Foundation for Minimal Solaris 10 Systems

by: Glenn M. Brunette, Jr., 10/19/2004

http://www.securitydocs.com/library/2644

The topic for this article is the Solaris 10 Reduced Networking Software Group (also commonly known as the Solaris 10 Reduced Networking Meta Cluster). This software group is new and joins the five existing software groups available in Solaris today: Core, End User, Developer, Entire and Entire + OEM software groups. The Reduced Networking Software Group is positioned as a subset of Core and represents the smallest amount of Solaris that can or should be installed and have a working and supported system. (Note that for support reasons, it is not advised to remove packages installed by the Reduced Networking Software Group.)

To install the Reduced Networking Software Group, simply select it from the list when doing a graphical installation. If you are using JumpStart, then you should use the *cluster* keyword with the new value *SUNWCrnet*. The following is a sample JumpStart profile that uses the Reduced Networking Software Group. This profile was also used to build the system used as an example in this article.

```
install_type    initial_install
cluster         SUNWCrnet
partitioning    explicit
filesys         rootdisk.s1     768     swap
filesys         rootdisk.s0     free    /
system_type     standalone
```

During the installation process, you will see messages similar to the following:

```
Processing profile
        - Selecting cluster (SUNWCrnet)
        - Selecting all disks
        - Configuring boot device
        - Using disk (c0t0d0) for "rootdisk"
        - Configuring swap (c0t0d0s1)
        - Configuring / (c0t0d0s0)
```

One thing that may draw your attention is the following install-time message:

```
Verifying space allocation
        - Total software size:  152.67 Mbytes
```

Yes, it's true - the size of this installation is just a little over 150-Mbytes. Note that this size is based on the build of Solaris 10 that I was using and will certainly change before Solaris 10 is finalized, but I did want to mention it as an example of how small a Solaris installation can be. By leveraging the Reduced Networking Software Group, you are providing yourself with a solid foundation on which to deploy a minimized platform. So, let's see what we have...

```
# df -k
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/dsk/c0t0d0s0    7929156  164697 7685168     3%    /
```

```
/devices                    0       0       0     0%    /devices
ctfs                        0       0       0     0%    /system/contract
proc                        0       0       0     0%    /proc
mnttab                      0       0       0     0%    /etc/mnttab
swap                   956144     224  955920     1%    /etc/svc/volatile
objfs                       0       0       0     0%    /system/object
fd                          0       0       0     0%    /dev/fd
swap                   955928       8  955920     1%    /var/run
swap                   955920       0  955920     0%    /tmp
```

By the time all is said and done, the installed system is up to 161M. At present, this accounted for about 81 packages. This default configuration includes 28 set-uid programs and 11 set-gid programs. This is all much less than what is typically installed on most systems today. (As noted above, this will certainly change before Solaris 10 is finalized, so don't hold me to those exact numbers.)

What is actually running on this system by default on this system? To answer this question, we look at the output of ps *-aef*:

```
# ps -aef
     UID   PID  PPID   C    STIME TTY         TIME CMD
    root     0     0   0 21:52:19 ?           0:06 sched
    root     1     0   0 21:52:22 ?           0:00 /sbin/init
    root     2     0   0 21:52:22 ?           0:00 pageout
    root     3     0   0 21:52:22 ?           0:01 fsflush
    root   432   376   0 22:31:05 console     0:00 ps -aef
    root     7     1   0 21:52:24 ?           0:03 /lib/svc/bin/svc.startd
    root     9     1   0 21:52:24 ?           0:16 svc.configd
    root   394   385   0 22:00:00 ?           0:00 /usr/lib/saf/ttymon
  daemon   335     1   0 21:53:40 ?           0:00 /usr/sbin/rpcbind
    root   340     1   0 21:53:40 ?           0:00 /usr/sbin/keyserv
  daemon   279     1   0 21:53:27 ?           0:00 /usr/lib/crypto/kcfd
    root   376     7   0 21:59:59 console     0:00 -sh
    root   278     1   0 21:53:26 ?           0:00 /usr/sbin/nscd
    root    79     1   0 21:52:46 ?           0:00 /usr/lib/sysevent/syseventd
    root   411     1   0 22:00:03 ?           0:00 /usr/lib/fm/fmd/fmd
    root   367     1   0 21:59:58 ?           0:00 /usr/lib/utmpd
    root   385     7   0 22:00:00 ?           0:00 /usr/lib/saf/sac -t 300
    root   389     1   0 22:00:00 ?           0:00 /usr/sbin/syslogd
    root   395     1   0 22:00:00 ?           0:00 /usr/lib/inet/inetd start
    root   397     1   0 22:00:00 ?           0:00 /usr/sbin/cron
```

