Global Information Assurance Certification Paper

IBM's Cluster 1600 Security Aspects

written by John P. Belliveau
GSEC Practical version 1.3

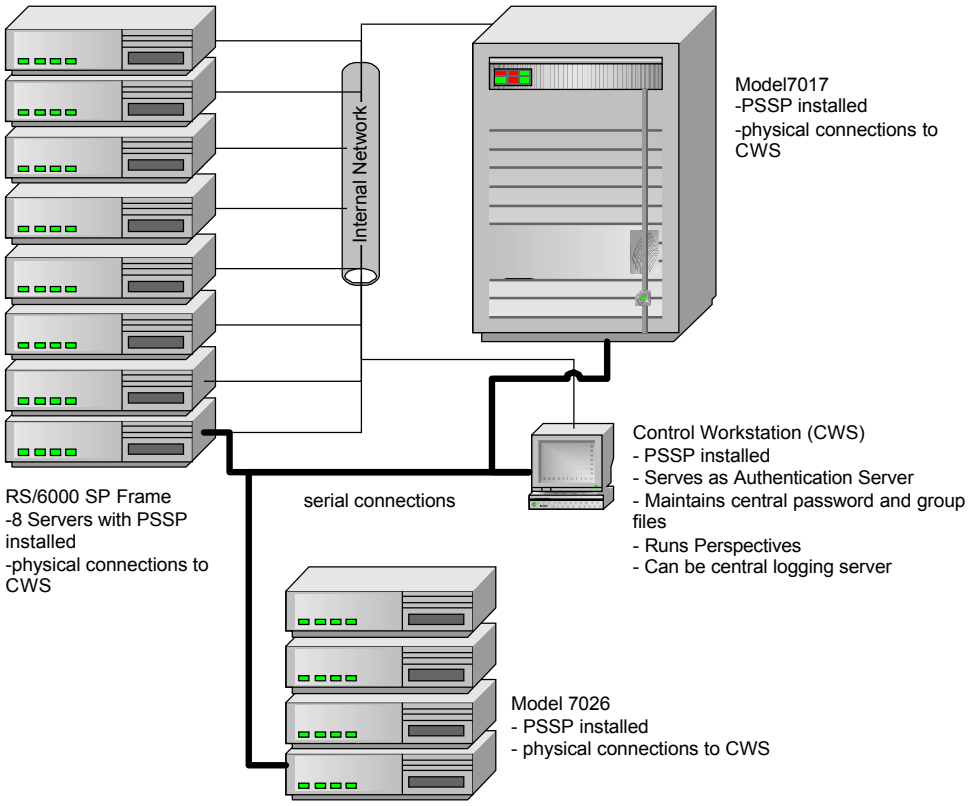
Abstract

This paper discusses the new International Business Machines (IBM) Corporation environment known as Cluster 1600, the software at the heart of the environment, and the security features of this software. In combination with IBM's Unix operating system, AIX, the Cluster 1600 environment is based on IBM's Parallel System Support Programs. These software packages provide security features that address secure remote administration, centralized and secure password administration, delegation of root authority, log monitoring, and file auditing. Cluster 1600 takes security seriously. It provides simplified and centralized administration that gives a system administrator the appropriate tools to be effective.

Background

The IBM Corporation has embraced their supercomputer software package, Parallel System Support Programs (PSSP) for AIX, and begun including other IBM hardware platforms to create the Cluster 1600 system. PSSP for AIX is a suite of applications that IBM implemented on the RS/6000 SP supercomputer to manage the entire parallel processing system. Though many people may not be familiar with PSSP, they may be familiar with the 1996 historic chess match between Garry Kasparov, a world chess champion, and Deep Blue, an IBM supercomputer. Shockingly, IBM's supercomputer beat the chess champion at his own game. So, what does this have to do with PSSP? PSSP is the heart of the supercomputer. This suite of applications has components dedicated to security, and IBM has begun supporting PSSP on some of their mid-range and high-end servers, too. This new implementation of PSSP, referred to as Cluster 1600, not only includes the IBM supercomputer, but it can include large enterprise servers, like the new p690, and mid-range RS/6000 and p-series Unix servers as well. The importance of this new development is the growing environment in which PSSP can run.

Cluster 1600



Remote Administration

Administration of large Unix environments can be cumbersome and insecure, if not managed properly. One difficulty administrators experience is the gathering or presentation of data from multiple servers. Remote commands like rsh and rcp allow Unix administrators to manage remote systems, but the security implications are significant. Rsh and rcp can reference files to grant the user access without supplying a password. The .rhosts, hosts.equiv, and hosts.lpd trusted system files, can open huge security holes if the files do not have the appropriate rights, the contents of the file are not configured properly, or the contents of the file are altered to create a back door for unauthorized entry later.

