

# Beyond Patch Management


Dan Shauver

**Abstract:**

Systems maintenance, including operating system and software upgrades and patch management, has long been a major factor in security-related incidents. Application upgrades and patches can be equally necessary to system integrity, yet are equally likely to be ignored. In some cases, compromise of an unpatched system could have been avoided through proper system configuration.

Implementing a configuration management system can increase the ability of a systems administrator to deal with patch releases and upgrades, on both the operating system and software sides, as well as ease the management of system configurations across any environment, large or small. Yet nothing is perfect, and even the best of systems have security implications that should be considered.

"Most IT organizations still install and maintain computers the same way the automotive industry built cars in the early 1900's: An individual craftsman manually manipulates a machine into being, and manually maintains it afterward." (1)

Recent events have proven, once again, that faulty or non-existent patch management policies can lead to system compromise (2). This is due, largely, to the slow adoption of configuration management applications and methodologies in the world of system administration. Configuration management systems, or self-healing systems, are becoming more widely accepted. It is likely, within the near future, that they will become standard practice. "Even in the smallest local area network you will want to build a scheme for automating host configuration and maintenance, because networks have a way of growing from one host into many quite quickly." (3) The security implications of such applications, both good and bad, should be examined and understood during or before the initial implementation of such tools.

## Defining Terms

Configuration management systems, also referred to as self-healing systems and discussed under the heading of Computer Immunity, are a piece or set of software intended to maintain the integrity of computer systems in a distributed environment. The intent is to allow some measure of automation in remote management, especially of operating system patches and configurations.

There are many types of configuration management systems available today. They generally operate on a client-server basis, and utilize both a "golden image", and the concept of templates. The "golden image" is the base operating system install for a computer or set of computers, while the templates are site- or class- specific customizations that are applied "over" the golden image. Both the golden image and the templates are stored centrally for distribution to the rest of the computing environment.

Most operating system vendors have some method of maintaining operating systems post installation. There are a wide variety of 3rd party vendor tools, and the open source community has also contributed some applications to this field.

Each system has varying capabilities and supports a varying range of operating systems. Tools supplied by operating system vendors generally don't support operating systems from other vendors, while the expense of third party tools like Tripwire or BMC can be prohibitive, especially in larger environments. Open source tools, such as cfengine and radmind, are priced attractively, but support of the software is the responsibility of the user of the software; some organizations also have policies against the use of open source software.

The systems all operate, however, in roughly the same fashion. The terminology and featureset may be different, but the intent and basic functionality is the same.

Each distinct configuration management system will have distinct benefits and drawbacks. The intent of this practical is to discuss some of the general highlights and concerns of managing a computing environment in this fashion. The discussion will be related to security; it is perhaps useful to recall that security has three main tenets – confidentiality, integrity, and availability.

**The Good**

The automation supplied by configuration management systems is their greatest benefit. The dangers of not patching systems have been demonstrated, discussed, and written about frequently and consistently. One need only look at the CERT homepages to see the variety of vulnerabilities in existence today. The majority of these vulnerabilities can be remedied with publicly released patches. The difficulty lies in maintaining proper patchlevels on a number of different servers in a computing environment. Configuration management systems reduce the burden of this portion of systems administration.

Patches, software updates, or configuration changes can be integrated into either golden images or templates, as appropriate. All clients can then automatically apply the patches according to scheduled updates. Reboots, where necessary, can also be scheduled through the configuration management system.

Normally, application of patches can be a very significant time requirement in larger environments; insufficient time is one of the more common explanations for not installing patches. The likelihood of forgotten systems also decreases, as the only requirement for the application of patches is that each computer have a configuration management client installed upon it.

Similarly, configuration files can be modified and/or maintained through a configuration management system. Required changes to system and network configurations, including the disabling or enabling of ports and services, can be distributed in this fashion, allowing for more consistent configurations across a site's entire computing environment. As with patches, the only requirement for the maintenance of a proper system configuration is the installation of the client piece of the configuration management system. All clients will be given an approved set of configuration files.

As one would expect, other applications can be managed in a similar fashion. The use of templates allows for application installations on specific subsets of systems, allowing for, as an example, one particular installation (and configuration file) for Sendmail on the majority of Unix servers, with a different installation on mailhubs and gateways. Upgrades can be handled in a similar fashion, with a new template generated to include the new software version, and

servers being told to request that template in stages, allowing for simple, low effort, and low impact rollouts of new software and versions.

