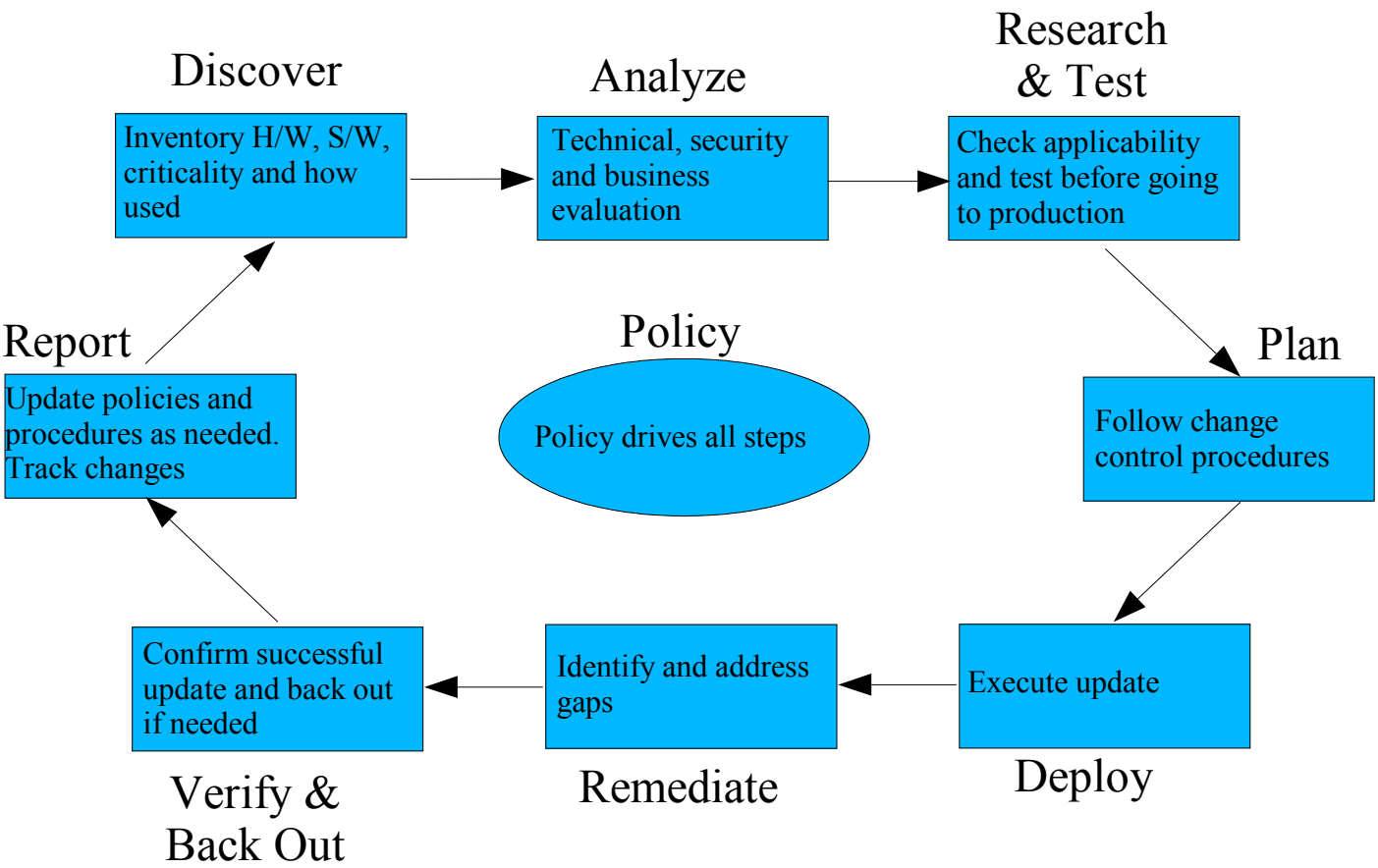

Figure 1.2 This diagram shows a high-level overview of the software update cycle.

1) Discover

The first step in establishing a new patch management and software update process is to perform a complete inventory of your environment. At a minimum, the inventory needs to include a description of the system's hardware (including firmware), operating systems (all versions, all updates, and all patches applied), application software (all versions and all patches applied), and storage devices.

This inventory should reflect systems, network appliances, routers, and other network infrastructure. The criticality of the system (such as test, development, and production) and the type of deployment (such as firewall and web server) should also be tracked.