PSSP has built-in security. It integrates standard AIX security, including r-commands, if desired, but it also includes Kerberos 4 authentication. These options are included with PSSP free-of-charge. When choosing Kerberos 4 authentication, the options exists install a Kerberos server on the control workstation of the cluster or to use an existing Andrew File System (AFS) authentication server. AFS is a commercial application that uses an implementation of Kerberos 4 that can be used by the Cluster 1600. Additionally, PSSP offers the integration of a Kerberos 5,

Distributed Computing Environment (DCE), however, this is not part of the PSSP software.

The Kerberos included with PSSP is based on Kerberos version 4 developed by the Massachusetts Institute of Technology (MIT). It is used to identify and authenticate a user running a remote command and then authorize the user by confirming the user has the rights to run the command on the server requested. This is all done without transmitting open text passwords to the application or remote server. For example, if John Doe tries to run a remote command on an application server, the Kerberos process first asks who is running the command. In this case, John Doe is prompted for his Kerberos principal or his Kerberos login. Then, the server authorizes John Doe by asking for a password. This password is not transmitted to the application server John is trying to access. The password is used to decrypt a ticket sent to John by the Kerberos server which he will use later to get to the application server. After John is identified and authorized, the command is executed. The application server receiving the request checks whether the user is authorized to run the command. The application server looks in an access file to see if the Kerberos principal running the command is authorized to do so. The application server knows John's authenticated based on tickets granted to him by the Kerberos server. In other words, he has a ticket. The Kerberos ticket granting process is all done using the Data Encryption Standard (DES) algorithm for encryption of the Kerberos tickets. DES is used to encrypt the passwords used in the Kerberos process, though it is not used to encrypt the data. For more specific information on the Kerberos security process, refer to PSSP administration documentation or search the internet. There is plenty of information on Kerberos.

Kerberos has several components and functions. Among them are the Key Distribution Center (KDC), which includes the Authentication Service. and the Ticket Granting Service (TGS). These are the third-party authorities for authorizing users and granting tickets to be used by the client and server. The installation of PSSP provides an easy way to install and configure these main components. After installing the PSSP software, the installer runs the command *setup_authent.* This command asks a few basic commands before creating the Kerberos database and configuration files. There is also an option to create a secondary Kerberos Authentication Server for redundancy, but this is not a requirement.

One big advantage of the Kerberos integration into PSSP is the option to remove .rhosts and hosts.equiv files. With Kerberos authentication, an administrator can securely run a root-level command on one or more remote hosts without the existence of these files. As a matter of fact, any user given Kerberos credentials can execute remote commands without giving a password. The execution of the command is based on the rights of the individual who is executing the command. So, a person in the system group can only execute commands or view files that the system group or he/she has rights to access. The .rhosts files and/or hosts.equiv files do not need to be maintained. The only administrative requirement is to make sure .rhosts files do not exist.

Another advantage are the PSSP distributed commands, or parallel management commands, that take advantage of Kerberos. When PSSP is installed, the AIX version of rsh, remote shell, is replaced by a version of rsh that uses Kerberos. An administrator can, therefore, securely execute a command on a remote host using the rsh command. The *dsh* command uses this version of rsh and can be used to run commands on multiple remote hosts concurrently. For

example, to execute the *date* command on all hosts in the Kerberos environment, an administrator can execute the command *dsh -a 'date'*. The results are shown below.

```
Server1: Tue Dec 18 08:48:41 EST 2001
Server2: Tue Dec 18 08:48:41 EST 2001
Server3: Tue Dec 18 08:48:41 EST 2001
Server4: Tue Dec 18 08:48:41 EST 2001
Server5: Tue Dec 18 08:48:41 EST 2001
```

If an administrator wanted to make sure the /etc/inittab on all servers contained an entry for /etc/rc.shutdown, he or she could execute *dsh -av 'egrep "rc.shutdown" /etc/inittab'*. PSSP includes many other parallel management commands that use the kerberized remote commands. A few include *pcp*, used to securely copy a file to multiple remote hosts, *pfps*, used to list processes like the ps command, and *pdf*, used to run the *df* command.

Root privilege for others

In larger organizations, there are times when the delegation of authority and administration is required. In other words, some administrators are tasked to perform some functions that require root-privileges, but are not required to or should not be able to perform other root-privilege tasks. PSSP provides a program that allows the delegation of root authority without giving out the root password. This is particularly important in a PSSP environment where the /etc/passwd and /etc/security/passwd files are all the same. If you give the root password for one host, you are effectively giving out the root password to every host that uses the PSSP file update option.