As you can see, really only the bare minimum. This is also confirmed by our look at those network ports that are in use as reported by *netstat -an*:

```
# netstat -an

UDP: IPv4
   Local Address         Remote Address     State
-------------------- -------------------- -------
      *.111                                Idle
      *.*                                  Unbound
      *.32772                              Idle
      *.514                                Idle
      *.*                                  Unbound
```

```
TCP: IPv4
   Local Address        Remote Address    Swind Send-Q Rwind Recv-Q  State
-------------------- -------------------- ----- ------ ----- ------ -------
      *.*                  *.*                0      0 49152      0 IDLE
      *.111                *.*                0      0 49152      0 LISTEN
      *.*                  *.*                0      0 49152      0 IDLE


TCP: IPv6
   Local Address         Remote Address      Swind Send-Q Rwind Recv-Q   State
-------------------- ---------------------- ----- ------ ----- ------ ----------- -
      *.*                    *.*               0      0  49152      0  IDLE

SCTP:
   Local Address       Remote Address    Swind  Send-Q Rwind  Recv-Q StrsI/O  State
-------------------- -------------------- ------ ------ ------ ------ ------- -------
0.0.0.0              0.0.0.0                 0     0   102400      0  32/32  CLOSED

Active UNIX domain sockets
Address  Type          Vnode     Conn  Local Addr      Remote Addr
30001307e08 stream-ord 30001292a80 00000000 /var/run/.inetd.uds
```

As you can see, only a handful of ports are actually open by default on a system installed using the Reduced Networking Software Group. The ports open in the above example belonged to the *rpcbind* process (ports TCP/111, UDP/111, and UDP/32772) and the *syslogd* process (UDP/514). If you did not want these services running, you can disable them with the following commands:

 # svcadm disable network/rpc/bind # svcadm disable system/system-log

Alternatively, you could have also configured *rpcbind* to use TCP Wrappers by running the following commands:

```
# svccfg
svc:> select network/rpc/bind
svc:/network/rpc/bind> setprop config/enable_tcpwrappers = true
svc:/network/rpc/bind> quit
# svcadm restart network/rpc/bind:default
```

Certainly, you would then need to configure your TCP Wrappers hosts.allow(4) and hosts.deny(4) files accordingly. For syslogd, you could also have set the LOG_FROM_REMOTE parameter in the /etc/default/syslogd file to *NO*. This would have caused the *syslogd* process to not listen for incoming connections from remote hosts.

But I digress...

Now, since only 150-Mbytes of software was installed, it should come as no shock to you that there is a lot of other software that was not installed. This is why the Reduced Networking Software Group is a *foundation* for minimization. You will need to add any software packages (either manually or by defining them in your JumpStart installation profile) that you need for applications, services, management or support.

For example, let's look for some common programs and services to see what happens:

```
# echo $PATH
/usr/sbin:/usr/bin
# which telnet
no telnet in /usr/sbin /usr/bin
# which ftp
no ftp in /usr/sbin /usr/bin
# which rcp
no rcp in /usr/sbin /usr/bin
# which rsh
no rsh in /usr/sbin /usr/bin
# which ssh
no ssh in /usr/sbin /usr/bin
# which mount
/usr/sbin/mount
# mount -F nfs -o ro 10.1.1.100:/export/disk1 /mnt
mount: Operation not applicable to FSType nfs
# truss
truss: not found
# snoop
snoop: not found
```

As you can see, the Reduced Networking Software Group does not come with very much! It is precisely this reason however why it will help customers wishing to build minimal configurations. By providing a solid, core set of packages, customers are free to take an additive approach to building minimal systems by simply adding in those packages that they want or need. This approach is much improved from the typical method employed today that requires users to remove unnecessary software packages - as this approach was prone to error and often raised problems for the supportability of such configurations.

Since I believe that many people will want to have Secure Shell in their default configuration, I did want to provide the JumpStart installation profile entries that would help. If you would like Secure Shell (but do not care about tunnelling X11 connections), then you can use the following profile:

```
install_type    initial_install
cluster         SUNWCrnet
cluster         SUNWCssh add
package         SUNWgss  add
partitioning    explicit
filesys         rootdisk.s1     768     swap
filesys         rootdisk.s0     free    /
system_type     standalone
```

Well, that's all for now. Check back soon for another installment of lesser known and/or publicized security enhancements to the Solaris 10 OS. I still have a bunch lined up for you! Let me know what you think of this series of articles as well as ideas for future updates. Take care!