Global Information Assurance Certification Paper

Setting up a Secure DNS server with Bind 9.1.3 on HPUX 11.00

By: George E. Frankle
SANS Unix Security Practical Assignment v1.8
11-12/2001 to 01/2002

Overview

A small home office domain "fubar.com", had lost its secondary domain name server and was in need of a replacement. A HP 712/80 workstation and a copy of HPUX 11.00 was available. The network consisted of a 16 IP address subnet behind a DSL bridge. The network for the purposes of this exercise will be defined as XXX.YYY.ZZZ.112 to XXX.YYY.ZZZ.128 with a subnet mask of 255.255.255.240. The network is directly exposed to the Internet and contains two PC systems and a number of UNIX machines that serve as the primary DNS server and a group of web servers. The PC's are protected by Black Ice Defender, which is running in paranoid mode. The existing DNS server was running bind 4.9.3 with HPUX latest patches. The plan is to shut down the bridge temporarily and to set up and configure the new DNS server, test it, and place it on line. The existing DNS server would be rebuilt from scratch to duplicate the new server and apache web server would be set up on foo2 as well. This document will focus on the new DNS server. All other systems on the network are experimental (Linux and Sun workstations) and are usually not on line. This leaves us with a network consisting of two HPUX 11.00 boxes and the two PC's. The existing HPUX box runs an Apache web server as well as DNS and SSH. The box is secure and has been on line for about two years with no evidence of being hacked. Even so, the existing server will be rebuilt to the same standards as the new server. The strategy for the upgrade will be to use the existing HPUX server to collect the software and patches needed to install the new server. Once the new server is brought on line and tested the existing server will be taken off line and rebuilt and the apache web server restored. Once the systems are back on line, both systems will be subject to a high intensity SARA intrusion scan to test vulnerability. TARA will also be run to uncover any internal UNIX issues.

Table of Contents

Risk Assessment

The network containing the existing foo2.fubar.com, and the proposed new foo3.fubar.com, is connected, by a DSL bridge, directly to the Internet. With this in mind, every effort was made to only install and configure what was needed to accomplish the defined tasks.

Remote access to the systems is necessary to provide system administration and monitoring. As telnet, ftp, and the "r" functions are inherently insecure and easily compromised, openssh2 will be installed and configured to use only ssh2 protocol. Openssh2 will also be configured to permit only non-root access.

All inetd services will be deconfigured to prevent compromise.

Sendmail will not be run, as it is not necessary to send mail to or from the systems. All mail for this domain is handled by the ISP to minimize risks associated with sendmail.

A secure version of named (9.1.3) will be set up and configured on both dns servers. This named server will be run in a chrooted environment to minimize the impact of a possible break-in.

No nfs services will be permitted to or from the servers. NIS or NIS+ will also not be run.

All SNMP services will be disabled to prevent compromise.

NTP, advisable from a forensics standpoint, will be implemented, using two publicly available stratum 2 servers to provide ntp data.

The Netscape browser will be installed on both servers to permit rapid access to software and patch updates for the servers. Care will be taken to use a Netscape version (4.72) or above that will minimize the risk of compromise while using this software.

An Ignite image of each server will be made to permit restoral of the server to original condition in the event of compromise. This image is to be made before the network is brought on line to the Internet to insure its integrity. Additionally this image could be restored (at a later date) to a spare 4GB disk to provide a baseline comparison of the original systems, which would be useful to detect possible alterations to the system if a compromise is suspected.

A new HP intrusion detection product (IDS/9000, a free download) will be installed. This product will continuously monitor filesystem changes, user logins, and user activity to provide early warnings of possible compromise.

CDE will be installed but not used on the servers (many of the software bundles installed require the libraries installed with the package). The CDE will be disabled. Any “X” client services needed on the servers will be displayed on an Xserver (exceed) on one of the PC systems on the network. This system runs an ssh2 and sftp client and is set up to use "X” tunneling between the servers and the PC.

Hardware Definition

foo2.fubar.com: (existing system)

HP 715/100 HPUX workstation
PA-RISC Processor
2 2gb internal SCSI HDD.
1 4gb external SCSI HDD.
1 Internal CDROM drive
1 external DDS-2 Tape drive (moved between systems for backups)
128 MB Memory
Built in 10BaseT Ethernet interface.

foo3.fubar.com: (New DNS Server)

HP 712/80 HPUX Workstation
PA-RISC Processor
1 4 GB external SCSI HDD
1 External HP CDROM drive
1 external DDS-2 Tape drive (moved between systems for backups)
128 MB Memory
Built in 10BaseT Ethernet interface.

Required Software with Download Locations

Officially distributed HPUX software used in project:

Location: http://www.software.hp.com

QPK1100_11.00.depot (latest HP quality pack of patches for HPUX 11.00)
perl_11.00.depot (version 5.6.1 of PERL needed for Security patch check tool, and SARA, to work.)
B6834AA.depot (New HP security patch check tool. This shows missing security patches from the system)
upgrade_bind812.tar (latest official bind package from HP. Used for first upgrade step from 4.9.3)
ignite11_11.00.tar (HPUX software for creating reinstall image.)
Netscape Communicator version 4.7.9
J5083AA_11.00.depot (HPUX IDS/9000 product for intrusion detection)
PHKL_21360 (Patch required for J50583AA from HP Patch depot ftp site)
sdk_13102os11.depot (Java software needed for J50583AA)

Software from HPUX porting center (binaries, HPUX packages):

Location: http://hpux.cs.utah.edu

gcc-3.0.1-sd-11.00.depot.gz (gnu c compiler. Needed to compile bind and SARA)
binutils-2.11.2-sd-11.00.depot.gz (required for GCC to work/install)
bison-1.29d-sd-11.00.depot.gz (Needed to compile bind)
zlib-1.1.3-sd-11.00.depot.gz (required for Openssh2)
openssl-0.9.6-sd-11.00.depot.gz (required for Openssh2)
openssh-2.5.1p1-sd-11.00.depot.gz (this worked after install)
openssh-3.0p1-sd-11.00.depot.gz (latest but I had trouble making it work!)
lsof-4.51-sd-11.00.depot.gz (useful security analysis tool)

Software from the Internet Software Consortium:

Location: http://www.isc.org/products/BIND/bind9.html (now bind9.2.0)
(ftp://ftp.isc.org/isc/BIND/src for bind9.1.3)

bind-9.1.3.tar.gz (source code) Note that bind-9.2.0 has come out since this project was started. Bind 9.1.3 still available via ftp from same site. Note: I tried to compile bind 9.2.0 with no success in the same environment. This may be a 64-bit problem.

Center for Internet Security:

Location: http://www.cisecurity.org

sara-3.5.1.tar.gz(source code needs Perl5 to work)

Other:

HPUX 11.00 Install CD
TARA: http://www-arc.com/tara/index.shtml tara-2.0.9.tar.gz (system scanner generic HPUX only)

Installation of the new DNS server (foo3.fubar.com)

1) Using the Netscape browser on foo2.fubar.com download all the required additional software and collect in a directory (/archive2/upgrade) on foo2.fubar.com.

2) Once this has been done turn off the DSL bridge isolating the network from the external internet. This is important to prevent any potential break-in during the install process before foo3.fubar.com is hardened.