There is a popular freeware program that allows the delegation of root authority. It is called *sudo*. Many experienced Unix administrators are familiar with this tool. The *sysctl* system included with PSSP provides the capability to provide root authority to non-root users. It basically provides the same capability, but it also allows for the distributed and parallel execution of these commands or scripts. The sysctl system is secure, because it uses the Kerberos authentication system.

The sysctl program can be customized by a system administrator to include any script or command in the environment. The PSSP Administration Guide gives complete instructions on how to setup the sysctl client and provide access control for a user or group of users. The administrator can not only authorize a user or group of users to execute a command, but he or she can control on which servers the user or users can execute the command. For example, a group of users may need to execute the *exportfs* command on servers one and three. Using sysctl to allow this access is secure, since it requires Kerberos authentication; it is controlled using access control lists; and it is efficient, because the lead administrator does not have to be bothered every time the user needs to export a particular file system.

User and Group Security

Password security is greatly complicated when users need to maintain passwords on multiple

systems. One password changes on system A one day, another changes on system B two weeks later, and the pattern continues. PSSP offers a program called *supper* that allows for the automated distribution of files to remote servers. By simply selecting an option from a menu during the configuration of the PSSP, multiple user and group files are included in the automated distribution cycle. The supper command is similar to the *rdist* command, a popular command for remote file distribution on Unix.

The supper program uses the *sup* protocol to distribute files to remote servers. Upon installation of the PSSP software, an entry is added to the root crontab of the server. This entry runs the supper command every hour to update the /etc/passwd, /etc/security/passwd, /etc/group, /etc/security/group, and other files. If desired, an administrator can simply update the crontab on each server to update the files in the file collection at another interval, like ten minutes. This way a user only has to wait ten minutes for his/her password to take affect on all nodes in the cluster environment.

The automated distribution of the /etc/passwd, /etc/security/passwd, /etc/group, and /etc/security/group files serves a few functions. First, it allows a user to change their password at one location, the control workstation in the PSSP environment, and have that password distributed to all other servers. One positive security aspect of this process is the ease of password management for users. The more difficult it is to change a password for a user, the more they'll try to circumvent the process. Just look under the keyboard of the person in the next cubicle and there is a good chance there will be a password list.

Another positive aspect is ease of administration. An administrator only needs to create a user once. The account is then distributed to all servers in the PSSP environment. That brings up a potential vulnerability, though. If an administrator does not maintain the /etc/security/user file for each server, a user may have access to systems they shouldn't be on. Of course, the PSSP software has another script that allows an administrator to restrict access to a server. This script is called *spacs_cntrl.* On the other hand, the automated distribution of security files like /etc/passwd provides some protection, too. If a hacker breaks in to a system and updates the /etc/passwd file, that file will be overwritten in the scheduled period. To hide an account in the system to later use, a hacker would need to break into the control workstation from which the "real" security files are maintained and distributed.

Monitoring and Log Management

Log review and auditing are essential for intrusion detection. Of course, gathering information does not force someone to review it, but without first properly gathering information, the next step is impossible. PSSP offers a few ways to monitor logs and events on remote systems. These include Perspectives, a graphical-user-interface program, and parallel commands that extract data from remote logs.

Perspectives has a component called Event Perspectives. This part of the program provides a default set of events and conditions to monitor throughout the Cluster 1600 environment. For

example, the program can check that a particular daemon is running on a system. It also checks file system usage and monitors the error logs for permanent error entries. When conditions change or are outside the designated range, an action is triggered. The events, conditions, and actions are all customizable. The Event Perspectives program gathers its information from daemons that are installed and running on all nodes in the Cluster 1600 environment. Therefore, no special scripts or crontab schedules are necessary to distribute and maintain across the nodes.

For log management, the capability to monitor file system usage is important. An administrator can copy the default file system events and customize the conditions and actions taken when an event is triggered. By default, the /var and /tmp file systems are monitored. However, if an administrator has a separate file system for logs, this file system can be monitored as well. The default condition for the file system event is 90 percent full. If the system becomes 90 percent full, an action occurs. This default condition can be copied or set higher or lower. The default action is simply to show the error in the Event Perspectives notification logs. This default notification is not effective, though, unless and administrator has the Event Perspectives program running on the system in front of them. The Event Perspectives program allows an administrator to add to or completely change the error notification process and actions. The event can be logged to the AIX error log, an SNMP trap can be sent to an SNMP manager, or a custom script can be run. An effective command action is to mail or page the event to administrators. If desired, the command script can increase the file system that is filling after notifying the administrator. Since an event and action are triggered immediately when the specified condition is met, administrators do not have to wait for a crontab schedule to kick off before noticing a change in the environment. This is very important for allowing quick response.