You need to keep this inventory up-to-date, including the locations of specific applications.

2) Analyze

At this step, perform a risk assessment to determine the answers to these questions:

- Is the consequence of not updating this software (OS, application, and so on) so great that not even a cursory test deployment is completed?
- Is a series of tests to determine the effect on the environment appropriate?

This step involves performing the following:

- **Technical evaluation:** Assesses whether the updated software will correct a problem with the services and features of the applications used by your company.
- **Business impact assessment:** Determines whether updating or not updating the software will impact business processes. This assessment also determines the appropriate time for updating the software, for example, immediately, over a weekend, or at the end of the quarter.
- **Security evaluation:** Determines whether any security implications were not identified during the technical evaluation. Even though there might not be performance benefits to applying a patch, there might be security benefits.

In addition, companies that maintain their own internal computer networks, but utilize vendor-supplied applications, might be reluctant to apply patches to their operating system or update their software until the application vendor has assured them that a given operating system patch or update will not interfere with their software. Thus, companies should work closely with their vendors to

ensure that new patches are evaluated as soon as possible. Sun runs a number of vendor programs to ensure quick vendor certification of new software releases.

3) Research and Test

Before applying each patch, test it to ensure that it functions as expected and is compatible with the environment. Test each patch at the system level as well as in a complete quality assurance environment prior to applying it in the production environment. Integration testing helps to ensure a patch's compatibility with the system and with other components in the environment.

Evaluation and testing should also show that applying a patch or updating software does not open vulnerabilities previously corrected or produce new vulnerabilities. Sun chooses to accumulate its Solaris patches to avoid these types of regressions. Applying patches in the production environment is subject to normal change management procedures to minimize the risk of disruption due to applying the patch.

Also perform testing in the production environment after updating the software. At the very least, the standard production software stack *must* be exercised in a "flame on" mode to ensure that it is viable.

4) Plan

Plan in conjunction with normal change management procedures. Map out the specific steps, the time required, the proper notification of stakeholders, and contingency steps if something does not proceed as intended.

While this step might be less important for test and development systems, it is certainly relevant for a production system. The degree to which Change Control is performed is dictated by experience and existing company policies.

Having a detailed plan can expedite the overall deployment effort and provide for possible contingency plans, should they be needed.

5) Deploy

After the tests have been completed satisfactorily, you deploy, or push, the changes to the remaining systems. If the testing was conducted on a representative system, the chances of having trouble deploying the changes elsewhere are limited. Unfortunately, a production system frequently has unique characteristics, such as size, security configuration, or traffic volume, that are not readily duplicated in testing. Sun recommends that behaviors in areas where the

testing and production environments diverge be monitored the most closely after the deployment step.

Changes that do not deploy properly to other systems must be addressed through remediation of the gaps.

6) Remediate

Patches can be deployed manually or using tools that automate the process. Automation is particularly useful in reducing costs in large-scale deployments, scheduling deployments for specific maintenance times, and reducing the potential for human error.

One challenge to deploying patches is if remote users are not connected at the time of deployment. This leaves networks vulnerable from the remote users' systems because they have not yet been patched. An example is the Federal General Accounting Office's network being affected by the Microsoft Remote Procedure Call (RPC) vulnerability when remote users plugged their laptops into the network after being exposed to the vulnerability from other sources.[iii]

7) Verify and Back Out

While the previous analysis and test processes minimize risk to the environment, it is also necessary to provide for returning to the previous state in the event unforeseen conflicts, regressions, or errors occur. Solaris patches offer an advantage because they have a backout capability that is not available with Linux or Microsoft patches, for example.

While this might not be an actual “step” in the process, failing to provide a safety net can greatly increase the vulnerability of your company to system outages and increased patch management costs.

8) Report

A wide range of reports related to the software update cycle are produced by Sun and its customers. In many cases the reports pertain to fulfilling the need for data related to reentering the “Discover” phase. However, there is a distinct need for some data related to all steps in the cycle. For this reason, “Report” is a separate step in the process. Reports might include the following:

- **Policies and procedures that document the patch management program:** Are often incorporated into existing policies and procedures, such as an information security policy or systems development and implementation policies.
- **Clear definition and assignment of responsibilities for patch management at a functional level:** Includes contact information for all individuals responsible for system administration and maintenance.
- **Tracking of both implemented and rejected patches:** Includes changes between past, current, and future states.
- **Deployment progress, compliance, or file corruption across the enterprise:** Could include all of these factors.