Perhaps the greatest benefit comes from not having to remember each individual server. As long as the server is under the oversight of the configuration management system, patches, upgrades, configuration changes, and new applications will be applied automatically, on a presumably predetermined schedule. One no longer needs to rely on memory or documentation, both of which can be faulty, to maintain properly configured and patched servers.

Another benefit is the reduction of human error. Rather than having to apply changes or patches manually on multiple machines and maintain or write scripts for multiple servers and operating systems, the systems administrator only has to maintain the golden image and templates. Error checking becomes much simpler, and the amount of time required for quality assurance for changes, patches, etc is greatly reduced. Outages and compromises related to human error should become much less common, greatly increasing the availability of systems and data to the enduser.

Change management procedures can also be enforced with a configuration management system. Any changes that are not applied to the golden image or template will not last through a single update. This reduces the incidence of undocumented changes; however, a forgetful systems administrator may have to repeat work, if the proper procedure is not followed. Any changes that will be both applied and remain applied will be included in the configuration management system, and thus "documented" to some extent.

Along with change control, a configuration management system also allows automated incident response. Any intrusions or incidents that do not damage the client software will be automatically recovered; the next scheduled run of the configuration management system will overwrite any changes, returning the compromised system to a known good state. The same facility that will frustrate careless administrators can also allow for automated incident response, potentially reducing downtime and decreasing damage caused by compromised systems.

With a properly configured configuration management system in place, the installation process can become nearly completely automatic. A system can have a base, vanilla operating system installed, along with the client piece of a configuration management system. The client can then take care of additional, site specific configurations, as well as ensuring that the system is up-to-date on critical patches and software updates. Any non-operating system software appropriate to the class in which the client resides can also be installed. Depending on the complexity of the configuration management system, restore jobs could be launched to put data in place. In short, the computer system, after the installation of a basic operating system and single piece of client software,

could be put into a production-ready state, complete with data, automatically. This removes a great deal of opportunities for error on the part of a system administrator during the various phases of system builds.

If the initial build were performed in an automated matter, by booting from a custom installation CD or from a networked installation service (such as Jumpstart or Ignite), the complete build process for a system could be reduced to a single command line. While this does not entirely eliminate the possibility for error, it does reduce it.

All of these benefits reduce the demands on the systems administrator(s). The time saved could then be spent increasing the overall security of the environment in many ways, including, but not limited to, research and quality assurance testing of patches and software updates and professional development.

Reducing the manual, repetitive, and frequently boring tasks required of a systems administrator, aside from increasing overall staff productivity, is also likely to increase morale. Salary surveys (10) often cite interesting or challenging work as a reason for job satisfaction or job change. Staff turnover is frequently damaging to the security of an environment. It is also widely acknowledged that the majority of security related incidents are caused by disaffected staff. Decreasing the necessity for repetitive, boring work should aid in increasing the morale of the support staff, hopefully reducing the likelihood of staff turnover or security incidents.

A configuration management system, once implemented, could also be used to encourage participation in organizational policies and best practices. While many corporations do not have difficulties enforcing security related policies, many educational and not for profit organizations have a considerably harder time, often having to deal with both fragmented IT organizations and poorly defined or enforced policies.

Management support for strictly enforced policies at such organizations is usually poor, at best. However, by granting access to golden images and templates, other internal IT organizations could be encouraged to follow good security practices by realizing the benefits of a configuration management system, rather than relying on management that may be unable or willing to enforce such practices.

**The Bad**

The good features and capabilities of a configuration management system have been described above. No product, however, is without its drawbacks. There are certainly issues to consider with implementing such a system. These issues range from the merely annoying to the potentially dangerous and should be understood prior to implementing such a system.

One of the benefits mentioned above is automated incident response. This is accomplished by returning the client to a known good state. While this does return the client to expected functionality, it also has negative effects. First, it puts back in place whatever vulnerability allowed the compromise in the first place. It is almost certain, in an automated response situation, that any evidence related to the incident will be altered or destroyed. Log files may not change, but any installed or modified files would likely be removed.