3) Turn on the power to the external CDROM and 4 GB disk drive connected to the new foo2 CPU.

4) Turn on the 712/80 (foo3) CPU and hit escape to escape to a boot shell.

5) Run "search scsi" to insure and identify that all necessary hardware is recognized prior to the install.

6) Make sure that the HPUX 11.00 install CD is loaded in the CDROM drive.

7) Run boot scsi.0.2 to boot the install CD.

8) Once booted select the option to install HPUX 11.00. Select the advanced install option. You will be given the choice of installing a minimum system or a full CDE system. I first tried to build the system using the minimal install without CDE to minimize any security issues with CDE. Unfortunately this option does not install many of the shared libraries (mostly X-Windows related libraries) which are needed for the other software to function. I ended up reinstalling using the full CDE environment. By choosing the advanced install you can select your file-system sizes. This is critical if you hope to be able to maintain the system in a secure state as regular security patching takes up a significant amount of space in /var and /usr. Many of the additional software packages take up a large amount of space in /opt. One additional item to be considered is to use a minimum of a 4 GB disk to install vg00. Having vg00 all be on one disk greatly simplifies the backup process using Ignite (make_recovery). A bdf output from the completed system gives some guidelines for filesystem sizes to choose:

```
foo3# bdf
Filesystem          kbytes    used   avail %used Mounted on
/dev/vg00/lvol3     143360   23542  112373   17% /
/dev/vg00/lvol1      83733   29175   46184   39% /stand
/dev/vg00/lvol9     716800  263058  427421   38% /var
/dev/vg00/lvol8     757760  500763  240990   68% /usr
```

/dev/vg00/lvol4 65536 1259 60263 2% /tmp
/dev/vg00/lvol7 602112 250442 329708 43% /opt
/dev/vg00/lvol6 20480 1118 18157 6% /home
/dev/vg00/lvol5 1536000 627103 853280 42% /archive

Note: I use the /archive directory as my software repository and development environment.

9) Prior to beginning the install process you can also select a root password and configure the network interface. The network parameters for this system (altered to protect the real system) follow:

IP Address: XXX.YYY.ZZZ.117
Netmask: 255.255.255.240
Default Gateway: XXX.YYY.ZZZ.113
Primary DNS Server: XXX.YYY.ZZZ.116

10) The install can now proceed (takes 2-3 hours on one of these old slow systems). The installation concludes with the system coming up on the network (this is why we isolate the system from the internet prior to hardening).

11) The next step is to move all of the required software from foo2.fubar.com to foo3.fubar.com via ftp. All of the software will be collected in the /archive directory on foo3.

12) In the archive directory gunzip all the gzipped files. All of the binary packages can be installed via the following commands:

For .depot files:

swinstall –s /archive/package-name.deopt

For binary .tar packages use:

swinstall –s /archive/package-name.tar

Many of the packages have dependencies and must be installed in a specific order. The order I installed the binary packages follows:

ignite11_11.00.tar (HPUX software for creating reinstall image.)
QPK1100_11.00.depot (latest HP quality pack of patches for HPUX 11.00) (includes security patches)
perl_11.00.depot (version 5.6.1 of PERL needed for Security patch check tool, and SARA, to work.)

B6834AA.depot (New HP security patch check tool. This shows missing security patches from system)
upgrade_bind812.tar (latest official bind package from HP. Used for first upgrade step from 4.9.3)
binutils-2.11.2-sd-11.00.depot (required for GCC to work/install)
gcc-3.0.1-sd-11.00.depot (gnu c compiler. Needed to compile bind and SARA)
bison-1.29d-sd-11.00.depot (Needed to compile bind)
zlib-1.1.3-sd-11.00.depot (required for Openssh2)
openssl-0.9.6-sd-11.00.depot (required for Openssh2)
openssh-2.5.1p1-sd-11.00.depot (this worked after install and key generation)
lsof-4.51-sd-11.00.depot.gz (useful security analysis tool)
Netscape 4.7.9 browser.
J5083AA_11.00.depot (HPUX IDS/9000 product for intrusion detection)
PHKL_21360 (Patch required for J50583AA from HP Patch depot ftp site)
sdk_13102os11.depot (Java software needed for J50583AA)

13) Once all the above software was installed it was necessary to get all the path stuff adjusted to make the programs accessible. Most of the software installed in /opt. A useful bit of scripting (contributed by one of my esteemed co-workers) that can be appended to .profile and .dtprofile(if CDE is used) and will accomplish most of this follows:

```
#Additional paths
#
PATH=\
$PATH:\
`echo /opt/*/bin | sed -e 's/ /:/g'`:\
/usr/ucb:\
/usr/openwin/bin:\
/usr/ccs/bin


MANPATH=\
`echo /opt/*/man | /usr/bin/sed -e 's/ /:/g'`:\
/usr/man:\
/usr/openwin/share/man
export MANPATH
#
```

14) **Setting up the new DNS server:** The first step in setting up the new bind server was to get the old bind.4.9.3 configuration files from foo2.fubar.com. These files: /etc/named.boot and /etc/named.db/* were ftp'd to foo3. The bind.8.1.2 binaries downloaded from the HP website came with a nice conversion filter to covert named.boot to the new style configuration file, named.conf. The tool is used as follows:

```
cd /etc
cat named.boot|/usr/bin/namedbootconf.pl >named.conf
```

I needed to change the location of the default perl binary location to /opt/perl/bin/perl to get the script to work. The created named.conf file contents follow:

```
// generated by named-bootconf.pl

options {
        check-names response fail;      // do not change this
        check-names slave warn;
        directory "/etc/named.db";
        query-source address * port 53; (note: I uncommented this. It was preceded by a # in
original output)
        version "not a chance!!";
        allow-recursion { list of machines on network subnet separated by ; };

};

zone "fubar.com" {
        type master;
        file "db.fubar";
        allow-transfer { none; };
};

zone "ZZZ.YYY.XXX.in-addr.arpa" {
        type master;
        file "db.XXX.YYY.ZZZ";
        allow-transfer { none; };
};

zone "0.0.127.in-addr.arpa" {
        type master;
        file "db.127.0.0";
        allow-transfer { none; };
};

zone "." {
        type hint;
        file "db.cache";
};
```

The items in red were added later to increase the security of the bind application. Reasoning follows:

version "not a chance!!"; (added to prevent bind version being obtained via dig)
allow-recursion { list of machines on network subnet separated by ; }; (added to prevent all systems external to the local lan from using these servers as their dns lookup servers. I.E. these servers will only return IP addresses for the local network.
allow-transfer { none; }; (to prevent anyone from doing a zone transfer)

15) The bind version 4.9.3 files db.fubar, db.127.0.0 and db.XXX.YYY.ZZZ have a different SOA record layout as of bind-8.2 (HP version 8.1.2 also uses this configuration). An example of the modified db.fubar file follows:

```
$TTL 3h (this statement also needed at start of each file.)
fubar.com.  IN SOA  foo2.fubar.com.  username.mailservername.com.  (
                    10      ; Serial
                    3h      ; Refresh after 3 hours
                    1h      ; Retry after 1 hour
                    1w      ; Expire after 1 week
                    1h )    ; negative caching TTL of 1 hour
```

Once these changes were made bind-8.1.2 was successfully started by running /usr/sbin/named. Bind functionality was now tested with the following:

nslookup foo2.fubar.com XXX.YYY.ZZZ.117.