9) Policy

At the center of each of these eight steps is the policy, which drives all phases of the software update cycle. A typical policy includes rules about scheduling downtimes or scheduled maintenance on systems, potentially different treatments for security-related patches, and delegation of responsibility for determining the specific versions of software on different systems.

It is in the policy that a company explicitly creates the relationships among and thresholds for cost, risk, availability, and timing for its unique environments. These relationships and thresholds are driven by the company's availability, auditing, and business support requirements and standards. The policy needs to be fully integrated into the company's Service Management process and be flexible enough to accommodate any future change in operational requirements.

An important component of all patch management policies is an explicit list of “trigger” events that start the patch process. Typical triggers include frequent monitoring of CERT advisories, notification from vendors (such as the Sun Alert program), and periodic (for example, quarterly or semi-annual), scheduled reviews that correspond with a company's business cycles.

Tool Recommendations

This section describes many of Sun's patch-related tools, the applications to which they are best suited, and how to use them. Some of the tools require a service contract and are noted as such. Sun recommends that you use these tools:

- Sun Patch Manager 2.0 or Solaris Patch Manager 1.0 with the PatchPro plug-in
- Sun Explorer software, to understand the installation environment
- The workflow outlined in "Solaris Patch Manager Base Version 1.0 Strategy White Paper" for applying patches to multiple machines. Go to http://www.sun.com/bigadmin/content/patchpro/.

Customers that qualify for on-site support can also use the following:

- Traffic Light Patchtool (TLP) with Sun Checkup as the analysis module
- TLP with Sun Explorer software

The next two sections, "Patch Management Tools" and "Patch Deployment Assistance," list Sun supported tools and services for updating software, along with a short description of each, a URL for more information, or both.

Patch Management Tools

- Sun Patch Manager 2.0
- Solaris Patch Manager 1.0
- PatchPro
- Traffic Light Patchtool
- Patch utilities

Sun Patch Manager 2.0

Sun plans to include Sun Patch Manager 2.0 in the Solaris 10 release. Version 2.0 offers the Sun Patch Manager browser interface, as well as the `smpatch` command-line interface. While you can use the browser interface to perform basic patch management tasks, you can use the `smpatch` command to perform those tasks and many others. The `smpatch` command is also available in two modes: local mode and remote mode.

Sun Patch Manager 2.0 for Solaris 10 incorporates PatchPro functionality to automate the patch management process. This process includes performing patch analyses on systems, then downloading and applying the resulting patches. Available patches can be selected individually, with automatic selection of

dependencies, or as a complete set, for action specified by the user: download or download and apply the patches to the system.

Sun Patch Manager 2.0 for Solaris 10 allows you to specify a local collection of patches or a local patch server on your intranet as an alternative to downloading via HTTPS directly from the Sun patch server.

Sun Patch Manager 2.0 is also available for download for Solaris 8 and Solaris 9. Go to http://www.sun.com/software/download/products/40c8c2ad.html. However, all features in Sun Patch Manager 2.0 for Solaris 10 are not included in the Solaris 8 and Solaris 9 versions of the Patch Manager product. See the Solaris product documentation at http://www.sun.com for all version-specific features.

For more information about Sun Patch Manager 2.0, go to http://www.sun.com/service/support/sw_only/patchmanager.html.

Solaris Patch Manager 1.0

Solaris Patch Manager 1.0 is included in the Solaris 9 release and provides both a graphical user interface and a command-line interface for the `patchadd`(1M) and `patchrm`(1M) utilities. Integration with PatchPro, which requires a separate download, is provided to allow for complete software analysis and automated download capabilities.

Solaris Patch Manager 1.0 utilizes the capabilities of the Solaris Management Console to provide multisystem remote patch management capabilities, scheduling, and logging. For more information about Version 1.0, see http://www.sun.com/service/support/sw_only/patchmanager.html.

Additional Notes About the Patch Manager Tools

Sun is actively engaged in reducing the amount of downtime that can occur due to rebooting and reconfiguring software during the patch management process. Some patches, however, must be installed in single-user mode or require user interaction to be safely installed. Patch Manager displays a dialog box that identifies these patches, and any patches that depend on them, for interactive installation by the user. See the README files for these patches to learn about any special installation instructions.