While some configuration management systems can be configured to send notifications or make copies of changes, it is usually not the default behavior of the software. Others do not have that capability. After all, one of the greatest benefits of a configuration management system is that it takes work away from the system administrator. Notification of changes without automatic fixes/rollbacks, while useful from a security standpoint, does not significantly reduce the workload on the system administrator. In fact, it increases the workload, informing them of changes that may or may not be harmful, but which certainly would not have been noticed earlier. This can easily lead to information overload – the administrator may get bogged down looking at thousands of insignificant changes, while a crafty intrusion slips through in the tide of information.

Second, if a system administrator were to take quick action to fix a vulnerability, prior to or following an intrusion, they then need to remember to update the configuration management system appropriately, or their work will be undone. The process by which critical security fixes are applied is often carried out in abnormally stressful moments. Adding complexity to this process will delay the speed at which the process can move, and increases the likelihood of error.

In addition, a configuration management system can be a vehicle for the rapid distribution of mistakes. If a systems administrator were to follow proper procedure, and use the configuration management system to push out changes and/or fixes for a critical problem, any mistakes made, especially in the “heat of the moment”, would now be automatically distributed to the entire environment. While mistakes made without automated systems are likely to be noticed and corrected during the process of implementing them across an large environment, a configuration management system ensures rapid and exact duplication of any mistake. The response to any incident, then, could lead to much wider-spread damage and outages than the incident itself.

Furthermore, the key to automated response to intrusions and/or changes is having a centralized repository of configuration information and golden images or templates. The configuration management clients need to know where to look to determine what may have changed since the last update. This makes the repository server a single point of attack for all systems managed by the

configuration management software. Compromise the golden image server and you can now automatically compromise every managed system.

There are other concerns revolving around the golden image server. Some configuration management systems, for example, do not authenticate between client and server. This makes spoofing the golden image server possible, and allows for any system to request updates and/or information about any client. An unauthorized workstation could potentially download operating system images for all managed systems from the golden image server. At the very least, this allows for anyone with the proper network access to gain an enormous amount of information about potentially sensitive systems. Depending on the extent of management, this could mean access to user information, including passwords in an easily crackable format.

Some systems make this even easier by not encrypting the data transferred between client and server. While requesting an image from a server might require specific knowledge of the configuration of the client and server, unencrypted traffic makes such knowledge un-necessary or, if necessary, readily available with the use of a variety of tools. A sophisticated attacker could also intercept and alter data in transit, leading to system compromise and false reporting.

Finally, any configuration management system adds a degree of complexity to the computing environment. While this should be largely offset by reduced workload in managing a large variety of systems, it does require additional training for system administrators, and adds additional steps to any system or application builds. While the distribution of applications is simplified, any new application must be installed, tested, and packaged via the configuration management system; the application should also be tested prior to distribution. New systems must go through a similar process. Disaster recovery procedures should also be modified to encompass the configuration management system; it could be beneficial in a disaster recovery environment, but must be taken into account and tested with the rest of the environment.

A poorly configured configuration management system could lead to more work than it saves. Environments in which large numbers of similarly configured systems exist could see a reduction in required effort for system management. More complex environments, with many specialized classes of machines, could see an overall increase in management requirements; each template or system image requires management and customization.

**The Ugly**

There are other issues with configuration management systems that may or may not adversely impact the security of an environment, but should be considered.

The generation of golden images and/or templates can be exceedingly difficult. While some systems allow for snapshots of prebuilt systems, some systems require manual creation of images. In some cases, the golden images, while snapshots of systems, are single cpio/tar images, making updating images a timely and potentially disruptive process. Other systems do not allow for overlays of templates, requiring a separate image for each distinct server. While none of these are insurmountable problems, they do make the management of the configuration management system more difficult. Most importantly, from a security perspective, this increases the opportunity for mistakes, and can discourage the regular updates of images. Out of date images will not accurately reflect the state of the environment, and could both rollback previously applied patches, and potentially cause outages or loss of data, should an especially old image be applied to a production system. If a disaster recovery plan were to include the use of a configuration management system, images that are not current could dramatically increase the time required to recover all systems.

