Global Information Assurance Certification Paper

BA VAN NGUYEN
Practical Assignment for
Level One Security Essentials

HOW TO SET UP A SECURE SERVER WITH SUN SOLARIS In Our Environment

Introduction/Background

When choosing the topic for my practical assignment, I intend to choose something that I can use in my workplace. (I hope SANS gives me credit for applying one of their goals: SANS would not teach you anything which you cannot use the next day, or something to that effect). The company I work for, bought two new E5500 servers: one to replace our current E5000 finance server and one to run the newly acquired Human Resource software. About four years ago, when we acquired the E5000 to run finance software, we did not pay much attention to security of the server and its data as there were much more important things (I think) to do such as stabilizing the server which seemed to always require more storage, memory, and CPU to support the application and the gradually-added users, the network which was still in the building period, and the financial application which required a lot of customization to meet the need of our business.

Nowadays our company is very much aware of the security issues and the importance of CIA or Confidentiality, Integrity, and Availability of data. We also have many in-house experts who maintain and secure LAN and WAN. The E5000 server is much more stable. With the two new servers, we have a chance to do all over again. In addition to optimizing the layout of file systems for the Sun Solaris and the database for upgrade and growth, we have an opportunity to put in place all security features best applicable in our own environment so as we can make the servers secure from the beginning.

Because a secure server is the result of installing technical features and user awareness, this paper has two parts: The Things to Secure the Server and User Education.

Things to Secure the Server

To have a secure place for the server is the first thing we should do. Simpson Garfinkel and Gen Spafford in Practical Unix & Internet Security call it "One Forgotten Threat". They mention one New York investment firm spent ten of thousand of dollars on computer security measure to prevent break-in during the day but forgot that the janitors left the door of the computer room wide open when he mopped the computer room's floor at night (1).

No one should use the console. Any one who needs access to the server in X-window mode should have X-term client software on the user's PC like Hummingbird. One reason to limit physical access to the server is that a regular can use Stop-A to access Open Boot and reboot the system or change many environment variables. We also need

to protect the Open Boot parameters by setting its security level to "command" (all Open boot commands require password except *boot and go*) or to "full" (all Open Boot commands require password except *go*).

According to Alex Noordergraaf and Keith Watson, one way to reduce system vulnerabilities is to minimize the amount of software on the server. They add that a majority of intrusions has been done through the holes in the Operating System. There 4 selections of the installation clusters of Solaris: the core has 39 packages and needs 53 Mbytes, The End User 142 packages and 242 Mbytes, the Developer 235 packages and 493 Mbytes. (2). Our server is set up for End User, but some development is needed. Minimization is an effective security measure against intrusion because system and application were not originally designed to withstand hackers' attack. According to Peter Gregory, client/server application designed for intranet level security has been the target for attack as they are exposed to the Internet. Another reason for software vulnerability is that many applications are very complex and it is impossible to make a complex program secure. Sendmail program is a case in point. Sendmail design has many flaws and hackers have exploited these flaws over years. Even though many security holes of Sendmail have been plugged, hackers continue to find more weaknesses in the program. (3)

Immediate after installing Solaris, we should download and apply all SUN recommended patches, which include security ones. This is the one thing we often forget after the server goes into operation (in other words, it suffers from entropy). We need to put in our schedule to check it monthly. Other thing we need to mention here is that the size of the system's filesystems such as /, /var, /opt … must be big enough to accommodate patches and upgrade over years.

The University of Oregon's security paper (4) recommends installing TCP Wrappers to control inetd daemons based on IP addresses. We will install TCP Wrappers, but we also need to do one more thing because of the application that we use on the server. Our application is a client/server application. When a user run the application it asks for Unix login and Unix password. However it does not read .login file or the like. It means that our users can log in outside the application and get access to the server. To prevent this disastrous potential, we create home directories for users by location and modify .login file to have only one line: exec exit. The application does not read .login so there would be no impact on the application users. But if the user attempts to telnet to the server outside the application his or her access will be denied. So far this method has worked well on the old server. Jamie Wilson on "Securing Your Solaris Server" recommends replacing daemons, which are spawned from inetd with standalone daemons. In other words, these daemons can start up and listen themselves. He also recommends installing SSH server which has all the features cited above.

Ftp is needed for our budget module where user can download budget data, do maintenance on upload to the server. Ftp is also needed for our interface programs where Merchandising and Human Resources data is imported to finance system. In our old system, we use /etc/ftpusers (including root) to log off all users who do not need ftp. Our Security Officer wants to install Proftp to replace the one on the system. According to Jamie Wilson, Proftp would allow us to specify which port to bind to, to disallow file or directory overwriting. With plugins, we can create ftp users without creating a real user account.

Dan Farmer and Wietse Venema say that finger command is a powerful tool for hackers to get information, especially finger with @, "0", and "". (5). Hacker can use finger to collect user information throughout the network. We do not have any need for finger, so we would turn it off.

