New User To Technical Expert – Solaris Bookshelf

Stewart Watkiss

This document is currently
under construction

This version is to be
considered a preview only

Solaris

Networking Guide

Copyright

Stewart Watkiss

Contents

Preface to Bookshelf

This is the second UNIX book that I've written. The first is a book on AIX and incorporates the full range of New User to Technical Expert in a single book. The book started out as an aide-mémoire as part of my revision towards becoming an AIX Certified Advanced Technical Expert. I passed the exams and continued writing until it took the form of a complete book on the subject.

After completing my AIX Certification and Book I set myself a second challenge of applying my acquired UNIX skills to learn Solaris. This bookshelf is intended to pass that on to others that may be interested. Rather than writing this as a single book I decided for the bookshelf approach, as it is easier to manage as several smaller books.

This was written whilst I had hands on role implementing some new services on Solaris. This has given me practical experience of how the operating system can be used and in some areas how it really works as opposed to what the manuals say.

Introduction to this book

This is book 3 in the "New User to Technical Expert Series".

This book describes the interaction required with the SPARC hardware. It is primarily for when running the Solaris Operating System however some of the sections are equally applicable if the machine is to run Linux or another operating system.

Hardware is an area that can cause a lot of confusion for those new to SPARC machines as it is very different in operation to the x86 architecture that most are familiar with.

Most of the topics described are from the OpenBoot screen (also known as the OK prompt) where the discussion is specific to Solaris then it will say so in the text.

Chapter

1

Basic TCP/IP Networking

This chapter explains how TCP/IP
networks operate.

TCP/IP is an abbreviation for Transmission Control Protocol / Internet Protocol. It is a set of protocols that define how two or more computers can communicate with each other. This is a set of rules that describe how the data is passed between the computers. The protocol is an open description of how to write the software so that it can be developed for use on any type of computer. Within the TCP/IP networking protocol there are lots more protocols. These provide different functions as part of the networking. These can be integral to the operation of the networking, such as the Domain Name System or could be an application that uses the network such as E-mail (both of these are discussed in further detail later).

TCP/IP is not limited to a certain computer, it is an open protocol that independence from any particular operating system. A heterogeneous network can therefore be created consisting of any combination of UNIX, Windows, Apple or OS/2, OS390 (plus any other) computers. We will also be covering UDP (User Datagram Protocol). This often goes alongside TCP. The main difference is that TCP is connection based protocol whereas UDP is connectionless. In other words when TCP is being used there is a session setup between the hosts and the transfer is guaranteed. This compares with UDP where the data is sent but there is no checking that it has been received.

A common way of comparing these is to liken TCP to the telephone system and UDP to the Postal system. With the telephone when you establish a connection with the other person, you know for certain that the user receives the message. If you were disconnected during the telephone conversation then you would know about it and be able to phone the other person again. With the postal system after you post the letter then you do not know for certain whether or not the mail will be received. After you have posted the letter it could be lost or destroyed on it's way to it's destination. Or if the person has moved house they may never receive the letter.

At first it may sound that there is no reason to choose UDP over TCP after all if you can have the extra reassurance then why would you care about UDP. The reason for this is that there is a lot of overhead involved in TCP. For each data being sent a confirmation has to be generated and even if there is no data being sent there will normally be some kind of keep alive signal. Whereas for some less important data you may just want to send and forget it with the hope it will reach the other end. Some applications even add their own checking on top of the UDP protocol to overcome the problems associated with it being a connectionless protocol.

OSI Model

Networking protocols are often described relating to the OSI model. The OSI model splits the different functions of networking into different layers. By describing the networking protocols in layers it allows the layer to be changed without affecting other layers.

The networking models are particularly useful in that it allows the protocol to be

implemented on any system. Allowing UNIX computers to talk as a peer with PC's or even mainframes.

Still to add
- more information on the networking models / layers
- IP addressing information
- Brief DNS Information

Chapter

2

Network Configuration

This chapter explains the
configuration required to set-up
networking on a Solaris Computer.

Basic Network Files

There are a number of files involved in the initial network configuration. This includes the hostname, the IP addresses and the definitions of subnet masks. Solaris uses several different files that cross-reference each other. This is different from some systems where the details are held in a smaller number of files. It is important therefore to ensure that all the required files exist and are updated as appropriate.