The storage requirements of configuration management systems can also be quite problematic. Most products require a central repository for configuration files, binaries, and, in some cases, entire directory structures. While a golden image for a single operating system and application combination may not require more than a few gigabytes of disk space, more complex environments may end up storing multiple images for various operating systems, release levels, and server functions. Disk space requirements can become quite large in these cases. Mirroring of data for fault tolerance or the implementation of multiple configuration management system servers for load distribution purposes can further increase these requirements. While disk space is cheap these days, each additional gigabyte increases resource requirements for backup solutions, and increases the time requirements of disaster recovery scenarios.

The golden images and templates must also be compared to, and sometimes pushed to, servers in the computing environment. In some cases, this process can increase the load on a server enough to cause poor performance, up to the point of perceived outage to the user population. In systems where changes are automatically corrected, or in cases where the configuration management system is used to roll out applications or patches, significant network traffic can be generated. This traffic can be enough to impact the operation of a production server. If a significant number of clients attempt to get updates from the configuration management system, the system itself could be overloaded, leading to failed updates. In certain situations, such as the rollout of security patches or major configuration changes, clients could end up causing a denial of service on the configuration management system, leading to a delayed or failed rollout.

Depending on the initial setup and capabilities of a configuration management system, it still may require a base operating system install prior to updates from golden images or templates. As this will also require network connectivity, this

could either lead to servers being available on the network in an unpatched state, or the maintenance of a separate patch repository and install process.

Not all configuration management products support a wide variety of operating systems, either. Some are limited to a single platform, while open source systems may require custom development if non-standard or older operating systems exist in the computing environment.

**Conclusion**

The main security benefits of a configuration management system are in the arena of availability and integrity. The availability and integrity of a server are increased by the automated synchronization against a set of templates and golden image. This allows for assurance of proper configuration as well as rapid deployment of critical patches, configuration changes, and application updates.

One risk is quite similar – a configuration management system allows for the rapid deployment of human errors. It could also lead to the rapid compromise of an entire computing environment, should someone with malicious intent gain access to the configuration management system itself. Finally, the golden image and template data transferred between server and client is sometimes transmitted in the clear; this situation could lead to quick and thorough information gathering for those wishing to make unauthorized changes to the computing environment.

Some of the risks are unavoidable, and can only be mitigated through careful and consistent application of security best practices. Others can be eliminated through the implementation of other technologies – LDAP servers for authentication issues, SSL or IPSEC for unencrypted data. This type of application is likely to become more and more common. The risks and benefits should be examined and understood, hopefully prior to implementing such a solution.

**References:**

1. Infrastructures.org homepage. April 2004. http://www.infrastructures.org
2. Information Technology Systems and Services, Stanford University, "Multiple UNIX compromises on campus -- 10 April 2004." 10 April 2004. http://securecomputing.stanford.edu/alerts/multiple-unix-6apr2004.html
3. Burgess, Mark. "Managing Network Security With Cfengine." 1999. http://www.cfengine.org/docs/cf-security.html
4. Burgess, Mark. "Computer Immunology." 1998. http://www.iu.hio.no/~mark/research/immune/AIdrift/AIdrift.html
5. Jurevicius, Adam. "Configuration Management in a Heterogeneous Environment." 04 January 2004. GSEC Assignment 1.4b-Option 1. http://www.giac.org/practical/GSEC/Adam_Jurevicius_GSEC.pdf
6. Cromar, Scott. "Configuration and Patch Verification on Solaris Systems". GSEC Assignment 1.4b-Option 1. 22 Jan 2003. http://www.giac.org/practical/GSEC/Scott_Cromar_GSEC.pdf
7. Limoncelli, Thomas and Hogan, Christine. "The Practice of System and Network Administration." Addison Wesley. August, 2001.
8. Craig, Wes, et al. Radmind documentation. http://rsug.itd.umich.edu/software/radmind/documentation.html
9. The Rsync Team. Rsync homepage and documentation. http://samba.anu.edu.au/rsync/
10. SAGE. SAGE Salary Surveys. http://sageweb.sage.org/jobs/salary_survey/