PatchPro

PatchPro performs the analysis, download, and installation functions for Patch Manager and is offered in the following ways:

- Fully integrated as the analysis engine for Sun Patch Manager 2.0 in Solaris 10
- As a downloadable add-on analysis module to Solaris Patch Manager 1.0 in Solaris 9
- Available separately as Solaris Patch Manager Base Version 1.0 for Solaris 2.6, Solaris 7, and Solaris 8

PatchPro performs a detailed analysis of the software and patches currently installed on a system, and identifies any patches appropriate to that specific system. Digitally signed patches, including any soft or hard dependencies, are downloaded, validated for security, and installed in the correct order. Full control of the download and degree of automated installation is provided by a configuration file. Scheduling is provided by the `cron` daemon.

Patches downloaded by PatchPro that should not be installed automatically, either as a result of the policy set by the user or due to patch types that are not appropriate for automation, are sequestered for subsequent action by the user.

Messages from PatchPro are logged to the `/var/adm/messages` file by `syslogd` by default. However, specific PatchPro logs can be configured as needed, and the configuration can be set to notify the administrator via email. Go to `http://patchpro.sun.com/`.

PatchPro Interactive

In addition to the PatchPro technology that is used with the Patch Manager products, a variation of PatchPro, PatchPro Interactive, can be found at `http://www.sun.com/sunsolve/patches`.

PatchPro Interactive allows you to interactively input system information. PatchPro Interactive generates a custom patch list based on this system information. Use this implementation to generate patch lists for systems that do not have direct access to the SunSolve site.

Traffic Light Patchtool

The Traffic Light Patchtool (TLP) is a multisystem patch management tool, often run in conjunction with Sun Checkup tool output and the Enterprise Installation Service (EIS) CD. While eRAS services provide recommendations only, TLP helps to remediate patch issues. TLP features include the following:

- Several small scripts and files to make applying patches easier
- An extra directory with firmware and Flash PROM patches
- A summary of README files

- A list of patches that require special handling, for example, patches that require that you reboot the system or modify configuration files

TLP can be used in two ways:

- **Off-site:** Sun Explorer files are transferred to Sun, TLP is run, and results are delivered to the customer by using FTP or a CD-ROM.
- **On-site:** When TLP is installed on a customer system, Sun Explorer files are collected and analyzed on that system. Results are made available on the same system. The customer can install patch clusters directly from this system.

With on-site TLP use, the customer also gets an overview of each system's implementation status, which depends on what patches are applied. The customer can define sets of patch clusters to apply to a set of systems, and these clusters can be tracked by the last successful cluster installation. The status can be set to, and reported as, GREEN, AMBER, or RED (Traffic Light).

The patch cluster installation is logged in the `/var/sun/EIS-CD.log` file, which provides an audit trail.

TLP is available for a fee, or as part of certain Sun Services contracts. For TLP, EIS, and other related Sun Preventive Services, go to `http://www.sun.com/service/preventive/index.html`.

Patch Utilities

For information about patch utilities, see the following man pages:

- `patchadd`(1M)
- `patchinfo`(1M)
- `patchrm`(1M)
- `pkgchk`(1M)
- `smpatch`(1M)
- `showrev`(1M)

Patch Deployment Assistance

Sun Explorer

Sun Explorer software is a large set of scripts that is used to gather information about a system and to store the output in the form of a tar file ready for analysis by another program or for transmitting to Sun. Sun Explorer can be set to email the output (with user consent) to the online database `ConfigDB`. You can then select your system and generate a multitude of reports from the database. These

reports cover Sun Alert issues, panics, cluster errors, disk errors, power problems, OS messages, and runtime information, or looking at disk configuration. From this data, the system administrator can pro-actively keep the system at the desired patch level or detect system messages that have been generated. An engineer can also use this information to address a problem the customer might be experiencing. For additional information about Sun Explorer software, go to http://sunsolve.sun.com/pub-cgi/show.pl?target=explorer/explorer&nav=patchpage.

Sun Checkup

Sun Checkup is an eRAS Services system configuration analyzer tool that can be downloaded for portable use by Sun Support Engineers. This tool enables the engineers to deliver a "lighter" version of eRAS Services' product knowledge into a given customer's environment. The tool includes a rules-based checklist.