Hosts

Each Interface address on the system has a corresponding hostname, which needs to be defined on the system. This is held in the /etc/inet/hosts file. In addition any hosts that are not resolved into IP addresses using DNS, or another naming service (see the chapter on naming), can have their entry added into this file.

The file consists of the IP address followed by one or more hostnames. The hostnames may be local names, or fully qualified names. Typically the first entry will also have the loghost alias for logging purposes.

```
#
# Internet host table
#
127.0.0.1        localhost
192.168.0.3   thishostname   loghost
10.12.35.122  interface2.domainname.com
```

Hostname

The hostname of the computer can be display by using the "hostname" command. This command can also be used to change the entry dynamically by following the command with the new hostname. If changed dynamically the value will revert back to its previous setting when the system is next rebooted. To change the value permanently then the file /etc/nodename should be edited. The file contains a single entry being the fully qualified hostname for the system.

Interfaces

Each Interface that is to be used has to have it's IP address associated with it. The file used is /etc/hostname.*int*. Where int is replaced by the interface address (hostname is always the same, it is not replaced with the hostname of the system). E.g. /etc/hostname.hme0, or /etc/hostname.eri1
The file has a single entry being either the hostname (which must have an entry in the /etc/inet/hosts file), or the IP address.

It can also include additional parameters that are used in the ifconfig file (see the troubleshooting section for more information).

```
192.168.0.3 netmask 255.255.255.0 broadcast 192.168.0.255 up
```

Network Masks

If the networks that the interfaces are connected to are subnetted then the
/etc/netmasks file can be used to describe the different subnets to provide correct
routing decisions. A entry should be added for any specific subnets. e.g.

```
192.168.0.0      255.255.255.0
```

Static Routes

The default route is normal specified in the file /etc/defaultrouter (although this is
sometimes not used for systems with multiple routes). The defaultrouter file has a
single entry for the IP address or hostname (if listed in /etc/hosts), or the default
router.

Additional routes can be added dynamically by using the route command. In the
following example a route will be added to the 10.0.1.0 subnet via 192.168.1.1.

```
route add -net 10.0.1.0 -netmask 255.255.255.0
192.168.1.1
```

Other routes can be added by creating a script within the startup. This could be in
/etc/rc2.d called S77routes. Typically this is created by adding a file to /etc/init.d and
creating a link to rc2.d.

This file would contain the route commands the same as if added from the command
line. The only difference being to include the full path to the route command
/usr/sbin/route.

Not Router

If there are multiple routes on the system then it could act as a router. Creating the file
/etc/notrouter will prevent the system for forwarding traffic between interfaces. This
is normally required unless the computer is to be a firewall or router.

Chapter

3

Naming

This chapter explains the name to address translation. Including using the hosts file, NIS and DNS. It includes information on DNS and on running a DNS server on Solaris.

Naming – to be created

Configuring a Primary DNS Nameserver

This section describes the steps required in making a Solaris machine act as a primary DNS nameserver. It does not go into details on the contents of the configuration files for which you should refer to "DNS and BIND"[1], or another appropriate guide.

Standard BIND 8 Configuration

The following describes the steps required to set the Solaris system as a primary DNS nameserver. This is using the standard install of Solaris BIND. This is not however the recommended option, as it is both an old version (BIND 8, whereas BIND 9 has many improvements), and is lacking certain tools (e.g. the most useful dig command). See later for details of installing BIND 9 from source (or it can be installed from pre-compiled binaries at www.sunfreeware.com).

Set resolv.conf so that it has local domain listed and is set to have a nameserver 0.0.0.0

```
domain domainname.com
nameserver 0.0.0.0
```

The file /etc/named.conf is then the file that causes Solaris to start the named service. For management purposes it is sometimes easier if this is stored in a different directory. In that case you may want to create the directory /usr/local/dns and store the named files in there, creating a link to /etc/named.conf.

The rest of the files should be in the base directory specified within the named.conf file, unless otherwise specified.

The named.ca file contains details of the top level domain name servers. This should be downloaded using anonymous ftp from rs.internet.net and is located in the domain directory.