Discovering drastic changes in logs or other monitored events is an obvious sign of trouble, including numerous failed attempts to break in to a system or a denial of service attack. One type of denial of service attack is a disk full attack. If, for example, /tmp fills, many actions will not be able to take place, since /tmp is often the default temporary space for applications. If a file system is used for shared file access, filling the file system renders it unusable to the user community. This disrupts work and could cause loss of money and reduce productivity. If the file system that contains security logs is filled, no additional information can be logged. The administrator of the system should immediately determine the cause of the problem and resolve the issue. Even if the file system is not filling as a result of an attack, an administrator needs to keep security logging running for auditing purposes.

Among the parallel commands that come with PSSP are *psyslrpt* and *psyslclr*. These commands use Kerberos like the other parallel commands, so they are relatively secure. An administrator can run these and any of the other many parallel commands from one server and obtain information from remote servers without .rhosts or hosts.equiv files.

The psyslrpt command is used to gather information from syslog files on remote servers. This simplifies the task of gathering syslog information from multiple servers. It also gathers the information to a central location for analysis and long-term archiving. Syslog remote logging can be enabled on servers to do a similar task, but the psyslrpt command does not leave evidence of the remote logging. A good hacker would more than likely look in the /etc/syslog.conf to cover

his/her tracks and would notice remote logging to the log host.

The syslog is very important for gathering information that could lead to the discovery of a security breach and possibly assist with the identification of the hacker. The TCP Wrappers program, for example, can log attempts to use services on the host and log the success or failure of the attempt and the location from which the attempt was made. Telnet and ftp are services that can be monitored by TCP_Wrappers.

When logs become unruly, they become difficult to review. The psyslclr command can be used to trim syslogs the same way psyslrpt is used to gather the information from the logs.

File Auditing

After an intruder gains access to a system, they may try to throw administrators and auditors off their trail by installing a "root kit." A root kit is a package of commands that are named like basic Unix commands, but have the purpose of hiding the damage done by a hacker. In other words, a hacker will replace a basic command, like netstat, with the root kit version so an administrator who runs that netstat command will see normal results, not the true results.

When a thief plans a heist of a jewelry store, he may have a plan similar to that of a root kit. After breaking into the store, he may disable the video cameras, open or alter the back door for other thieves or for a return heist later, and wear gloves the entire time to avoid leaving clues. The Tornkit, a popular root kit, uses this plan as well.

Like the thief who disables video cameras, the Tornkit software first disables logging. By doing this, the intruder's activities are more difficult to track To check for remote monitoring, such as remote syslogging, the Tornkit may check for the @ symbol in the /etc/syslogd.conf. The @ symbol is used to designate a remote host where certain syslog messages will be sent. Tornkit also replaces daemons, including sshd and fingerd. The Tornkit may also attempt to activate disabled telnet, shell, and finger daemons by removing the # symbol from the beginning of the corresponding statement in the /etc/inetd.conf. When inetd is refreshed, those daemons could open more doors. Programs on the system, including login, the finger daemon, the ssh daemon, ls, netstat, ps, ifconfig, top, du, and find are replaced by root kits. The altered fingerd daemon opens another port, which basically unlocks or alters the lock to the back door. The other commands installed by the root kit hide the presence of any processes, files, or network connections owned by the Tornkit. When an administrator uses the common commands replaced by Tornkit, they seem to work properly, but they are actually hiding evidence of the break-in. This is similar to a thief wearing gloves.

This thief goes a little farther. Tornkit starts a password sniffer in the background. Since Tornkits are installed by taking advantage of other security holes, not necessarily having the root password, gathering passwords can make break-ins easier or provide a key to getting into other systems on the same network.

To combat these tactics, IBM's AIX comes with an installable product called Trusted Computing Base (TCB). It is similar to the popular product Tripwire. TCB can be used to check a file's permissions, checksum, size, and other attributes. If TCB is run against a file that is altered in any way, it will be detected. For example, if a hacker were to install a root kit that replaced the netstat command with another version, the tcbck command will show the results below.

```
[root@server1]# tcbck -n /usr/sbin/netstat
3001-025 The file /usr/sbin/netstat has the wrong file group.
3001-027 The file /usr/sbin/netstat has the wrong TCB attribute value.
3001-023 The file /usr/sbin/netstat has the wrong file mode.
3001-028 The file /usr/sbin/netstat has the wrong checksum value.
3001-049 The file /usr/sbin/netstat has the wrong file size.
```

Root kit software may be able to present the trojan copies so they appear to be the same as the original file. They might even try to make the trojan programs the same size or at least appear to be the same size, but altering a file while maintaining the checksum is not an easy task by any means. The checksum calculated by the TCB software is calculated using the *sum -r* command. Changing only one character in a file will change the checksum dramatically.

