Eghia Aramian
GSEC 1.4b
Nessus on a Solaris 8

Nessus is a free, powerful easy to use open-source network vulnerability scanner. This security scanner will remotely audit any given network and determine whether it is vulnerable to any misuse or break-in. Nessus is designed in two parts, a server, which runs the security scans, and a client front-end that configures your scans and collects the results. The server runs on Linux, BSD, and Unix Boxes such as Solaris. There are many client versions, a version that works with GTK and others written for the Windows platforms [1]. This paper will discuss the setup of Nessus on a Solaris 8 System and the Nessus client on a Windows XP box.

Before Nessus can be installed on the Solaris 8 system it first be secured from attacks against it. This must be done to ensure that Nessus and other tools on the system cannot compromise your network. This is a very sweet target for an attacker to try to compromise. If it was compromised then the whole network can be mapped out for attacks at leisure. In addition, this would be a great launching pad for attacking many other networks. This would put your company and you at great risk of having legal actions taken upon you. To ensure this does not occur, this paper will go through the minimum steps necessary to secure the Solaris 8 system:

- Patching the OS
- Minimizing Network service in /etc/inet/inetd.conf
- Kernel, Network, and Memory Tuning
- Logging
- File/ Directory Access/ Permissions
- System Access, Authentication, and Authorization
- User Accounts and Environment
- Install KEY Security Tools

Once the Solaris 8 system is secure then we will go through the steps to install and configure the Nessus server and the Nessus client.

Patching the OS

The first thing you must do before hardening the Solaris OS is to get the latest updates and patches. To get the latest Sun recommended patches download it from ftp://sunsolve.sun.com/pub/patches/8_x86_Recommended.zip after downloading the file you need to execute the following commands[2]:

```
cd /(directory file is downloaded to)
unzip –qq 8_x86_Recommended.zip
cd 8_x86_Recommended
./install_cluster –q –nosave
```

While the patch cluster is installing there may be some patches that do not install. There will be codes shown when patches attempt to get installed; code 2 is given if the patch is already installed; code 8 is shown when the patch is for an operating system package that is not installed. If any other code is presented then check the log file located at

/var/sadm/install_data. The –nosave option installs the patch cluster with no backup and no uninstall is available [2].

Minimizing Network service in /etc/inet/inetd.conf

The file /etc/inet/inetd.conf starts up a number of network related services. Many of these services are ones that are rarely used and all of them have at least one or more reported vulnerabilities. It is also cluttered with comments that can make the file difficult to read and maintain. The best approach if you must connect to the computer over the network is to run SSH and completely disable the /etc/inet/inetd.conf. If you need to run /etc/inet/inetd.conf then ensure that only the required mission-critical services are running and TCP-Wrappers configured to protect them. The following commands will rename inetd.conf to a back up file which you can still reference if needed. Then a new empty inetd.conf file is created and the proper permissions are set. [3],[2]

```
cd /etc/inet
mv inetd.conf inetd.conf.orig
touch inetd.conf
chown root:sys inetd.conf
chmod 444 inetd.conf
```

Minimizing Boot/Startup services

There are many services that are started at boot time that do not need to be. A good rule of thumb is. "It you don't need it, turn it off. If you are not sure whether you need it or nor, turn it off and see what breaks [3]. These services are started by start up scripts located in the /etc/rc*.d directories. To disable them you only need to rename them that way you can restore them if necessary.

ex. mv /ect/rc3.d/S15nfs.server /etc/rc3.d/_S15nfs.server (this starts the NFS server daemon) [2].

Kernel, Network, and Memory Tuning

It would be prudent to disable core dumps for several reasons. The core dumps contain an image of the memory used at the time of the dump. This dump can be used to troubleshoot program errors. These files can easily used up disk space and can contain sensitive information. To disable this use the following command [4]:

```
set sys:coredumpsize = 0
```

There are security exploits that take advantage of the Solaris OE kernel executable stack. This can be avoided by adding a few lines to the /etc/system file.

```
set noexec_user_stack = 1
set noexec_user_stack_log = 1
```

Next, you must configure the network parameters by creating a script that will run at boot time.