The University of Oregon emphasizes that many problems mentioned in SANS Top 10 Vulnerabilities were caused by buffer overflows and recommend to add these two lines to /etc/system:

Set noexec_user_stack = 1
Set noexec_user_stack_log = 1 (6)

Our finance application interfaces with scanning device to process invoices. The scanning software on client PC uses NFS to store images. Even though all SUN security experts discourage NFS, we need to have one NFS for the scanning interface. However we will monitor its access closely.

Passwords for the users are our big concern as the system will handle more than 1500 users located around the U.S and one foreign country. First of all we have to create user login names in a way that we can identify where they log in. Second the passwords need to expire often enough to protect the logins, but not too often to the point that we have assigned someone to help in changing passwords because, of 1500 users, one third of them cannot change their password regardless how detailed our instructions are. Some users even request that their passwords would never expire because they have too many passwords to remember. Of course we politely say that for security reason we cannot do that. We advise them that they should not choose their password with something associated with their name or their family and each letter would be a beginning letter of a phrase that help them remember it. Kerberos and PAM are highly recommended by the SUN (5). Kerberos provides a secure way for a client to authenticate through the network, exchange shared encryption key after authentication and controls all resources to prevent misuse of logins. PAM allows a single login to get access to multiple secure services. We will be looking into implementing Kerberos because Kerberos protocol is already part of Solaris.

Su is another concern. We will create a su group. Only the member of su group can do su. We would make sure that root has umask 077 and it should have its search path in /usr/bin:/sbin:/usr/sbin.

Set-uid and set-gid programs can cause many security problems. The Solaris Security FAQ (8) recommends that we should a list of all these programs and check each of them to see if un-sid or un-gid it would have any impact. We will have a list of those by find */ -perm 4000 (set-uid) and 2000 (set-gid)* and check each of them. At a glance, mailx, write, wall can be un-gid and cu, ct … can be un-uid.

In our old system, we use cron jobs to run many database tuning programs to check available space and growth rate of many important tables of our database. We also use cron jobs to run many interface programs when data are available, usually late into the night. Although cron jobs play a critical part in our system, we have not used cron.allow and cron.deny to control who can use cron. Because a false modification of our cron jobs will compromise the integrity of our system.

As recommended by many experts, we need to set up regular auditing schedule

using many free tools such as:

ASET: a Solaris software package, to check permissions and contents of system files.

COPS: to uncover known weaknesses in the system like world writable Filesystems or directories.

Tripwire: it has a configurable configuration file. Tripwire reads the Configuration file to find out which files and directories need to check against its baseline database. The difference will be listed in a report for validation. According to Seth T. Ross, Tripwire has the best attributes of A good security tool. It is tamper-proof, portable, configurable, and scalable. (9)

Finally we need to set up the backup and restore procedure for this server so that it can be restored quickly if a disaster strikes. The system will be running 24 by 7. We need to find a window to do a full system backup at least once a week. The system should be backed up incrementally on the other days of the week.

To facilitate system restore, Peter Gregory (10) recommends:

- To keep high degree of segregation between Operating System partition and partitions for Applications.
- Do not alter contents of /usr/bin and /usr/sbin and put all local tools in separate directory.

The restore procedure would only include a series of restore of all settings for the site and applications after OS is loaded, not a series of re-installing all tools and applications.

User Education

We are in the support function of our organization. For this reason, supporting users are very important to us. To make a secure server that is not too unfriendly for users to work with are not difficult if users follows some guideline as we follow some guidelines to maker the server safer.

Many users do not want their passwords to expire. We will not do that because so many users left and their supervisors never report to us. So many users still use their names or their family members for their passwords. We will send out instructions on how to choose password. Also users should follow some basic procedure such as lock their PC when they are away from their desk a long period of time and log off and shutdown their PC before going home. The application allows them to run reports or post their General Ledger entries using background process. They should use this feature so they can turn off their machines before going home. Other procedure is that supervisors must notify us when employees leave so we deactivate the employees' logins so that all logins can be accountable for. All logins, which have not been used in 30 days, will be locked.

System administrator also needs to be careful on everything he does to the system. He should have a regular account to login and su to root when he needs it because the system needs protection from intentional destruction as well as from careless

mistakes.
He should automate all routine tasks and setup notification routines so he can be notified when something in the system goes wrong. Kirk Waingrow mentions in his book (11) that tuning is a troublesome issue for system administrator. When he thinks he tunes one thing, he might break other. Waingrow recommends that the system administrator should have plan to test, to back out if the change does work. He also needs to consider the impact of the change on the users in term of the timing of the change. To use Seth Ross's words: he needs to "do it right before someone does it wrong for you"(12).

He also needs to monitor disk space and the general performance of the system so he can prepare for growth.