Now create an entry in named.conf for any domains that this server is to be authoritative for:

```
zone "thisdomain.com" in {
        type    master;
        file    "primary/thisdomain.com";
};
```

---

[1] Albitz & Liu, DNS and BIND - 4th Edition, O'Reilly, Sebastopol 2001

Then create the file specified in the above entry

```
$TTL 86400
; Nameserver Data File for thisdomain.com.
;
; ADD COMMENTS AT BOTTOM
;
;
;----------------------------------------------------------------------
--
; Start of Authority record
;----------------------------------------------------------------------
----
@               IN SOA ns1.thisdomain.com. email.thisdomain.com.     (
                        2004042902  ; date of last update YYYYMMDD##
                              3600  ; refresh 2ndary every hour
                               900  ; 2ndary to retry every 15 minutes
                           2419200  ; 2ndary expiry in 4 weeks
                             86400  ; resource record time-to-live
                                                                    )
;
             IN     NS     ns1.thisdomain.com.
             IN     NS     ns2.thisdomain.com.
;
;-------------------------------------------------------------------
;
ns1          IN     A      192.168.1.1
ns2          IN     A      192.168.1.2

www             IN  A      192.168.1.3
```

A named.ca file is also required to specify the root level nameservers. This is installed
in the named directory. You can get the latest copy using FTP. Download the file
/domain/named.root from rs.internic.net. This file should be updated
regularly.

BIND 9 Installation and Configuration

An alternative to running the older BIND 8 is to install the later BIND 9 version. This
has a number of new security features and is now the standard for nameservers.

This describes installing from the source, but binary packages are also available from
www.sunfreeware.com . The following instructions assume that you already have a
compiler installed on your system (typically gcc).

First remove the following files:
/usr/sbin/in.named
/usr/sbin/named-bootconf
/usr/sbin/named-xfer

The source code can be freely downloaded from www.isc.org , normally the most
recent stable version should be used. Transfer the source file to the target system,
gunzip and untar the file. Then run the following:

```
./configure --prefix=/installdir
make
make install
```

The installdir could be /usr/local/dns, /var/named /opt/named or any other appropriate directory.

The binary files will be installed in the /installdir/bin directory and some configuration files will need to reside in other directories. The preferred way of resolving this is using symbolic links between the files. The following links should be created.

ln -s /installdir/sbin/named /usr/sbin/in.named
ln -s /installdir/sbin/rndc /usr/sbin/rndc
ln -s /installdir/bin/dig /usr/local/bin/dig

Set resolv.conf so that it has local domain listed and is set to have a nameserver 0.0.0.0

```
domain domainname.com
nameserver 0.0.0.0
```

The file /etc/named.conf is then the file that causes Solaris to start the named service. This should be installed in the installdir, but with links to /etc/named.conf and /installdir/etc/named.conf.

The rest of the files should be in the base directory specified within the named.conf file, unless otherwise specified.

A named.ca file is also required to specify the root level nameservers. This is installed in the named directory. You can get the latest copy using FTP. Download the file /domain/named.root from rs.internic.net. This file should be updated regularly.

Now create an entry in named.conf for any domains that this server is to be authoritative for:

```
zone "thisdomain.com" in {
        type    master;
        file    "primary/thisdomain.com";
};
```

Then create the file specified in the above entry

```
$TTL 86400
; Nameserver Data File for thisdomain.com.
;
; ADD COMMENTS AT BOTTOM
;
;
;---------------------------------------------------------------------
--
; Start of Authority record
;---------------------------------------------------------------------
----
@               IN SOA ns1.thisdomain.com. email.thisdomain.com.     (
                        2004042902  ; date of last update YYYYMMDD##
                              3600  ; refresh 2ndary every hour
                               900  ; 2ndary to retry every 15 minutes
                           2419200  ; 2ndary expiry in 4 weeks
                             86400  ; resource record time-to-live
                                                                     )
;
            IN     NS      ns1.thisdomain.com.
            IN     NS      ns2.thisdomain.com.
;
;-------------------------------------------------------------------
;
ns1         IN     A       192.168.1.1
ns2         IN     A       192.168.1.2

www            IN  A       192.168.1.3
```

Using BIND 9 the daemon is managed using the rndc command. This requires the
existence of the file /installdir/etc/rndc.conf which needs to be created.

```
# Start of rndc.conf
key "rndc-key" {
        algorithm hmac-md5;
        secret "randomsecretkey";
};

options {
        default-key "rndc-key";
        default-server 127.0.0.1;
        default-port 953;
};
# End of rndc.conf
```

Configuring a Secondary DNS Nameserver

Standard BIND 8 Configuration

The following describes the steps required to set the Solaris system as a secondary
DNS nameserver. This is using the standard install of Solaris BIND. This is not
however the recommended option, as it is both an old version (BIND 8, whereas
BIND 9 has many improvements), and is lacking certain tools (e.g. the most useful
dig command). See later for details of installing BIND 9 from source (or it can be
installed from pre-compiled binaries at www.sunfreeware.com).