TCB must be chosen to be installed during the installation of the AIX software. If this option is not chosen, the operating system must be reinstalled in order to take advantage of the TCB programs. These programs include tcbck, usrck, pwdck, grpchck, and others. These programs can be run interactively, and they can be used in a script to run checks periodically, log the results, and notify an administrator if problems are identified.

The *tcbck* command is used to check the security of a file, directory, or device based on the parameters in the sysck database (sysck.cfg). Options can be used to check individual files, classes of files, or all files. The usrck command verifies user definitions exist for each user in the /etc/security/passwd, /etc/security/limits, and /etc/security/user files. It also verifies that each group in the /etc/group file has a stanza in /etc/security/group. The grpck command confirms that all members of groups are actually users in the user database. All of these commands have options to report errors, but not fix them; report and fix errors; fix errors, but not report them; and report errors and ask if they should be fixed.

After installing TCB, an administrator should schedule a script to run the various commands on a consistent basis. The TCB does not do this on its own. Simply installing TCB does not result in any checking or notification to administrators if changes are notified. Instead of running checks against all files on the system, an administrator could choose to run the tcbck command, for example, against only the files that are commonly replaced by a root tool kit. Or, an administrator could choose a class or subset of commands to check. In addition to checking file integrity on a scheduled interval, an administrator should check user accounts and group accounts for consistency. Whatever interval is chosen to run the commands, notification should be included in the logic. The errors of these commands could be piped to a temporary file that is mailed to the administrators if it is not empty. For more immediate notification, a paging program could be used to page an administrator, especially if it is alpha-numeric.

Conclusion

The Cluster 1600 environment is not only an attempt by IBM to provide standard tools and administration across multiple hardware platforms, but the expansion of the PSSP realm of control is an excellent way to standardize security practices. PSSP addresses many of the areas of exploitation by intruders. In larger more complex environments, security practices are even more difficult to manage. Logs must not only be collected, but reviewed and analyzed. An effective way of gathering them and organizing the information is, therefore, important. Remote administration can provide opportunity to intruders if access is not managed properly. While trying to maintain security, users and managers are often yelling about business needs and the "inconveniences" of security. PSSP even included a way to securely centralize password management and maintenance. A user changes his/her password in one place. With the constant development of intrusion tools used by hackers, holes almost inevitably exist. That is why it is important to monitor the system for abnormalities. The Trusted Computing Base is like a home security system for the systems. If used properly, changes in files and programs on a system can be detected and corrected. PSSP and the Cluster 1600 environment address these areas and many more. It is a super duo that is sure to be a success.

References

"AFS Frequently Asked Questions." 9 Jul. 1998. Version 1.1.3.
URL: http://www.angelfire.com/hi/plutonic/afs-faq.html. (7 Dec. 2001).

CERT. "Widespread Exploitation of rpc.statd and wu-ftpd Vulnerabilities." CERT Incident Note IN-2000-10. 15 Sep. 2000.
URL: http://www.cert.org/incident_notes/IN-2000-10.html. (5 Jan. 2001).

Garfinkel, Simon and Gene Spafford. Practical Unix Security. Sebastopol: O'Reilly, 1994. 142, 236-237, 319, 322, 335.

International Business Machines Corporation. eserver Cluster 1600 Planning, Installation, and Service. 2d ed. n.p.: IBM, 2001.

International Business Machines Corporation. Parallel System Support Programs for AIX Administration Guide Version 3 Release 4. 4th ed. n.p.: IBM, 2001.

International Business Machines Corporation. Parallel System Support Programs Installation and Migration Guide Version 3 Release 4. 4th ed. n.p.: IBM, 2001.

International Business Machines Corporation. RS/6000 SP Planning Volume 2, Control Workstation and Software Environment. 7th ed. n.p.: IBM, 2001.

"Managing logging and other data collection mechanisms." 1 May 2000.
URL: http://www.cert.org/security-improvement/practices/p092.html. (5 Jan. 2002).

"May 1, 1997: Ancient game, modern masters." 1 May 1997.
URL: http://www.research.ibm.com/deepblue/press/html/g.1.2.html (5 Jan. 2002).

Rautiainen, Sami. "Tornkit." F-Secure Virus Descriptions. March 2001.
URL: http://www.europe.f-secure.com/v-descs/torn.shtml. (7 Dec. 2001).