Installing the Trusted Information Systems Internet Firewall Toolkit

1

The TIS Firewall Toolkit

- A set of components for building firewalls
- Does not enforce or mandate any particular policy
- Does not preclude using other firewall or security software in addition to the toolkit (e.g., COPS, SOCKS, Tripwire)

2

The TIS Firewall Toolkit *(cont)*

- Provides a minimum functionality for services that can be implemented with high security
  - FTP
  - Telnet / rlogin
  - X
  - SMTP e-mail
  - WWW

3

The TIS Firewall Toolkit *(cont)*

- Assumes that ***something else*** has managed blocking traffic or providing packet-level access control between networks

4

Blocking Traffic Between Networks

- Site policies may determine how traffic is blocked
- What ***is*** important is that traffic ***is blocked*** appropriately
- What ***is not*** important is ***how*** it's blocked

5

Blocking Traffic *(cont)*

- May be a mixture of:
  - Screening via a router
  - Complete traffic blocking via disabling routing and forwarding
  - Other means
- Two preferred approaches:
  - Dual homed gateway
  - Screening router

6

Blocking Traffic *(cont)*

- Dual homed gateways provide a high degree of assurance that traffic is blocked
- Screening routers are more flexible since you have the option of letting selected traffic through

7

Blocking Traffic *(cont)*

- Choose the approach that best suits your goals
- Choose the approach you know best how to implement in a secure manner

8

Installing the Toolkit

1: Shut everything off
2: Verify that it is indeed turned off
3: Enable each service one at a time
4: Test that each service is installed correctly
5: Document what you did
   – ***Especially*** anything site-specific

9

Helpful Hint: Tripwire

- ***Now*** is a ***good time*** to make a tripwire database of your system!
  – Self-documents all changes made during firewall install
  – You'll want to install tripwire on the firewall anyhow so why not do that part first?

10

Follow Conventions

- When you modify a system configuration file save the original for future reference
- Use the same notation throughout the system when saving an original file

```
cp /etc/mumble  /etc/mumble.orig
vi /etc/mumble
```

11

Reboot Often While Testing

- Be careful not to make changes and forget to test them
- Prevent embarrassing failure to reboot on power-up
- Avoid frequent system restarts
- Firewalls need to be stable hosts

12

Reboots

- Firewalls should not reboot often
- Reboots may be a security concern
- Send the administrator mail on reboot!

```
( echo “system rebooted”; echo; echo; dmesg ) | \
   /usr/ucb/mail -s “firewall reboot” root
```

13

Configure TCP/IP

- Configure the system network address
- Configure routing based on site policy
  - Generally routing is a simple “default” route to the “outside” and a set of routes to “inside” networks

14

Adaptive Routing

- Adaptive routing
  - Trust only routes from routers belonging to your organization or your internet service provider
  - Prevent internal bozos from causing problems
  - *Gated* is good for tightly controlling routing

15

Gated

- Various levels of stability and complexity
- ***Very*** nice feature to restrict where to take routes from

```
rip yes {
        broadcast;
        interface 128.175.38.1 noripin;
};
```

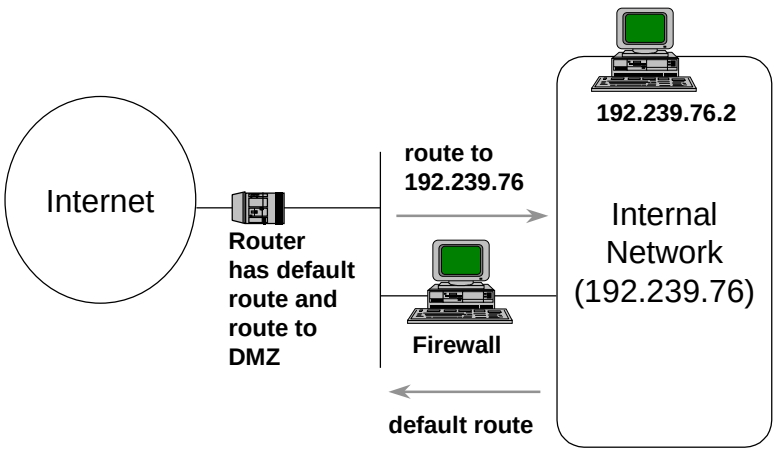
16

Static Routing

- Static Routing
  - Consider using static routing
  - Static routed machines less likely to be affected by user error or misconfigured routers on internal or external networks
  - Firewall routing should not change often