Here is the script you should run [2]:

```
cat <<END_SCRIPT >/etc/init.d/netconfig
#!/sbin/sh
ndd -set /dev/ip ip_forwarding 0
ndd -set /dev/ip ip6_forwarding 0
```

```
     ndd -set /dev/ip ip_strict_dst_multihoming 1
     ndd -set /dev/ip ip6_strict_dst_multihoming 1
     ndd -set /dev/ip ip_send_redirects 0
     ndd -set /dev/ip ip6_send_redirects 0
     ndd -set /dev/ip ip_ignore_redirect 1
     ndd -set /dev/ip ip6_ignore_redirect 1
     ndd -set /dev/ip ip_forward_src_routed 0
     ndd -set /dev/ip ip6_forward_src_routed 0
     ndd -set /dev/ip ip_forward_directed_broadcasts 0
     ndd -set /dev/tcp tcp_conn_req_max_q0 4096
     ndd -set /dev/tcp tcp_ip_abort_cinterval 60000
     ndd -set /dev/ip ip_respond_to_timestamp 0
     ndd -set /dev/ip ip_respond_to_timestamp_broadcast 0
     ndd -set /dev/ip ip_respond_to_address_mask_broadcast
0
     ndd -set /dev/arp arp_cleanup_interval 60000
     ndd -set /dev/ip ip_ire_arp_interval 60000
     END_SCRIPT
     chown root:root /etc/init.d/netconfig
     chmod 744 /etc/init.d/netconfig
     ln -s /etc/init.d/netconfig /etc/rc2.d/S69netconfig
```

Logging

Solaris systems do not capture logging messages sent to the LOG_AUTH by default. You must create the file /var/adm/authlog, which is only readable by the superuser. You must also modify the SYSLOG_FAILED_LOGINS parameter in /etc/default/login to control the amount of failed logins allowed before log messages are generated[2]. You should set it to zero then all failed attempts will be logged.[3]

```
     echo "auth.info\t\t\t/var/log/authlog" >>/etc/syslog.conf
     touch /var/log/authlog
     chown root:root /var/log/authlog
     chmod 600 /var/log/authlog

     touch /var/adm/loginlog
     chown root:root /var/adm/loginlog
     chmod 600 /var/adm/loginlog
```

File/ Directory Access/ Permissions

You must protect the file system from having unauthorized programs, especially set-UID programs placed there. Wherever you have programs stored, such as /usr, /opt/ /usr/local, you should have them mounted as “ro” read-only. If you need to install updated, patches, or other software the run the command: mount –o remount, rw /usr. Then reboot when done and it will be read only again. The root file system should have logging enabled. This will not allow attackers that have physical access to the console the ability to compromise the file system [2], [3].

System Access, Authentication, and Authorization

There are several files that you will need to configure to ensure tighter access control to your system. First, you should never have root accessing your system over the network. You should access the system thru an unprivileged account and su. In /etc/default/login make sure that CONSOLE=/dev/console is set. Check your password file and remove accounts that are not used such as uucp, nuucp, and smtp users. There are other system accounts such as adm, daemon, bin should have their shells set to /dev/null to block access.[3] Edit the /etc/pam_conf and remove any lines containing rhosts_auth. If SSH is configured check your sshd_config does npt allow .rhosts. Create cron.allow and at.allow files that list authorized users of cron and at that can add, modify, and delete cron jobs[3].

Install KEY Security Tools

Several other packages need to be installed to harden Solaris. You need to install and setup SSH. OpenSSH is freely distributed and you should use it as an alternative to using Telnet, FTP, or any other need to access the system over the network. You can obtain it precompiled from http://www.sunfreeware.com you need to choose which platform you are using as they have packages available for Intel and Sparc [2]. TCP Wrappers is a must to control the various network services based on the IP address of the remote connection. NTP, network time protocol, is very important if you plan to investigate any security incidents [3]. Fix-modes is a great free program that will go through your system and set all the appropriate permissions on the various OS files and directories [3].