Additional Tools and Services

To further assist you when managing patches or updating your software, see:

- **Tailored SunPS℠ program (for professional services)**
  http://www.sun.com/service/consulting/index.html
- **N1™ Grid Service Provisioning System**
  http://wwws.sun.com/software/products/service_provisioning/index.html
- **Solaris Live Upgrade software**
  http://wwws.sun.com/software/solaris/liveupgrade/
- **JumpStart™ software**
  http://www.sun.com/bigadmin/content/jet/
- **Solaris Flash software**
  http://wwws.sun.com/software/solaris/webstartflash/
- **Sun Control Station software**
  http://wwws.sun.com/software/controlstation/
- **Configuration and Service Tracker**
  http://www.sun.com/service/support/cst/index.html
- **Sun Alert Notifications**
  http://sunsolve.sun.com/pub-cgi/show.pl?target=sunalert_patches
- **Patch Club Report**
  http://sunsolve.sun.com/pub-cgi/show.pl?target=patchpage
  or https://subscriptions.sun.com/subscription_center/ecomm.jsp
- **Withdrawn Patch Report**
  http://sunsolve.sun.com/pub-cgi/show.pl?target=patchpage
- **Current Solaris Update**
  http://www.sun.com/software/download/products/40c8c2ad.html

Use Cases

This section contains a few scenarios and how they would be approached using the strategy and recommended practices outlined in this document.

Case 1: Minimizing Changes

A system has been built using a current Solaris update, which brings the system into a known state that is consistent with neighboring systems. Sometime later, two patches are applied. Patch 111111-01 addresses a security concern, and patch 222222-01 supports a new card that has been installed. Six months after the initial patch baseline was applied, the system is assessed to see if there are any new patches to apply from the latest Recommended Patch Cluster. Patches 111111-03 and 222222-05 are new. Should they be applied?

In this case, neither of the patches would be applied. With regard to the security patch, once the security issue was resolved with the -01 revision six months earlier, the bug need not be addressed a second time. Regarding the second patch, the new card is already supported with the -01 patch. Unless the new revisions address some problem or provide needed enhancements, they should not be applied.

Case 2: Replacing Obsoleted Patches

Patches 111111-01 and 222222-01 have been obsoleted by a new patch, 333333-01. This obsolescence means that the original patches will not appear in new patch and cluster releases. Thus, no new revisions of the original patches will be identified. The new patch will appear. Should the new patch replace the original two patches on the system?

No, there is no benefit to changing the system. The presence of the two patches will not have an adverse effect on future changes. By not replacing the patches, the rule of minimizing change to not disturb the system is being followed.

Case 3: Consistency

Several systems are built using the Solaris 9 4/04 release and are subsequently updated with several patches. A year later new equipment is to be added to the data center and should be made consistent with the original systems. Should the new systems be built using the 4/04 release or should all systems be built with the 4/05 release?

If the new systems are built to the 4/04 baseline, no existing systems need to be changed, which meets the strategy of minimizing change. This is an advantage of establishing baselines within your environment. However, if the business climate has changed and the existing systems are in need of some enhancements to meet the new requirements, you might consider updating all systems, existing and new, to the 4/05 baseline. If such a change is warranted, using more recent software helps to ensure the best quality and meets the strategy of using the most current software for your business needs.

Case 4: Booting the System Now or Later

Many of the security patches require a boot after they are applied. Because these patches are released at various times, the system will have to be booted each time for each patch to have an immediate effect. Would it be better to install the patches without booting, then wait to perform a single boot so that all patches are applied at the same time, which would reduce overall downtime by reducing the frequency of boots?

You must analyze your site's environment and each patch to determine the applicability and likelihood of experiencing a problem. If possible, it might be preferable to apply the patch but wait to boot the system, and use a workaround until the next planned downtime.

i See http://www.cert.org/stats/cert_stats.html

ii Beattie, Arnold, Cowan, Wagle, Wright, and Shostack, *Timing the Application of Security Patches for Optimal Uptime*, 2002 LISA XVI, Philadelphia, PA

iii Dacey, Robert F., Director, Information Security Issues, *Effective Patch Management is Critical to Mitigating Software Vulnerabilities*, Testimony Before the Subcommittee on Technology Information Policy, Intergovernmental Relations, and the Census, House Committee on Government Reform, General Accounting Office, September 10, 2003