17

Typical Firewall Routing



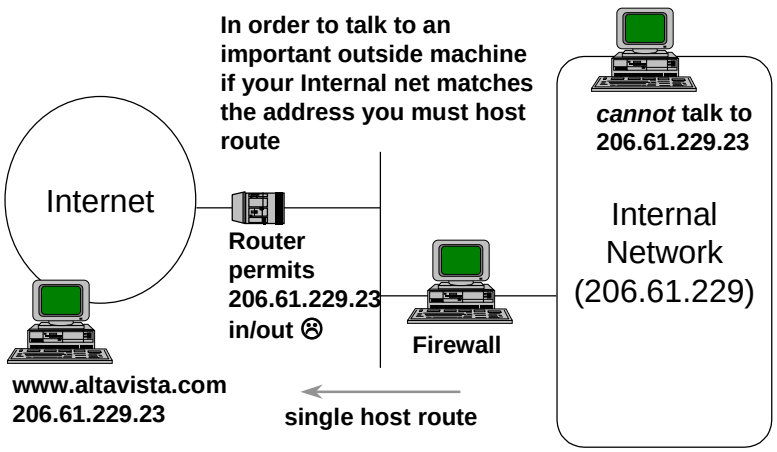
18

Nasty Routing

- If inside network address is non-assigned must use all static routes w/Internet
- If specific hosts on outside are worth talking to may need host routes
  - Can make router screening ***extremely*** ugly!

19

Nasty Routing *(cont)*



20

Kernel Configuration

- Unless your firewall will need kernel-based services such as NFS, rebuild a kernel that does not include them (if possible — ***this is vendor dependent***)
- If your kernel has network management hooks or built in SNMP, and you're not using them, disable them

21

Ipforwarding

- If you are using a dual-homed gateway and your O/S permits compile-time setting of *_ipforwarding*, disable it as you rebuild the kernel
- SunOs: (**/sys/netinet/in_proto.c**)

```
ip_forwarding == -1 never forward; never change this value.

int     ip_forwarding = -1;
```

22

Ipforwarding *(cont)*

- Solaris: (**/etc/rc2.d/S69inet**)

```
# Machine is a host: if router discovery finds a router then
# we rely on router discovery. If there are not routers
# advertising themselves through router discovery
# run routed in space-saving mode.
# Turn off ip_forwarding
ndd -set /dev/ip ip_forwarding 0
```

23

Ipforwarding *(cont)*

- On just about any UNIX ipforwarding can be disabled using the debugger to change the value of *_ip_forwarding*

```
# adb -w /vmunix /dev/kmem
_ip_forwarding? W 0
^D
# reboot
```

24

Server Processes

- Network service processes are generally started either at boot time or from service listeners such as *inetd*

Step 1: turn off boot time servers in */etc/rc.**

Step 2: turn off excess servers from */etc/inetd.conf*

25

Server Processes *(cont)*

Step 3: check the process table using the *ps* command

Step 4: check network sockets using the *netstat* command

Step 5: if there are still un-accounted-for processes running, return to **Step 1**

26

Boot Time Servers

- Servers started in */etc/rc**
  - NFS client mounts (comment them out)
  - Accounting (optional)
  - Inetd (*leave this one running*)
  - Lpd (comment it out)
- Consider starting with a bare slate