This now worked successfully.

16) Further research at this point revealed that version 8.1.2 of bind, while considerably more secure than the default HPUX version 4.9.7, still had the potential for compromise. The choices at this point were to go to bind8.2.5 or to try and compile bind9.1.3. Bind.9.1.3 was chosen as it is compiled by default with static libraries, which made the conversion to running in a chrooted environment much easier. The first step in this process was to gunzip bind-9.1.3.tar.gz (source code) in the /archive directory. The resulting bind-9.1.3.tar file was extracted with tar xvf bind-9.1.3.tar. This created the directory /archive/bind-9.1.3 which contained all the necessary source code to build the package. From the /archive/bind-9.1.3 directory the build process consists of running ./configure and then make. Unfortunately this did not work. Even though I was sure to have the path to gcc first in the PATH environment the ./configure script kept picking /usr/bin/cc as the c compiler. Running make would fail using this compiler would fail as gcc or the ansi c compiler was required. The problem was finally solved by setting the variable CC=/opt/gcc/bin/gcc. The ./configure script then found gcc and the make ran successfully. Make install was run and the software was installed in the /usr/local subtree by default and this was not changed permitting rollback to bind.8.1.2 if necessary. The bind8.1.2 daemon /usr/sbin/named was then stopped and the /usr/local/sbin/named daemon for bind.9.1.3 was started. Examination of syslog.log and testing from foo2 revealed that the bind.9.1.3 was successful.

17) The next step was to set up bind to run in a chrooted environment. The first step here is to create the user named and the group named as follows:

echo " named:*:42:42:named:/: " >> /etc/passwd
echo "named::42:named " >> /etc/group

Be sure to Use ">>" not ">" or you will replace your passwd or group file with the output of the echo command. The directory /var/named was created. Within /var/named etc, dev, lib, user and var were created. All but var were owned by root:sys. var was owned by named:named and is used to collect logs. The directories named and run were created under var. Permissions on all directories under /var/named were set to 755. A copy of /etc/named.conf was made to /var/named/etc/named.conf. The directory /var/named/etc/named.db was created and the configuration files from /etc/named.db were copied to this directory.

18) Named is started in the chrooted environment with:

/usr/local/sbin/named -u named -t /var/named

The chrooted named application was tested from foo2 with named and syslog.log was checked and all worked as planned.

19) The stock /sbin/init.d/named file was backed up to /sbin/init.d/named.4.9.7 and /sbin/init.d/named was edited as to start the bind.9.1.3 application on system reboot. The file /etc/rc.config.d/namesvrs was edited as follows:

```
#       @(#)namesvrs: PHNE_16470
###############################################
# named (BIND) configuration. See named(1m). #
###############################################
#
#  Name server using the Domain Name System (DNS) protocol (RFC 1034/1035)
#
# NAMED:      Set to 1 to start nameserver daemon.
# NAMED_ARGS: Arguments to the nameserver daemon
#
#  Configuration of a named boot file (e.g. /etc/named.boot) is needed for
#  successful operation of the name server.
#
NAMED=1
NAMED_ARGS=""
```

The /sbin/init.d/named file used to start the chrooted bind.9.1.3 application follows:

```
#!/sbin/sh
#
# named $Revision: 1.2.214.2 $ $Date: 96/10/10 17:46:46 $
#
#          Copyright (C) 1993, 1994 Hewlett-Packard Company
#
# NOTE:    This script is not configurable!  Any changes made to this
#          script will be overwritten when you upgrade to the next
#          release of HP-UX.
#
#
# NOTE: The /etc/named.boot file is a host specific file that resides in a
#       private root directory.  In a diskless cluster, this file should be
#       created on the server node so that named is started only on the
#       server.  There are no restrictions against starting named on a client,
#       but this is not an optimal situation and is discouraged. A different
#       boot file path can be specified using the NAMED_ARGS variable in
#       /etc/rc.config (see named(1m) for argument syntax).
#

unset UNIX95
PRE_U95=true;export PRE_U95;

PATH=/sbin:/usr/sbin:/usr/bin:/usr/local/bin:/usr/local/sbin
export PATH

rval=0
set_return() {
        x=$?
        if [ $x -ne 0 ]; then
                echo "EXIT CODE: $x"
                rval=1  # always 1 so that 2 can be used for other reasons
        fi
}


case $1 in
start_msg)
        echo "Start name server daemon"
        ;;

stop_msg)
        echo "Stopping name server daemon"
        ;;
```

```
'start')
        if [ -f /etc/rc.config ] ; then
                . /etc/rc.config
                else
                echo "ERROR: /etc/rc.config defaults file MISSING"
        fi

        if [ "$NAMED" -eq 1 -a -x /usr/local/sbin/named ]; then
                /usr/local/sbin/named -u named -t /var/named $NAMED_ARGS && echo
"named \c"
                set_return
                if [ $rval -ne 0 ]; then
                        echo "Error in starting named. Recommend checking the"
                        echo "syslog file (usually /var/adm/syslog/syslog.log)"
                        echo "for possible reasons."
                fi
        else
                if [ ! -x /usr/local/sbin/named ]; then
                   echo "/usr/local/sbin/named is not executable"
                   rval=1
                else
                   rval=2
                fi
        fi
        ;;

'stop')
        if [ -f /etc/rc.config ] ; then
                . /etc/rc.config
                else
                echo "ERROR: /etc/rc.config defaults file MISSING"
        fi

        #
        # Determine PID of process(es) to stop
        #
        if [ "$NAMED" -ne 1 ]; then
           rval=2
        else
           if [ -r /var/named/var/run/named.pid ]; then
              if  kill `cat /var/named/var/run/named.pid` ; then
                        rm /var/named/var/run/named.pid
                        echo "named stopped"
                else
                        rval=1
                        echo "Unable to stop named"
```

```
            fi
        else
            rval=1
            echo "Unable to stop named (no pid file)"
        fi
    fi
    ;;

*)
    echo "usage: $0 {start|stop}"
    rval=1
    ;;
esac

exit $rval
```

Note: Items in red are changes made for bind.9.1.3.

The system was rebooted and the bind.9.1.3 application successfully started on boot.

20) **Setting up and configuring Openssh2:** The first step in getting the openssh2 application running is to, once the software is installed, generate the server keys. This is done by changing directories to /opt/openssh2/bin. There are only two users on this system, a root user and a non root user. As root, run ./ssh-keygen to generate the keys for root. Then after su – userid run ssh-keygen again to generate keyset for the user. The initial sets of keys for the sshd daemon (host-keys) are generated as follows:

while in /opt/openssh2/etc run,

```
/opt/openssh2/bin/ssh-keygen -t rsa1 -f /etc/ssh/ssh_host_key -N ""
/opt/openssh2/bin/ssh-keygen -t rsa -f /etc/ssh/ssh_host_rsa_key -N ""
/opt/openssh2/bin/ssh-keygen -t dsa -f /etc/ssh/ssh_host_dsa_key -N ""
```

The file /opt/openssh2/etc/sshd_config can be edited to prevent root from logging onto the system directly if desired (change PermitRootLogin yes to PermitRootLogin no). If you do not want to allow ssh1 connections (advisable) add Protocol 2 in lieu of the default Protocol 2,1. You should not have to change anything else to get ssh up and running.