Last Updated: January 28th, 2020

# Upcoming SANS Training

**Click here to view a list of all SANS Courses**

| | | | |
|---|---|---|---|
| SANS New York City Winter 2020 | New York City, NYUS | Feb 10, 2020 - Feb 15, 2020 | Live Event |
| SANS Northern VA - Fairfax 2020 | Fairfax, VAUS | Feb 10, 2020 - Feb 15, 2020 | Live Event |
| SANS London February 2020 | London, GB | Feb 10, 2020 - Feb 15, 2020 | Live Event |
| SANS Cairo February 2020 | Cairo, EG | Feb 15, 2020 - Feb 20, 2020 | Live Event |
| SANS Dubai February 2020 | Dubai, AE | Feb 15, 2020 - Feb 20, 2020 | Live Event |
| SANS Brussels February 2020 | Brussels, BE | Feb 17, 2020 - Feb 22, 2020 | Live Event |
| SANS Scottsdale 2020 | Scottsdale, AZUS | Feb 17, 2020 - Feb 22, 2020 | Live Event |
| SANS San Diego 2020 | San Diego, CAUS | Feb 17, 2020 - Feb 22, 2020 | Live Event |
| Open-Source Intelligence Summit & Training 2020 | Alexandria, VAUS | Feb 18, 2020 - Feb 24, 2020 | Live Event |
| SANS Training at RSA Conference 2020 | San Francisco, CAUS | Feb 23, 2020 - Feb 24, 2020 | Live Event |
| SANS Manchester February 2020 | Manchester, GB | Feb 24, 2020 - Feb 29, 2020 | Live Event |
| SANS Secure India 2020 | Bangalore, IN | Feb 24, 2020 - Feb 29, 2020 | Live Event |
| SANS Jacksonville 2020 | Jacksonville, FLUS | Feb 24, 2020 - Feb 29, 2020 | Live Event |
| SANS Zurich February 2020 | Zurich, CH | Feb 24, 2020 - Feb 29, 2020 | Live Event |
| Blue Team Summit & Training 2020 | Louisville, KYUS | Mar 02, 2020 - Mar 09, 2020 | Live Event |
| SANS Northern VA - Reston Spring 2020 | Reston, VAUS | Mar 02, 2020 - Mar 07, 2020 | Live Event |
| SANS Munich March 2020 | Munich, DE | Mar 02, 2020 - Mar 07, 2020 | Live Event |
| SANS Secure Japan 2020 | Tokyo, JP | Mar 02, 2020 - Mar 14, 2020 | Live Event |
| ICS Security Summit & Training 2020 | Orlando, FLUS | Mar 02, 2020 - Mar 09, 2020 | Live Event |
| SANS Jeddah March 2020 | Jeddah, SA | Mar 07, 2020 - Mar 12, 2020 | Live Event |
| SANS St. Louis 2020 | St. Louis, MOUS | Mar 08, 2020 - Mar 13, 2020 | Live Event |
| SANS Prague March 2020 | Prague, CZ | Mar 09, 2020 - Mar 14, 2020 | Live Event |
| SANS Paris March 2020 | Paris, FR | Mar 09, 2020 - Mar 14, 2020 | Live Event |
| SANS Dallas 2020 | Dallas, TXUS | Mar 09, 2020 - Mar 14, 2020 | Live Event |
| Wild West Hackin Fest 2020 | San Diego, CAUS | Mar 10, 2020 - Mar 11, 2020 | Live Event |
| SANS Doha March 2020 | Doha, QA | Mar 14, 2020 - Mar 19, 2020 | Live Event |
| SANS London March 2020 | London, GB | Mar 16, 2020 - Mar 21, 2020 | Live Event |
| SANS Norfolk 2020 | Norfolk, VAUS | Mar 16, 2020 - Mar 21, 2020 | Live Event |
| SANS SEC504 Nantes March 2020 (in French) | Nantes, FR | Mar 16, 2020 - Mar 21, 2020 | Live Event |
| SANS San Francisco Spring 2020 | San Francisco, CAUS | Mar 16, 2020 - Mar 27, 2020 | Live Event |
| SANS SEC401 Lille March 2020 (in French) | Lille, FR | Mar 16, 2020 - Mar 21, 2020 | Live Event |
| SANS Secure Singapore 2020 | Singapore, SG | Mar 16, 2020 - Mar 28, 2020 | Live Event |
| SANS Security East 2020 | OnlineLAUS | Feb 01, 2020 - Feb 08, 2020 | Live Event |
| SANS OnDemand | Books & MP3s OnlyUS | Anytime | Self Paced |