27

Boot Time Servers *(cont)*

- Servers started in */etc/rc.local*
  - Biod (comment them out)
  - Nfsd (comment them out)
  - Mountd (comment it out)
  - Syslogd (*leave this one running*)
  - Portmapper (comment it out)
  - Sendmail (comment it out for now, we'll fix it later)

28

Boot Time Servers (cont)

- If a server process is running find out what it does
  - Check the manual
  - Determine if it is necessary to the operation of the system
- If shutting it off doesn't hurt, it wasn't necessary

29

A Bare Process Table

- Examining the process table should show just about nothing running on the bastion host (for now)

```
# ps -ax
  PID TT STAT  TIME COMMAND
    0 ?  D     2:46 swapper
    1 ?  S     0:29 /sbin/init -
    2 ?  D     0:12 pagedaemon
   63 ?  S    18:19 syslogd
   77 ?  I    89:07 update
   80 ?  IW    3:55 cron
   82 ?  S     2:24 inetd
 2014 co IW    0:00 -sh
   87 b  IW    0:00 - std.19200 ttyb (getty)
 2228 co R     0:00 ps -ax
#
```

30

Netstat Output

```
% netstat -a
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address      Foreign Address     (state)
tcp        0      0  *.chargen          *.*                 LISTEN
tcp        0      0  *.daytime          *.*                 LISTEN
tcp        0      0  *.discard          *.*                 LISTEN
tcp        0      0  *.echo             *.*                 LISTEN
tcp        0      0  *.time             *.*                 LISTEN
tcp        0      0  *.smtp             *.*                 LISTEN
tcp        0      0  *.finger           *.*                 LISTEN
tcp        0      0  *.telnet           *.*                 LISTEN
tcp        0      0  *.ftp              *.*                 LISTEN
udp        0      0  *.syslog           *.*
udp        0      0  *.chargen          *.*
udp        0      0  *.daytime          *.*
udp        0      0  *.discard          *.*
udp        0      0  *.echo             *.*
udp        0      0  *.time             *.*
Active UNIX domain sockets
Address Type   Recv-Q Send-Q Vnode  Conn     Refs  Nextref Addr
f70930c dgram       0   0            0        0    0
f70808c dgram       0   0    f0cbc4  0        0    0     /dev/log
```

31

Restricting Root Login

- Root login should be restricted to console
- Administrators accessing system over network should log in with their login then “su” to root
- Consider using STEL or SSH for encrypted over the network firewall admin login

32

Restricting Root Login *(cont)*

- */etc/ttytab* lists what terminals root can log in on with a “secure” flag

```
#
# @(#)ttytab 1.7 92/06/23 SMI
# name  getty                           type            status  comments
console "/usr/etc/getty cons8"          sun             on local secure
ttya    "/usr/etc/getty std.9600"       unknown         off local secure
ttyb    "/usr/etc/getty std.9600"       unknown         off local secure
tty00   "/usr/etc/getty std.9600"       unknown         off local secure
```

- Delete all “secure” entries except “console”

33

Unpacking the Toolkit Sources

- Toolkit files are available for FTP from ftp.tis.com, in *pub/firewalls/toolkit*
- Toolkit sources come in two files
  - *fwtk.tar.Z* - Toolkit source code
  - *fwtk-doc-only.tar.Z* - Documentation (PostScript format)

34

Configure the Sources

- Master configuration file is *firewall.h*
  - Isolates system dependencies
  - Sets some default values
- *firewall.h* is already tailored for most BSD systems
- Some System V versions of UNIX require changes to Makefiles
  - See “fixmake”

35

Configure the Sources *(cont)*

- Authentication options are configured in *auth.h*
  - Must be edited in conjunction with authentication library Makefile *auth/Makefile* to reflect supported forms of authentication
  - Default authentication mode is plaintext passwords — plaintext passwords are not adequate for network use

36

Configure the Sources *(cont)*

- May want to change:
  - Log level (LLEV)
  - Log type (LOG_DAEMON)
- May need to change:
  - Declaration of malloc
  - Locking type
  - Struct direct/Struct dirent