Once the above has been completed the daemon can be started with /opt/openssh2/sbin/sshd. To get things to start automatically on bootup you can place the following script in /sbin/init.d/sshd and create a link /sbin/rc2.d/S995sshd -> /sbin/init.d/sshd:

```
#!/sbin/sh
#
# @(#) $Revision: 78.1 $
#
# NOTE:    This script is not configurable!  Any changes made to this
#          script will be overwritten when you upgrade to the next
#          release of HP-UX.
#
# WARNING: Changing this script in any way may lead to a system that
#          is unbootable.  Do not modify this script.

#
# Starts/stops the sshd daemon
#

# Allowed exit values:
#       0 = success; causes "OK" to show up in checklist.
#       1 = failure; causes "FAIL" to show up in checklist.
#       2 = skip; causes "N/A" to show up in the checklist.
#           Use this value if execution of this script is overridden
#           by the use of a control variable, or if this script is not
#           appropriate to execute for some other reason.
#       3 = reboot; causes the system to be rebooted after execution.

# Input and output:
#       stdin is redirected from /dev/null
#
#       stdout and stderr are redirected to the /etc/rc.log file
#       during checklist mode, or to the console in raw mode.

PATH=/usr/sbin:/usr/bin:/sbin:/opt/openssh2/sbin
export PATH

# NOTE: If your script executes in run state 0 or state 1, then /usr might
#       not be available.  Do not attempt to access commands or files in
#       /usr unless your script executes in run state 2 or greater.  Other
#       file systems typically not mounted until run state 2 include /var
#       and /opt.

rval=0

# Check the exit value of a command run by this script.  If non-zero, the
# exit code is echoed to the log file and the return value of this script
# is set to indicate failure.
```

```
set_return() {
	x=$?
	if [ $x -ne 0 ]; then
		echo "EXIT CODE: $x"
		rval=1  # script FAILed
	fi
}

# Kill the sshd process(es).
# $1=<search pattern for your process>

killproc() {
	pid=`ps -el | awk '( ($NF ~ /"'$1"'/) && ($4 != mypid) && ($5 != mypid)  ){ print
$4 }' mypid=$$ `
	if [ "X$pid" != "X" ]; then
		if kill "$pid"; then
			echo "$1 stopped"
		else
			rval=1
			echo "Unable to stop $1"
		fi
	fi
}


case $1 in
'start_msg')
	# Emit a _short_ message relating to running this script with
	# the "start" argument; this message appears as part of the checklist.
	echo "Starting the sshd daemon"
	;;

'stop_msg')
	# Emit a _short_ message relating to running this script with
	# the "stop" argument; this message appears as part of the checklist.
	echo "Stopping the sshd daemon"
	;;

'start')

	# source the system configuration variables
	if [ -f /etc/rc.config.d/sshd ] ; then
		. /etc/rc.config.d/sshd
	else
		echo "ERROR: /etc/rc.config.d/sshd defaults file MISSING"
	fi
```

```
        # Check to see if this script is allowed to run...
        if [ "$RUN_SSHD" -eq 1 -a -x /opt/openssh2/sbin/sshd ]; then
                /opt/openssh2/sbin/sshd
                set_return
        else
                rval=2
        fi
        ;;

'stop')
        # source the system configuration variables
        if [ -f /etc/rc.config.d/sshd ] ; then
                . /etc/rc.config.d/sshd
        else
                echo "ERROR: /etc/rc.config.d/sshd defaults file MISSING"
        fi
        if [ "$RUN_SSHD" -eq 1 ]; then
                killproc sshd
        else
                rval=2
        fi
        ;;

*)
        echo "usage: $0 {start|stop|start_msg|stop_msg}"
        rval=1
        ;;
esac

exit $rval
```

The following script called /etc/rc.config.d/sshd is also required:

```
# sshd daemon
#
# Set RUN_SSHD to 1 to start the sshd daemon at boot
RUN_SSHD=1
```

The system was tested from another system on the local lan with an ssh2 client and login worked with some error messages about an untrusted login system which disappeared after being overridden which adds the system to the clients database as trusted systems. The system was rebooted and again tested successfully indicating that sshd successfully started on bootup. Additional testing included trying to log in with an ssh1 client (this failed) and trying to log in as root (this also failed).

Openssh2, by default with this package, logs all attempted connections to /var/adm/syslog/syslog.log. Since the system has been online I have been getting about 1 –2 attempted connections per day indicating the presence of people probing the system for possible connections.

21) **Install the SARA software: (INSTALL document bundled with package follows with my additions in red). The install document listed below was contained within the SARA package sara-3.5.1.tar.gz. This package was provided by the Center for Internet Security website located at http://www.cisecurity.org/**

PREREQUISITES

The following are the minimum requirements for running SARA:

1. **UNIX/Linux Platform with perl 5.0.3 (or greater, we use 5.6.1), cc (or gcc)**
2. **A Web Browser (e.g., Netscape, Lynx, Mosaic)**
3. **Normal utilities, such a gunzip, make, etc.**

GETTING THE LATEST VERSION

The latest version can always be found by referring to http://www-arc.com/sara. We suggest that you join the sara-l list server while you are at it.

UNPACKING AND BUILDING

The SARA distribution is provided in a compressed tar format. The distribution file is in the format sara-x.x.x.tar.gz where x.x.x is the version number. Unpack the distribution with the following commands

```
gunzip sara-3.5.1.tar.gz
tar xvf sara-3.5.1.tar
```

The SARA build process assumes that the compiler is named 'cc'. If your environment uses gcc AND there is no link for cc, perform the following (assumes sh, bash, or ksh shell):

```
CC=/opt/gcc/bin/gcc
export CC
```

Next, build the program by typing make OSTYPE where OSTYPE may be:

aix osf bsd bsdi dgux irix4 irix freebsd
hpux9 hpux linux linux-nansi sunos4 sunos5

trusted-sunos5 sysv4

For instance, if you are building the program in a HPUX 11.00 environment, type:

make hpux

This command may protest if it cannot find all of the OS' system utilities. The programs nmblookup and smbclient are optional but SMB file/print sharing tests will not be possible. Other 'protests' indicate that certain SARA core tests will not be possible (e.g., showmount missing will limit SARA's ability to detect open exports).

SARA does not find the WWW browser that you want to use, edit the config/paths.pl file and change the line

$MOSAIC= "/opt/netscape/netscape";

22) **Install TARA:**

1) TARA is a group of shell scripts and does not have to be compiled on the system. The archive can simply be unzipped and untarred in a suitable directory. The software does not have to be installed further and can be run directly from the installed directory.

2) For the sake of completeness copies of the README and USING files from the documentation follow:

README:

Tiger Analytical Research Assistant (TARA) is an upgrade to the TAMU
'tiger' program. Since 'tiger' has not been updated since 1994,
there were numerous changes made to the 'systems' directories. Output
was streamlined to provide a more readable report file. Also, minor
bugs in the 'scripts' directory were corrected. TARA was tested
under Red Hat Version 5.2 (kernel 2.0.35), SGI IRIX 6.5, and
SunOS 5.7. In addition, an HTML option (tiger -H) is offered.
This upgrade was performed by the Advanced Research Corporation
under a contract from the National Institutes of Health.