Software Packages required for Nessus and Nessus Installation

Now that they operating system environment is secure, you can continue to prepare the system for Nessus by loading some third party software packages required by Nessus. The following is from Security Scanner 'Nessus' on Solaris by John Richardson, which can be found at http://www.sunhelpdesk.com/users/john/nessus.htm. This has been updated slightly to work on Solaris 8.

Gzip Compression Utility

Download the pre-compiled packaged version of gzip found at SunFreeWare.com

```
/opt# pkgadd -d ./gzip-1
```

Note: gzip automatically gets installed in /usr/local/bin. You will want to add /usr/local/bin to your environment PATH

Gcc GNU Compiler(Optional: that is if you don't have a capable compiler. The standard C compiler will not work)

Download the pre-compiled packaged version of gcc found at SunFreeWare.com

```
/opt# gunzip gcc-2.95.3-sol8-intel-local.gz
/opt# pkgadd -d ./ gcc-2.95.3-sol8-intel-local
```

Note: gcc automatically gets installed in /usr/local/bin. I symbolically linked /usr/ucb/cc to /usr/local/bin/gcc so that the system will use gcc as its default compiler.

```
/opt# mv /usr/ucb/cc /usr/ucb/cc.bak
ln -s /usr/local/bin/gcc /usr/ucb/cc
```

Gimp ToolKit(GTK)

You will need to install Gimp manually. You can obtain the source from ftp://ftp.gimp.org/pub/gtk. You will need both the glib and gtk tar source files. Nessus needs some of the header source files not included with the pre-compiled installation of gtk found at sunfreeware.com.

```
/opt# mkdir gtk
/opt# cd gtk
NOTE: Put latest releases of gtk and glib here
/opt/gtk# gunizp *.gz
/opt/gtk# tar xvf glib-1.2.10.tar
/opt/gtk# cd glib-1.2.10
/opt/gtk/glib-1.2.10# env LDFLAGS="-L/usr/local/lib -R/usr/local/lib" ./configure
/opt/gtk/glib-1.2.10# make
/opt/gtk/glib-1.2.10# make install
/opt/gtk/glib-1.2.10# cd ..
/opt/gtk# tar xvf gtk+-1.2.10.tar
/opt/gtk# cd gtk+-1.2.10
/opt/gtk/gtk+-1.2.10# ./configure
/opt/gtk/gtk+-1.2.10# make
/opt/gtk/gtk+-1.2.10# make install
```

Nmap Port Scanner

Download the pre-compiled packaged version of Nmap 3.00 found at SunFreeWare.com

```
 opt# gunzip nmap-3.00-sol8-intel-local.gz
/opt# pkgadd -d ./ nmap-3.00-sol8-intel-local
```

Nessus

Download Nessus at ftp://ftp.nessus.org/pub/nessus/nessus-1.2.7/src/ or one of their mirrors. Make sure to download the following:

- libnasl-1.2.7.tar.gz
- nessus-core-1.2.7.tar.gz

- nessus-libraries-1.2.7.tar.gz
- nessus-plugins-1.2.7.tar.gz

Add the following to your profile(not necessary if already have env variable set previously):

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
export LD_LIBRARY_PATH
# . ~/.profile                              <== Sources the profile
```

IMPORTANT: You must compile Nessus in the following order

Nessus Libraries

```
/opt/Nessus# gunzip nessus-libraries-1.2.7.tar.gz
/opt/Nessus# tar xvf nessus-libraries-1.2.7.tar
/opt/Nessus# cd nessus-libraries
/opt/Nessus/nessus-libraries# ./configure
/opt/Nessus/nessus-libraries# make
/opt/Nessus/nessus-libraries# make install
```

Libnasl