Waingrow also stresses that poor system administration practices are similar to open doors to intruders. Some basic but critical tasks are:

- Do not allow empty passwords.
- Clean up old accounts. In our case we have go after supervisors for unused logins.
- Pass out root password
- Ignore users so long that they attempt to the job themselves. (13).

Conclusion

I do not present a how to sep up a secure Solaris server in general, but a safe Solaris server based on our environment because there are good security features that can cause difficulties for users and developers in certain situation.

To us, a secure system is a reliable system where data is kept securely and users can finish their work timely.

No one person can maintains a secure server. It must be maintained by all people involved, from the system administrator, developers, and operators to everyday users of the system. I strongly believe that awareness by all people that security is a serious issue would go a long way to achieve Confidentiality, Integrity and Availability of our company data and systems.

The procedure to make a server secure needs to be revised regularly in order to keep up with changing environment which surely will present many new threats to the system. Keeping a server secure is a long-term project with all participation or awareness of all people using or maintaining the server.

I hope with all the preparation, our new two servers would be secure and reliable as they are supposed to.

Footnotes

(1) Garfinkel, Simpson and Spafford, Gene. **Practical Unix & Internet Security**.
O'Reilly & Associates, Inc. 1996. Page 357.

(2) Noordergraaf, Alex; Watson, Keith. "Solaris Operating Environment Minimization For Security: A Simple Reproducible and Secure Application Installation Methodology". Global Enterprise Security Service.
http://www.sun.blueprints. (16 April 2001). Page 2-3.

(3) Gregory, Peter. **Solaris Security**. Sun Microsystems Press/Prentice Hall, Inc. 2000. Page 6-9.

(4) University of Oregon. " Solaris Installation Secure Standard Practices".
http://ns.uoregon.edu/security/solaris.html (17 April 2001).
Page 3-5.

(5) Farmer, Dan; Venema, Wietse. "Improving the Security of Your Site by Breaking into it".
http://www.nsi.org/Library/Compsec/farmer.txt (17 April 2001).
Page 4-5.

(6) University of Oregon. "SANS Top 10 Vulnerabilities A Recent Survey".
http://ns.uoregon.edu/security/topten.html. Page 1.

(7) Sun Microsystems. "Security Technologies"
http://www.sun.com/security/technologies.html (11 April 2001).
Page 2.

(8) The S Galvin, B., Peter. "The Solaris Security FAQ". SunWorld.
http://ns.uoregon.edu/security/sun-sec-faq.html (17 April 2001).
Page 6.

(9) Ross, T. Seth. **Unix System Security Tools**". McGraw-Hill. 2000.
Page 152-153.

(10) Gregory, Peter. **Solaris Security**. Sun Microsystems Press/Prentice Hall, Inc. 2000. Page 220.

(11) Waingrow, Kirk**. Unix Hints & Hacks**. QUE Corporation. April 1999.
Page 10-11.

(12) Ross, T. Seth. **Unix System Security Tools**". McGraw-Hill. 2000. Page

23.

(13) Ross, T. Seth. **Unix System Security Tools**". McGraw-Hill. 2000. Page 96.

Bibliography

Internet Sources/URLs

Noordergraaf, Alex; Watson, Keith. "Solaris Operating Environment Minimization For Security: A Simple Reproducible and Secure Application Installation Methodology". Global Enterprise Security Service. URL: http://www.sun.blueprints. (16 April 2001).

University of Oregon. " Solaris Installation Secure Standard Practices". URL: http://ns.uoregon.edu/security/solaris.html (17 April 2001).

Wilson, Jamie. "Securing Your Solaris Server". Unix Insider 2/16/01. URL: http://www.itworld.com/Comp/2377/UIR010216hardening/ (17 April 2001).

Farmer, Dan; Venema, Wietse. "Improving the Security of Your Site by Breaking into It". URL: http://www.nsi.org/Library/Compsec/farmer.txt (17 April 2001).

Sun Microsystems. "Security Technologies" URL: http://www.sun.com/security/technologies.html (11 April 2001).

Galvin, B., Peter. "The Solaris Security FAQ". SunWorld. URL: http://ns.uoregon.edu/security/sun-sec-faq.html (17 April 2001).

University of Oregon. "SANS Top 10 Vulnerabilities A Recent Survey". URL: http://ns.uoregon.edu/security/topten.html. (17 April 2001).

Printed Works (Books)

Gregory, Peter. **Solaris Security**. Sun Microsystems Press/Prentice Hall, Inc. 2000.

Waingrow, Kirk**. Unix Hints & Hacks**. QUE Corporation. April 1999.

Ross, T. Seth. **Unix System Security Tools**". McGraw-Hill. 2000.

Garfinkel, Simpson and Spafford, Gene. **Practical UNIX & Internet Security**. O'Reilly & Associates, Inc. 1996.

Last Updated: May 28th, 2017

Upcoming Training

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