'tiger' is a set of scripts that scan a Un*x system looking for security
problems, in the same fashion as Dan Farmer's COPS. 'tiger'
was originally developed to provide a check of UNIX systems on the A&M
campus that want to be accessed from off campus (clearance through the
packet filter). As such, we needed something that *anyone* could run

if they could figure out how to get it down to their machine.

If you just want to run it, without regards to time considerations,
then just 'cd' into the tiger directory and run './tiger' as 'root'.

---> You should check to see if you have the latest digital signatures
for the system(s) you are checking. I regularly place updated
signature files on anonymous FTP at

net.tamu.edu :/pub/security/TAMU/tiger-sigs/*

The util/installsigs script can be used to install the updated
signatures. As of Tiger 2.2.2, installsigs is also capable
of installing signatures for new OS releases (not new platforms
or major releases though).

NOTE

The 'tigerrc' file is set up for TAMU hosts, and disables/reduces some
of the checks. You should probably copy 'tigerrc-dist' to 'tigerrc'
and edit it to taste. It is set for a fuller check mode (TAMU hosts
might want to run with this config file as well). 'tigerrc-all' has
everything already maxed out and enabled (except for PATH_ALL).

(Or use the '-c' switch to use an alternate tigerrc file as of 2.2.2)

I recommend that you read the USING file for anything other than the
aforementioned situation.

****

See the file COPYING for legal stuff.

If you have any thing to say about 'tiger', please let us know. New
things to check, how to improve things, *anything*, send it in... if
you think someone else has already sent in a bug report, suggestion,
etc., send it in anyway... the more times someone hits me over the
head with something, the more likely it is to get fix/included...

********** NOTE NOTE NOTE NOTE NOTE NOTE **********

There is now a mailing list available for 'tiger'. To subscribe,

send mail to 'majordomo@net.tamu.edu'. Include in the body of the message:

subscribe tiger

or

subscribe tiger alternate_email_address

The mailing list is managed via Brent Chapman's 'majordomo' package.

Doug.

Doug.Schales@net.tamu.edu

USING:

TARA INTRODUCTION

The Tiger Analytical Research Assistant (TARA) is an upgrade to the original Tiger program. Enhancements, include:

- o Minor bug fixes
- o Upgrade to systems features for Linux, SunOS, IRIX, and default
- o HTML output (e.g., tiger -H) option

TIGER INTRODUCTION

Here's a quick "HOW TO" on using 'tiger'...

First: Make sure you are using a 'tigerrc' file to your liking. The tigerrc-TAMU file disables a lot of checks. The 'tigerrc-dist' file enables all of them. You should probably edit one to your tastes, though I do recommend the full check.

Second, for just a test run, it is *NOT* necessary to install 'tiger'. Just 'cd' into the top-level tiger directory and run './tiger'. This will create a security report after some time (times vary based on system size and extent of checking defined in 'tigerrc').

If you fix some things, and want to run just part of the system without having to wait for the entire thing, 'cd' to the 'scripts' directory and you can run any of the scripts there standalone. Just use './scriptname'. The output will go to stdout, so if you want to save it to a file, you'll need to redirect it.

If you want more information on a particular message generated by 'tiger'

(or any of the scripts), you can use the 'tigexp' (TIGer EXPlain) facility. You have three choices here.

First, if you just want more information on a specific message, just use './tigexp msgid', where 'msgid' is the text inside the [] associated with each message. For example, to obtain more information about:

--WARN-- [acc001w] Login ID backup is disabled, but still has a valid shell (/bin/sh).

one would use './tigexp acc001w'.

Second, if you want to insert the explanations in the report, you can either run 'tiger' (or the individual scripts) with the '-E' option, or if you have already run it, then use 'tigexp -F report-file'. This will write a copy of the security report to stdout, with explanations inserted.

The third option is to generate a separate explanation file from a report file. To do this, use 'tigexp -f report-file'. An explanation report will be generated with message identifiers with each explanation. This can be used when the report file has lots of repeated message ID's and inserting explanations will increase the size of the report to absurd proportions.

Running 'tiger' regularly.

First: It still isn't necessary to "install" 'tiger'. Installing it is only a convenience. If you do not install it, then it will be necessary to either invoke 'tiger' (or the individual scripts) with the '-B' option or 'cd' to the 'tiger' directory before running it. The '-B' option informs the scripts where the top level 'tiger' directory is.

You have two options when running 'tiger' regularly. The first is to simply run 'tiger' out of cron. Since on large systems, a full run can take hours, this is probably not desirable.

The more desirable is to use 'tigercron'. With 'tigercron', it is possible to run the individual scripts spread out over a time period (some can be run three times a day, others once a week or month). In addition, 'tigercron' will (on some systems) e-mail a "change" report to the specified person (in 'tigerrc'). The "change" report will only contain "new" information and will only be mailed when

there *is* new information.

Installing 'tiger'. If you do decide to install 'tiger', simply edit
the 'Makefile' and set the variables at the top. Then type 'make
install'. NOTE NOTE NOTE: It is important that the destination
directory (TIGERHOME) is *NOT* be the same as the source directory
(where 'tiger' was extracted). The Makefile currently doesn't have
the smarts (I'm not sure it has any) to handle such a situation and
will mangle TIGER horribly.
---

23) **Install and set up IDS/900 intrusion detection software:**

A package such as tripwire or AIDE should be placed on the system (I use Sysguard, a somewhat obsolete but still useful package that was released to the public domain by Bellcore in 1999. Note that I was unable to locate a software archive for this product so it may not be available publicly any more. I obtained my copy from my employer.) to establish and maintain a database of key file permissions and checksums. A copy of this file should be kept off line to permit comparisons in the event of a suspected break-in. A new baseline should be created every time the system is patched. Note that I was unable to get either Tripwire or AIDE to compile in an HPUX 11.00 environment. With the above in mind I located and installed IDS/9000 from HP. This product provides real time monitoring of user activity and file changes. On line documentation for the product can be found at:

http://www.docs.hp.com/hpux/onlinedocs/J5083-90007/J5083-90007.html

and

http://www.docs.hp.com/hpux/onlinedocs/J5083-90006/J5083-90006.html

The product is dependent on having the HPUX Java environment installed on the system as well as a number of patches. Installation goes as follows:

swinstall –s /archive/ sdk_13102os11.depot (the Java environment – no reboot)
swinstall –s /archive/J5083AA_11.00.depot (makes kernel changes and causes reboot)

After the two installs run the following to determine needed patches:

/opt/ids/bin/IDS_checkInstall

The output from this command revealed that an additional kernel patch was needed. The patch was downloaded from the HP ftp web site and installed as follows:

swinstall –s /archive/PHKL_21360.depot (causes reboot as kernel filesets added.)

Once the system rebooted some additional configuration was needed to get the software up and running. All the steps are included in the above documentation references. I chose to run the software (agent and monitor console) configured to run on the 127.0.0.1 interface to prevent any other listeners running on the Internet exposed Ethernet interface.