```
/opt/Nessus# gunzip libnasl-1.2.7.tar.gz
/opt/Nessus# tar xvf libnasl-1.2.7.tar
/opt/Nessus# cd libnasl
/opt/Nessus/libnasl# ./configure
/opt/Nessus/libnasl# make
/opt/Nessus/libnasl# make install
```

Nessus Core

```
/opt/Nessus# gunzip nessus-core-1.2.7.tar.gz
/opt/Nessus# tar xvf nessus-core-1.2.7.tar
/opt/Nessus# cd nessus-core
/opt/Nessus/nessus-core# ./configure
/opt/Nessus/nessus-core# make
/opt/Nessus/nessus-core# make install
```

Nessus Plugins

```
/opt/Nessus# gunzip nessus-plugins-1.2.7.tar.gz
/opt/Nessus# tar xvf nessus-plugins-1.2.7.tar
/opt/Nessus# cd nessus-plugins
/opt/Nessus/nessus-plugins# ./configure
```

```
/opt/Nessus/nessus-plugins# make
/opt/Nessus/nessus-plugins# make install
```

Now you must create a nessusd account on the server. The Nessus server maintains its users database, each user having a set of restrictions. This allows you to share a single nessusd server for a whole network and different administrators who will only test their part of the network[8].

The utility *nessus-adduser* takes care of the creation of a new account :

```
$ nessus-adduser

Addition of a new nessusd user
------------------------------

Login : Joshua
Password : password
Authentification type (cipher or plaintext) [cipher] : cipher
Now enter the rules for this user, and hit ctrl-D once you are done :
(the user can have an empty rule set)
^D

Login          : Joshua
Pssword          : password
Authentification : cipher
Rules          :

Is that ok (y/n) ? [y] y

user added [8].
```

The third step is to start Nessus. As root you will enter the command nessusd –D. This will start the Nessus daemon.

Nessus Client WX version 1.4.2

You must download a version of the Nessus client for you to be able to configure the vulnerability scans. Scans can be configured at the command line but it will not be discussed here. The latest Nessus client version is NessusWX version 1.4.2 which can be downloaded from http://nessuswx.nessus.org/index.htm#download . The client has a windows interface and is easily configurable. the client can connect to the Nessus server using SSL communications or it can connect unencrypted, this will depend on if you have SSL configured on the Nessus server. All the results of the scans and settings for all sessions (configured scans) are saved in a database for future use. The results can be viewed before a report is generated. The reports can be formatted in several different formats, plain text, PDF, and HTML. You can export results into NSR, extended NSR, SQL command formats or directly to MySQL database [7].

Conclusion

Now we have a secure operating system from which to run our vulnerability scans from. This is very important so as not to compromise our network from attackers who may wish to use Nessus to launch attacks, probes on our network or other networks. Nessus is a very powerful and versatile tool to have in our arsenal to protect our networks from harm from outside or within.

Bibliography:

1. Nessus Introduction http://www.nessus.org/intro.html
2. SANS Security Essentials VI: Unix Security, CIS Appendices, Solaris Benchmark v1.0.1b
3. Solaris Security: Step by Step, Hal Pomeranz, Deer Run Associates http://www.deer-run.com/~hal/SolarisWebcast.pdf
4. Solaris Operating Environment Security, Updated for the Solaris 8 Operating Environment. http://www.sun.com.security/blueprints
5. Solaris Operating Environment Network Settings for Security, Updated for the Solaris 8 Operating Environment
6. Security Scanner 'Nessus' on Solaris, John Richardson http://www.sunhelpdesk.com/users/john/nessus.htm
7. NessusWX - Nessus Client for Win32, http://nessuswx.nessus.org/
8. Nessus Demonstrations, http://www.nessus.org/demo/first.html
9. UNIX System Administration Handbook Third Edition, Evi Nemith, Prentice Hall PTR, ISBN: 0-13-020601-6