37

Compile the Toolkit

- Default installation for toolkit is in */usr/local/etc*
  - Ensure that */usr/local/etc* directory exists
- Type "*make*" in the *fwtk* directory
- Once toolkit has built cleanly install the binaries by typing "*make install*"

38

Toolkit Does not Build

- If the toolkit fails to build check:
  - If programs do not link because of missing libraries check Makefiles and configuration options
  - Ensure that your system standard libraries are complete
  - Some systems require additional library names in Makefiles (e.g., for DBM database library, *-lndbm*)

39

Toolkit is Built, Now What?

- Configure proxies to run
- Configure authentication
- Test

40

Implementing Policy

- Each component of toolkit separately tailorable
- Generally all components should enforce consistent policy
- Suppose your inside network is 111.111 (class B)
  - Firewall proxies and services permit free access from within network 111.111.*

41

Implementing Policy *(cont)*

- Do not use DNS names to define security in toolkit configuration as DNS is spoofable
- Nodes not in DNS are filterable with special name “unknown”
- Firewall proxies may permit access from any other node (“*”) if authenticated

42

Implementing Policy *(cont)*

- SMTP Email is typically accepted from anyplace
- Typically each firewall acts as its own authentication server

43

Netperm-table

- Netperm-table is the firewall toolkit runtime configuration file
- Rules in netperm-table are read top-to-bottom and then left-to right
- First matching rule is applied
- Left hand side of each rule is a list of service names followed by a colon (a “*****” matches all services)

44

Netperm-table *(cont)*

- Right hand side of rule is a list of options

```
tn-gw:            welcome-msg /usr/local/etc/tn-gw.welcome
smap, smapd:      directory   /usr/spool/smap
*:                authserver  127.0.0.1 7777
```

45

Netperm-table *(cont)*

- Each toolkit application reads netperm-table at startup and “remembers” rules that apply to it
- Complete lists of the rules each application will use are listed in the manual page for the application

46

Netperm-table *(cont)*

- Rules match on the first word of the right hand side:

```
tn-gw:          welcome-msg /usr/local/etc/tn-gw.welcome
```

- Remaining words are extra parameters
- Netperm-table is not spell-checked
  - Misspelled rules will silently fail to match
  - Parameter mismatches or missing parameters are logged and proxy exits

47

Logging Errors

- Configuration errors and ***all*** diagnostics are logged via the *syslog* utility
- If something is not behaving properly the ***first*** place to check is syslog
- Positive events are also logged to syslog
  - Proxy startup and shutdown
  - Statistics

48

Logging Errors *(cont)*

- Administrators may wish to install shell scripts to summarize event logs or to flag warnings
  - Toolkit includes log reducers in "tools/admin/reporting"
- While installing the toolkit have all log messages directed to the console

```
# tail -f /usr/adm/messages &
```

49

Proxies

- Toolkit proxies (V1.3) assumed to be invoked from *inetd*
- For high volume sites *inetd* may overload and shut down service
  - More than 40 times/second causes "server failing, looping" message in syslog
  - Inetd stops listening on port
  - Newer version of toolkit will fix

50

Proxies *(cont)*

- Each proxy:
  - Starts and reads netperm-table
  - Checks client IP address and permissions
  - Logs the connection
  - Performs transactions on user's behalf
  - Logs the transactions
  - Logs transaction summary
  - Exits