Set resolv.conf so that it has local domain listed and is set to have a nameserver 0.0.0.0

```
domain domainname.com
nameserver 0.0.0.0
```

The file /etc/named.conf is then the file that causes Solaris to start the named service. For management purposes it is sometimes easier if this is stored in a different directory. In that case you may want to create the directory (e.g. /usr/local/dns, /var/named, or /opt/named are typical examples) and store the named files in there, creating a link to /etc/named.conf.

The rest of the files should be in the base directory specified within the named.conf file, unless otherwise specified.

A named.ca file is also required to specify the root level nameservers. This is installed in the named directory. You can get the latest copy using FTP. Download the file /domain/named.root from rs.internic.net. This file should be updated regularly.

Now create an entry in named.conf for any domains that this server is to be secondary for:

```
zone "thisdomain.com" in {
        type    slave;
        file    "secondary/thisdomain.com";
};
```

The secondary files referred to in named.conf do not need to be created. They will be created dynamically through the automated transfers. The directory does need to exist.

BIND 9 Installation and Configuration

An alternative to running the older BIND 8 is to install the later BIND 9 version. This has a number of new security features and is now the standard for nameservers.

This describes installing from the source, but binary packages are also available from www.sunfreeware.com . The following instructions assume that you already have a compiler installed on your system (typically gcc).

First remove the following files:
/usr/sbin/in.named
/usr/sbin/named-bootconf
/usr/sbin/named-xfer

The source code can be freely downloaded from www.isc.org , normally the most recent stable version should be used. Transfer the source file to the target system, gunzip and untar the file. Then run the following:

```
./configure --prefix=/installdir
make
make install
```

The installdir could be /usr/local/dns, /var/named /opt/named or any other appropriate directory.

The binary files will be installed in the /installdir/bin directory and some configuration files will need to reside in other directories. The preferred way of resolving this is using symbolic links between the files. The following links should be created.

ln -s /installdir/sbin/named /usr/sbin/in.named
ln -s /installdir/sbin/rndc /usr/sbin/rndc
ln -s /installdir/bin/dig /usr/local/bin/dig

Set resolv.conf so that it has local domain listed and is set to have a nameserver 0.0.0.0

```
domain domainname.com
nameserver 0.0.0.0
```

The file /etc/named.conf is then the file that causes Solaris to start the named service. This should be installed in the installdir, but with links to /etc/named.conf and /installdir/etc/named.conf.

The rest of the files should be in the base directory specified within the named.conf file, unless otherwise specified.

A named.ca file is also required to specify the root level nameservers. This is installed in the named directory. You can get the latest copy using FTP. Download the file /domain/named.root from rs.internic.net. This file should be updated regularly.

Now create an entry in named.conf for any domains that this server is to be secondary for:

```
zone "thisdomain.com" in {
        type    slave;
        file    "secondary/thisdomain.com";
};
```

The secondary files referred to in named.conf do not need to be created. They will be created dynamically through the automated transfers. The directory does need to exist.

Using BIND 9 the daemon is managed using the rndc command. This requires the existence of the file /installdir/etc/rndc.conf which needs to be created.

```
# Start of rndc.conf
key "rndc-key" {
        algorithm hmac-md5;
        secret "randomsecretkey";
};

options {
        default-key "rndc-key";
        default-server 127.0.0.1;
        default-port 953;
};
# End of rndc.conf
```

Chapter

4

Networking Services

This chapter describes some of
the networking services provided
by Solaris.

To be added information on rcommands / telnet / inetd / ssh etc.

Chapter

5

Network Troubleshooting

This chapter gives information on troubleshooting networking relating problems. It includes a description of the commands commonly used to check the state of different aspects of networking, such as netstat, ping and traceroute.

Too add PD information

Chapter

6

Troubleshooting

This chapter describes some common problems within Solaris and how to go about fixing them.

Glossary

Command Summary

Index