Note that the install creates a new account called ids. This account is blocked at creation. A passwd should be assigned to this account with the passwd ids command. The software is run from this ids account. Once the configuration items are complete (see above HP documentation) the agent daemon will start up automatically on reboot. The monitor console can be run as follows:

Foo3#su – ids
$ export DISPLAY=pc1.fubar.com:0 (PC system running exceed Xserver)
$ /opt/ids/bin/idsgui (two X sessions will start up on the PC system which will allow you to monitor and analyze system activity and possible intrusions.)

Note that the system has picked up new logins (me logging in via ssh2 from the PC) and file changes to the wtmp file on login. The system has also picked up changes to root's mail file (/var/mail/root) caused by root cron jobs generating rootmail on the system. As more experience is gained with the product the configuration can be fine tuned to monitor only critical security areas.

24) Now that we have nearly all of the necessary components in place we can begin to harden the system prior to placing on line to the Internet. The first and most important step is to copy /etc/inetd.conf to /etc/inetd.conf.original. We can then execute >/etc/inetd.conf and then inetd –c to turn off all services controlled by inetd.conf.
Note that installation of certain patches and products like Ignite can reverse this process. The inetd.conf file should be checked after every patch install and every software package install to insure that these services remain off (guess who this happened to).
Note also that currently I am using /etc/passwd to authenticate users with secure shell and secure ftp. I will be switching to an HPUX trusted system (has a shadow database in lieu of passwords in /etc/passwd for authentication) later in this process. I am also investigating using public key authentication but will not be able to implement this within the time frame mandated by this project.

25) The file /etc/securetty containing only the word console should be created and set to 400 permissions and root:root ownership. This will permit root login only on the console. With telnet disabled this may not be critical but is advisable as a backup.

26) All other unneeded services like sendmail, nfs, snmp, etc. should be shut down from the scripts in /sbin/init.d and the control variable values contained in the scripts located in /etc/rc.config.d should be set to zero to prevent starting of those processes during bootup. Once this has been done you can use netstat –a and ps –ef to identify any other listeners

that you may not want to run. These can be killed and the startup scripts in /etc/rcX.d moved from SXXXname to NOSXXXname to prevent startup on reboot. After reboot the process table and the output of netstat –a can be viewed to insure the elimination of unwanted services.

27) If it is necessary to send outbound mail from this server (in my case it is not) you can add the following line to the root crontab:

9 * * * * /usr/sbin/sendmail –q

This will process mail on this server once an hour without keeping the sendmail daemon up and vulnerable.

28) Under normal operating conditions where inbound ftp is used an /etc/ftpusers file should be set up containing root and all the system, non-user, accounts in /etc/passwd. This file should be owned by root:root and be 400 permissions. This should not be necessary on this system but is set up for the sake of completeness.

29) The system was converted to a trusted system at this time using SAM. This created a shadow database and replaced all passwords in /etc/passwd with a * which should make cracking passwords, if a break in occurs, unlikely. This is done by invoking SAM, then selecting the security and accounting option. (the system then asks you if you want to convert to a trusted system, answer yes. The system then converts to using a shadow database and adds * in place of all passwords in the /etc/passwd file.) You may at this time set such options as password aging. I change passwords every 30 days for the two accounts (root + nonroot user + ids user) so I did not enforce password aging. I set the option to require a password when booting to single user to prevent the system from being rebooted manually and then gaining root access by entering single user mode.

30) TARA was run at this point. I first used the default tigerrc and found no significant problems. I then copied tigerrc-dist to tigerrc and reran the scans. Additional permission information generated in this report was reviewed and modified where I believed necessary or safe.

31) At this point the system can be brought back onto the network by restarting the DSL bridge. DNS lookup can be tested remotely from a system external to the local network. This was done and all worked ok. Ssh was tested remotely and worked. I tried to telnet and ftp to the system remotely without success as planned. Additionally the 4.9.3 DNS server on foo2 was shut down and DNS lookups were tried remotely for the fubar.com servers. The new DNS server was able to correctly respond to all requests.

32) The new HP security-patch-check utility that was installed was then run as follows:

cd /opt/sec_mgmt/spc/bin
./ security_patch_check –r

The output of this program follows:

NOTE: Downloading from ftp://ftp.itrc.hp.com/export/patches/security_catalog.sync.

Use of uninitialized value in addition (+) at /opt/perl/lib/site_perl/5.6.1/Net/FTP/I.pm line 29.
NOTE: ftp://ftp.itrc.hp.com/export/patches/security_catalog.sync downloaded to ./security_catalog.sync successfully.

NOTE: Downloading from ftp://ftp.itrc.hp.com/export/patches/security_catalog.gz.

NOTE: ftp://ftp.itrc.hp.com/export/patches/security_catalog.gz downloaded to ./security_catalog.gz successfully.

WARNING: ./security_catalog is group or world writable.

NOTE: Recalled patch PHCO_14044 is present, but superseded by PHCO_22096 on the target system. If patch PHCO_22096 is ever removed, patch PHCO_14044 will become active. Read the recall notice to make the right decision for your situation. Patch recall notices can be seen using the security_patch_check -m option, through the Patch Database area of the ITRC, or from within ./security_catalog.

NOTE: Recalled patch PHCO_16795 is present, but superseded by PHCO_23705 on the target system. If patch PHCO_23705 is ever removed, patch PHCO_16795 will become active. Read the recall notice to make the right decision for your situation. Patch recall notices can be seen using the security_patch_check -m option, through the Patch Database area of the ITRC, or from within ./security_catalog.

NOTE: Recalled patch PHCO_17556 is present, but superseded by PHCO_23651 on the target system. If patch PHCO_23651 is ever removed, patch PHCO_17556 will become active. Read the recall notice to make the right decision for your situation. Patch recall notices can be seen using the security_patch_check -m option, through the Patch Database area of the ITRC, or from within ./security_catalog.

NOTE: Recalled patch PHKL_19800 is present, but superseded by PHKL_24027 on the target system. If patch PHKL_24027 is ever removed, patch PHKL_19800 will become active. Read the recall notice to make the right decision for your situation. Patch recall notices can be seen using the security_patch_check -m option, through the Patch Database area of the ITRC, or from within ./security_catalog.

NOTE: Recalled patch PHKL_20171 is present, but superseded by PHKL_22926 on the target system. If patch PHKL_22926 is ever removed, patch PHKL_20171 will become active. Read the recall notice to make the right decision for your situation. Patch recall notices can be seen using the security_patch_check -m option, through the Patch Database area of the ITRC, or from within ./security_catalog.

NOTE: Recalled patch PHKL_20333 is present, but superseded by PHKL_23127 on the target system. If patch PHKL_23127 is ever removed, patch PHKL_20333 will become active. Read the recall notice to make the right decision for your situation. Patch recall notices can be seen using the security_patch_check -m option, through the Patch Database area of the ITRC, or from within ./security_catalog.

WARNING: Recalled patch PHKL_23127 is active on the target system. Its record, including the Warn field, is available from ./security_catalog, through the Patch Database area of the ITRC or by using the -m flag (security_patch_check -m ...).

WARNING: Recalled patch PHKL_23955 is active on the target system. Its record, including the Warn field, is available from ./security_catalog, through the Patch Database area of the ITRC or by using the -m flag (security_patch_check -m ...).