51

/etc/inetd.conf (for a toolkit firewall)

```
# Time service is used for clock syncronization by folks to lazy to use NTP
time            stream  tcp nowait  root    internal
time            dgram   udp wait    root    internal
# Echo, discard, daytime, and chargen are used primarily for testing.
echo            stream  tcp nowait  root    internal
echo            dgram   udp wait    root    internal
discard         stream  tcp nowait  root    internal
discard         dgram   udp wait    root    internal
daytime         stream  tcp nowait  root    internal
daytime         dgram   udp wait    root    internal
chargen         stream  tcp nowait  root    internal
chargen         dgram   udp wait    root    internal
# Wrappers
ftp             stream  tcp nowait  root    /usr/local/etc/netacl         in.ftpd
telnet          stream  tcp nowait  root    /usr/local/etc/netacl         in.telnetd
login           stream  tcp nowait  root    /usr/local/etc/netacl         in.rlogind
finger          stream  tcp nowait  nobody  /usr/local/etc/netacl         in.fingerd
smtp            stream  tcp nowait  root    /usr/local/etc/smap           smap
nntp            stream  tcp nowait  root    /usr/local/etc/plug-gw        plug-gw nntp
# Authentication Server
auth            stream  tcp nowait  root    /usr/local/etc/authsrv        authsrv
```

52

/etc/inetd.conf (alternate form)

```
# Time service is used for clock syncronization by folks to lazy to use NTP
time            stream  tcp nowait  root    internal
time            dgram   udp wait    root    internal
# Echo, discard, daytime, and chargen are used primarily for testing.
echo            stream  tcp nowait  root    internal
echo            dgram   udp wait    root    internal
discard         stream  tcp nowait  root    internal
discard         dgram   udp wait    root    internal
daytime         stream  tcp nowait  root    internal
daytime         dgram   udp wait    root    internal
chargen         stream  tcp nowait  root    internal
chargen         dgram   udp wait    root    internal
# Wrappers
ftp             stream  tcp nowait  root    /usr/local/etc/ftp-gw         ftp-gw
telnet          stream  tcp nowait  root    /usr/local/etc/tn-gw          tn-gw
login           stream  tcp nowait  root    /usr/local/etc/rlogin-gw      rlogin-gw
finger          stream  tcp nowait  nobody  /usr/local/etc/netacl         in.fingerd
smtp            stream  tcp nowait  root    /usr/local/etc/smap           smap
nntp            stream  tcp nowait  root    /usr/local/etc/plug-gw        plug-gw nntp
# Authentication Server
auth            stream  tcp nowait  root    /usr/local/etc/authsrv        authsrv
```

53

Netacl: a TCP wrapper

- Front end “TCP wrapper” to control access to TCP-based services
  - General-purpose
  - Does not support UDP-based services
- You can use *tcp_wrappers* instead
- Process started by *inetd* replaces real server process

54

Netacl *(cont)*

- Checks to see if source of connection is permitted service
  - If source is permitted service the real service process is invoked
  - If source is denied service netacl exits and terminates connection
- Includes ability to chroot or set user-id of server process prior to invoking it

55

Netacl *(cont)*

- Service name is appended to "*netacl*" to generate netperm-table entry name
  - In.telnetd changes to netacl-in.telnetd
- "*-chroot directoryname*" option chroots service to specified directory
- "*-userid username*" option sets userid of service

56

Netacl *(cont)*

- "*-exec [options]*" must be last entry and specified server to invoke and its arguments
- Sample netacl rules

```
netacl-in.ftpd: permit-hosts 111.111.* -exec /usr/etc/in.ftpd
netacl-in.ftpd: permit-hosts unknown -exec /bin/cat /usr/local/etc/noftp.txt
netacl-in.ftpd: permit-hosts * -chroot /home/ftp -exec /bin/ftpd -f -l
netacl-in.fingerd: permit-hosts 111.111.* -exec /usr/etc/in.fingerd
netacl-in.fingerd: permit-hosts * -exec /bin/cat /usr/local/etc/finger.txt
```

57

TN-Gw: Telnet Proxy

- The first "permit-hosts" says the inside network is OK to use the proxy and that users may change passwords from there as it is a trusted network
- The second "permit-hosts" says any other host is OK to use the proxy if they authenticate first

58

TN-Gw *(cont)*

- Other rules define the command prompt and a welcome message

```
tn-gw:            prompt 'telnet proxy> '
tn-gw:            authserver 127.0.0.1 7777
tn-gw:            welcome-msg /usr/local/etc/tn-gw.welcome
tn-gw:            permit-hosts 111.111.* -passok -xok
tn-gw:            permit-hosts * -auth
```

59

Rlogin-Gw: Rlogin Proxy

- The first “permit-hosts” says the inside network is OK to use the proxy and that users may change passwords from there as it is a trusted network
- The second “permit-hosts” says any other host is OK to use the proxy if they authenticate first

60

Rlogin-Gw *(cont)*

- Other rules define the command prompt and a welcome message

```
rlogin-gw:          prompt 'telnet> '
rlogin-gw:          authserver 127.0.0.1 7777
rlogin-gw:          welcome-msg /usr/local/etc/tn-gw.welcome
rlogin-gw:          permit-hosts 111.111.* -passok -xok
rlogin-gw:          permit-hosts * -auth
```

61

FTP-Gw: FTP Proxy

- The first “permit-hosts” says the inside network is OK to use the proxy and that users may change passwords from there as it is a trusted network
- The second “permit-hosts” says any other host is OK to use the proxy if they authenticate first

62

FTP-Gw *(cont)*

- The "*-log { stor retr }*" option on the last rule causes file transfer filenames with "outside" to be logged

```
ftp-gw:           authserver 127.0.0.1 7777
ftp-gw:           welcome-msg /usr/local/etc/ftp-gw.welcome
ftp-gw:           permit-hosts 111.111.*
ftp-gw:           permit-hosts * -authall -log { stor retr }
```

- Ftp-gw mostly obsoleted by Web

63

X-gw: X11 Proxy

- Requires a user to manually "OK" connections to X server through firewall
  - Creates virtual X server on firewall
  - User sets DISPLAY=firewall:somenumber
  - Starts X application
  - Answers "OK" or not "OK" to proxy popup
  - Can shut down proxy anytime

64

X-gw *(cont)*

- X proxy started from Telnet or Rlogin proxy
- Telnet or Rlogin proxies need **-xok** flag

```
# X-forwarder rules
tn-gw, rlogin-gw:       xforwarder /usr/local/etc/x-gw
```

65

Http-gw: Web proxy

- Includes proxy capability for WWW, Gopher, and FTP
- No caching capability
- For sites with really large user bases consider a caching web server instead
  - If using a caching web server be attentive to server security

66

Http-gw *(cont)*

- Http-gw accepts defaults for http servers or gopher servers
- No (real) authentication supported
- May selectively filter URLs or sites

```
http-gw:          permit-hosts 111.111.*
```

67

Smap - SMTP queuer

- Goal is to prevent outsiders from communicating directly with large privileged processes like sendmail
- Mail is gathered to disk under chroot as unprivileged user
- Daemon process (*smapd*) sweeps spool directory and hands mail off to sendmail for delivery

68

Smap *(cont)*

- Smapd may perform additional scanning of message before handing off to sendmail
  - Current version performs some minimal checks for addresses in the envelope that contain pipe commands
  - More elaborate scanning may be desirable

69

Smap *(cont)*

- Smap is not a panacea
  - Original intent was to evolve into a direct mailbox delivery program
  - Eliminate *sendmail* altogether
- MMDF is probably a better mailer than smap+sendmail
  - Consider using MMDF or very carefully configured sendmail

70

Smap *(cont)*

- Smap requires a spool directory in which to queue mail
- Should be owned by same userid as smap and smapd execute under

```
mkdir /usr/spool/smap
chown uucp /usr/spool/smap
```

71

Smap *(cont)*

- Netperm-table rules include directory name

```
• smap, smapd:    userid uucp
• smap, smapd:    directory /usr/spool/smap
• smap:           maxrecip 4000
• smap:           maxbytes 1048576
• smap:           timeout 3600
```