WARNING: The installed PHNE_18546 was intended to address the issues described in security bulletins(s) 097 . This patch has been recalled and is not generally recommended by Hewlett-Packard. The security bulletin and recall notice information should be reviewed. Each customer should respond in a manner appropriate to their environment. Security bulletins can be found by number at

http://itrc.hp.com/cki/bin/doc.pl/screen=ckiSecurityBulletin

Patch recall notices can be seen using the security_patch_check -m option, through the Patch Database area of the ITRC, or from within ./security_catalog.

*** BEGINNING OF SECURITY PATCH CHECK REPORT ***
Report generated by: /opt/sec_mgmt/spc/bin/security_patch_check.pl, run as root
Analyzed localhost (HP-UX 11.00) from gef3
Security catalog: ./security_catalog
Security catalog created on: Fri Nov 23 19:45:25 2001
Time of analysis: Sat Nov 24 10:19:04 2001

List of recommended patches for most secure system:

#	Recommended	Bull(s)	Spec?	Reboot?	PDep?	Description
1	PHCO_25110	176	Yes	No	No	lpspool subsystem cumulative
2	PHNE_23274	144	Yes	No	No	Bind 4.9.7 components
3	PHNE_24034	169	Yes	Yes	Yes	ONC/NFS General Release/Performance
4	PHSS_23670	088	Yes	No	No	OV EMANATE14.2 Agent Consolidated
5	PHSS_24608	145	Yes	No	No	AudioSubsystem July 2001 Periodic
6	PHSS_25138	168	Yes	No	Yes	CDE Runtime SEP2001 Periodic

*** END OF REPORT ***
NOTE: Security bulletins can be found ordered by number at
http://itrc.hp.com/cki/bin/doc.pl/screen=ckiSecurityBulletin

All patches with the exceptions of the ones indicated in RED were downloaded from the HP website and installed. It was decided to not remove any of the superceded patches that were recalled. These superceded patches will work their way out of the rollback database as subsequent patching cycles take place. Cleanup can be run after every patch upgrade with the –1 option to only keep only one previously superceded patch. After reading through the patch documentation obtained with check-security-patches –m it was decided that for this system the additional recalled patched not superceded need not be removed. Your mileage may vary here. My experience has been that removing patches (many of which have complex dependencies) unless they are causing problems on the system is risky and can often destabilize the system. The final decision should be based on reading and understanding the output from check_security_patches –m.

33) Setting up NTP:

The link at http://www.eecis.udel.edu/~mills/ntp/servers.htm can be used to locate two stratum 2 NTP servers in your area of the world/country. Contact should be made with the system administrators for the ntp servers to secure permission to obtain ntp data from the servers.

The following lines should be added to /etc/ntp.conf: (note that this is for my area yours may be different.)

server Armada.KJSL.COM version 3 prefer
server lovenun.mainecoon.com version 3

The date command can then be used to set the system date/time as close as possible to real time. (I used another system from work that was running ntp to determine time.)
The file /etc/rc.config.d/netdaemons should be edited to turn on xntpd at startup. An excerpt for the xntp section follows:

```
#######################################
# xntp configuration.  See xntpd(1m) #
#######################################
#
#  Time synchronization daemon
#
# NTPDATE_SERVER: name of trusted timeserver to synchronize with at boot
# (default is rootserver for diskess clients)
# XNTPD:        Set to 1 to start xntpd (0 to not run xntpd)
# XNTPD_ARGS:  command line arguments for xntpd
#
# Also, see the /etc/ntp.conf and /etc/ntp.keys file for additional
# configuration.
#
export NTPDATE_SERVER=
```

XNTPD=1
export XNTPD_ARGS=

The xntpd daemon can then be started with /sbin/init.d/xntpd start.
Ps –ef|grep ntp can be used to verify that the daemon is running. The output of this command follows:

```
root  1348     1  0  Jan  5  ?         3:29 /usr/sbin/xntpd
root  3727  2295  6 03:05:00 pts/1     0:00 grep ntp
```

The xntpd daemon sends diagnostic output to syslog.log. Some sample output follows:

```
Jan  8 20:31:34 gef3 xntpd[1348]: time reset (step) 0.155517 s
Jan  8 20:31:34 gef3 xntpd[1348]: synchronisation lost
Jan  8 20:36:38 gef3 xntpd[1348]: synchronized to 206.55.228.149, stratum=2
Jan  8 20:36:38 gef3 xntpd[1348]: time reset (step) -0.185628 s
Jan  8 20:41:09 gef3 xntpd[1348]: synchronized to 63.192.96.3, stratum=2
Jan  8 20:36:38 gef3 xntpd[1348]: synchronisation lost
Jan  8 20:41:57 gef3 xntpd[1348]: synchronized to 206.55.228.149, stratum=2
```

Since installing xntpd the date/time information on foo2 and foo3 has compared to each other and to my reference system at work to the nearest second.

34) At this point a backup of the system using the Ignite make recovery should be run as follows:

make_recovery -A -d /dev/rmt/(no-rewind tape device name)

The tape should be labeled with the date and system name and date and be kept permanently to restore the system if necessary after compromise or hardware failure. New tapes should be run weekly to provide backups and also should be run in the event of any major system changes such as patches.

Now that we had a functional and secure DNS server for the domain fubar.com the original DNS server system, foo2, could be brought down and upgraded to the same level as the new system, foo3.

Reinstalling the existing HPUX server (foo2.fubar.com)

1) The filesystem layout on foo2 is different than foo3, which has a single 4 GB external disk. The foo2 system has two internal 2 GB disks that contain the vg00 filesystems and the operating system. The external 4 GB disk on this system contains the archive filesystem /archive2. The internal two disks have been rearranged over time to accommodate the expanding needs of the system. Unfortunately this had been done without a lot of detailed planning. This became evident when a make recovery tape was attempted on foo2 and it would not run. Tarballs of the apache web server were made and these were copied to tape and to foo3.

2) The plan going forward is to use the make_recovery tape from foo3 to reinstall HPUX on foo2 using the external 4 GB disk drive and then reinstalling and re-hardening apache on the new system and recovering the web site information. The two internal drives could then be combined into a vg01 and used to make a new /archive filesystem.

3) When using the foo3 make recovery tape to install the new foo2 system it is necessary to shut down the Bridge and also shut down foo3. Once the new system is on line the system name and IP address can be changed to the foo2 name and XXX.YYY.ZZZ.116. At this time foo2 can be brought back on line and the Bridge reenabled.

Test and Validation Plan

Testing to insure telnet, ftp, rlogin are disabled:

1) Using a valid user account on the system attempt to connect to the system with telnet, ftp and rlogin. This should be refused. This was the case.

2) Note that I also cat out /etc/inetd.conf. This file should be empty. It is necessary to check this after each patch install or after adding additional software to the system. Some software products will create a new inetd.conf (note that Ignite is an example of this!)

3) Another work around here is to not use the inetd daemon at all. This daemon can be disabled via the startup scripts.

Testing the security and functionality of BIND:

1) Checking for version: From a remote location that has the dig program installed run: (I used foo2 and foo3 to check each other.)

```
foo3# /usr/local/bin/dig txt chaos version.bind foo2.fubar.com
foo2# /usr/local/bin/dig txt chaos version.bind foo3.fubar.com
```

output: (from first command above.)

```
; <<>> DiG 9.1.3 <<>> txt chaos version.bind
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18222
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;version.bind.                  CH      TXT

;; ANSWER SECTION:
version.bind.           0       CH      TXT     "not a chance!!"

;; Query time: 12 msec
;; SERVER: XXX.YYY.ZZZ.116#53(XXX.YYY.ZZZ.116)
;; WHEN: Tue Jan  8 09:08:14 2002
;; MSG SIZE  rcvd: 57
```

This validates that the version of bind is not revealed.

2) Testing to see if zone transfers are permitted:

foo3# /usr/local/bin/dig @foo2.fubar.com fubar.com axfr

; <<>> DiG 9.1.3 <<>> @foo2.fubar.com fubar.com axfr
;; global options: printcmd
; Transfer failed.

foo3# /usr/local/bin/dig @gef3.gefrankle.com gefrankle.com axfr

; <<>> DiG 9.1.3 <<>> @gef3.gefrankle.com gefrankle.com axfr
;; global options: printcmd
; Transfer failed.

This shows that zone transfers have been successfully blocked on both foo2 and foo3.

Testing Openssh2: (done for both foo2 and foo3)

1) As a regular user try to log in to the system from a remote ssh1 client. This was tried and failed.
2) Using an ssh2 client try to log into the system as root. This was tried and failed.
3) Using an ssh2 client log in to the system as a non root account. This was successful.

Testing the system with SARA:

1) Turn off the DNS Bridge before beginning testing to prevent any possible scans of systems outside of the local network.

2) On foo2 or foo3 run: export DISPLAY=pc1.fubar.com:0 (PC where exceed is running)

Operating instructions from Package:

RUNNING SARA

SARA supports three different operational modes:

1. Local interactive execution:

   The mode is initiated by merely typing './sara'. It will summons your Web browser on the SARA hosted machine (under normal installations, this would be Netscape Communicator). The operational functions can be initiated by clicking on the blue ball to the left of the command. Refer to the 'Documentation' function for details on how to operate SARA in this mode.

If you are using an older version of Netscape (e.g., 4.5.x or less), you may encounter problems when initiating many of the functions. Specifically, SARA may present a Open dialog box rather than the expected Web page. For proper operation, Netscape must be configured to interpret SARA screens (often ending in .pl extensions) as html documents. The Documentation (FAQ) provides the steps to properly configure Netscape for proper SARA operation.

The installed copy worked without any modification.

2. Command line execution:

SARA can be directed to perform scans through the UNIX/Linux command line. Refer to the Documentation or the SARA manual page (sara.8) for all of the command line options.

3. Remote interactive execution:

SARA can be controlled from a remote UNIX/Linux workstation. This should be configured by an experienced UNIX/Linux administrator. Details on this mode are available on the initial SARA page when you initiate SARA in interactive mode.

PROBLEMS

If you experience problems, please subscribe (http://www-arc.com/sara) to the sara-l list server and post your problem. We continually monitor this site.

Security Scan report form SARA using heavy attack scan:

Host: foo2.fubar.com

General host information:

Host type: unknown type
Subnet XXX.YYY.ZZZ
SSH server (GREEN) * Green means no significant vulnerabilities.
WWW server (GREEN)

Host: foo3.fubar.com

General host information:

Host type: unknown type
Subnet XXX.YYY.ZZZ
SSH server (GREEN)

Notes on testing:

1) I always turn off the DSL bridge to keep anything from leaking out to my internet neighbors during the scans.

2) The SARA software need Netscape installed to work. When running the SARA tool on a system without an Xserver present (CDE not running) you must export your DISPLAY to a system with an Xserver (my PC with Exceed and ssh2 client to log into the system with SARA)

3) I ran the scans at the most intense standard level scanning first foo3 and then foo2. The scan reports above indicate no vulnerabilities detected.

Ongoing Maintenance Plan

1) Passwords for the root and non-root users are changed on or about the first day of each month. The passwords used are never the same for the two accounts. Passwords are different on foo2 and foo3. Passwords are unique to these systems and are not used on other systems at work or with my Internet provider or on any of my Internet or web based logins.

2) The syslog.log and sulog files is examined daily to check for evidence of any unusual activity.

3) Every three months the quality pack bundle is downloaded from HP and installed on the system.

4) The security_patch_check tool is run weekly and any new security patches that are relevant are added to the system.

5) A new make recovery tape is created any time the system is patched or if any major configuration changes are made to the system. The original make_recovery tape is kept to permit quickly restoring the system in the event of a hardware failure or discovered intrusion. The security_patch_check tool can be run on the restored system and any new security patches could be added after restoring the system. Similarly the latest quarterly quality pack could be installed on the system. The bridge to the DSL, Internet connection, would be shut down during any restoration process to prevent unwanted intrusions before the system is secured.

6) Sysguard reports are run daily, weekly and monthly and are analyzed after the run. The checksums and permissions database is fine-tuned to eliminate files that change regularly. A new database is created and a copy stored off line whenever patching or other major upgrades are done. Recommended corrections made in the weekly and monthly report are carefully considered and made when deemed appropriate.

7) SARA intrusion scans are done monthly and compared to previous results to detect any possible new vulnerabilities. I normally shut down the DSL connection when running scans to eliminate the possibility of causing problems with any of my Internet neighbors.

8) TARA is run weekly and the output is compared to previous reports to discover changes to the system.

9) I keep open an ssh2 connection to my PC system from foo2 and foo3 and frequently monitor network activity with netstat –a. Any unusual connections are investigated further with lsof.

10) NTP is set up on foo2 and foo3 to continuously keep Date/Time information current. Date/Time is compared on both systems using the date command. Syslog output is examined daily to detect if there are any ntp issues.

11) All non-OS packages installed should be reviewed on a monthly basis to see if newer versions are available that fix security issues. If available the packages should be upgraded as necessary.

12) Advisories from CERT and the SANS Institute are be subscribed to and are reviewed on a regular basis for vulnerabilities that might be relevant to the servers on this network.

13) The IDS/9000 intrusion detection software is online at all times on both foo2 and foo3. The "X" based output from the manager consoles is reviewed regularly from the PC system. Note that I use a separate manager console on each system, which picks up the agent information on 127.0.0.1. While this requires two manager consoles to be run on the PC it eliminates inter system communication between foo2 and foo3 and keeping any other listeners off of the Internet lan.

References

1) [Daniel Barrett & Richard E. Silverman 2001] SSH The Secure Shell, O'Reilly and Associates, Inc. 02-2001.

2) [Paul Abitz & Cricket Liu, 2001] DNS and BIND, O'Reilly and Associates, Inc. 04-2001

3) [The Center For Internet Security] HP-UX Benchmark v1.01 Draft 2 2001.

4) Documentation, Install Files, and README files for the various software packages installed. A listing of all the software packages and where they were obtained can be found in the software section of this document.

5) Web Site for "The Center for Internet Security" http://www.cisecurity.org/
Documentation and package for SARA under SANS top 20.

6) HP Documentation website at http://docs.hp.com/ for documentation on the many HPUX products used in this project. HPUX 11.00, Security Patch scanner, and the IDS9000 Intrusion Detection System.

7) [Lee Brotzman and Hal Pomeranz] Running UNIX applications securely. Security for BIND and NTP

8) [Steve Acheson, John Green and Hal Pomeranz] Topics in UNIX Security. Setup and security for ssh.