72

Smapd

- Smapd starts at system initialization
- Changes user-id and enters wait loop
- Periodically wakes up and hands mail off to sendmail
- Add to /etc/rc.local

```
if [ -f /usr/local/etc/smapd ]; then
     /usr/local/etc/smapd; echo -n “smapd”
fi
```

73

Smapd *(cont)*

- Sendmail needs to be configured normally
  - Optionally sendmail executable may be made setuid *uucp*
  - Change ownership of */etc/aliases** */usr/spool/mqueue* to *uucp*
- Sendmail no longer runs as a demon

74

Smapd *(cont)*

- Since sendmail no longer runs as a demon mail queue must periodically be drained
- Add an entry to crontab or have a sendmail demon to manage queue in rc.local:

```
/usr/lib/sendmail -q20m
```

75

Testing Smap

- Telnet to SMTP port and send a message
- Received message should appear in smap queue area
- Check file modes and ownership of queued message
  - Should not be root-owned

76

Testing Smap *(cont)*

- Verify that the message was properly passed to sendmail for final delivery
- Sendmail can be configured and tested without smapd or smap running
- If running non-setuid-root sendmail it may complain in logs

77

Authsrv

- Authentication server
- Can be compiled with support for multiple forms of authentication
- Configure authsrv authentication support by editing *auth.h* and the *auth/Makefile*

78

Authsrv *(cont)*

- To add support for S/Key must have S/Key distribution and S/Key options in Makefile
- To add support for Digital Pathways must have a compatible DES library and SNK options in Makefile
- New options may be added without harming existing database

79

Authsrv Initialization

- If authsrv is invoked by root user at command line it enters administrator mode
  - Can create/delete users
  - Can list database
  - Can create privileged users
  - Can change passwords
- All operations are logged

80

Authsrv Initialization *(cont)*

- Authsrv entries in netperm-table define location of database and other options

```
authsrv:                permit-hosts 127.0.0.1
authsrv:                permit-hosts 111.111.111.111
authsrv:                database /usr/local/etc/auth.db
authsrv:                nobogus true
authsrv:                badsleep 900
```

81

Authorizing a User

- To initialize database just add users

```
# authsrv
-administrator mode-
authsrv# ls
authsrv# adduser admin "Auth DB admin"
ok - user added initially disabled
authsrv# ena admin
enabled
authsrv# proto admin pass
changed
authsrv# pass admin "plugh"
Password changed.
authsrv# superwiz admin
set wizard
authsrv# ls
Report for users in database
user       group      longname        ok?  proto      last
----       -----      --------        ---  -----      ----
admin                 Auth DB admin   ena  passw      never
authsrv# ^D
```

82

Authorizing users *(cont)*

- There is no need of an administrator if users are only added by “root”
- Group administrators can create or delete users within their group
  - To create a group administrator create a user and set the group wizard bit
  - Group administrators can list the members of their group only

83

Authorizing users (cont)

- Users can be bulk-loaded with “*authload*”
  - Authload reads authentication server dump records and replaces them in the database
  - Can be used to exchange authentication databases between firewalls
  - Can be used to restore damaged databases

84

Backing up Authsrv

- Authdump can be used to generate a backup auth database
- Install a cron job to copy the database out to a text file nightly for safe keeping:

```
5 4 * * * /usr/local/etc/authdump > /var/auth.db.bak
```

- Authdump output files contain keys — keep them secure

85

Netperm-table

```
# if the next 2 lines are uncommented, people can get a login prompt
# on the firewall machine through the telnet proxy
netacl-telnetd: permit-hosts 127.0.0.1 -exec /usr/libexec/telnetd
netacl-telnetd: permit-hosts YOURADDRESS 198.6.73.2 -exec
   /usr/libexec/telnetd
#
# if the next line is uncommented, the telnet proxy is available
netacl-telnetd: permit-hosts * -exec /usr/local/etc/tn-gw
#
# if the next 2 lines are uncommented, people can get a login prompt
# on the firewall machine through the rlogin proxy
netacl-rlogind: permit-hosts 127.0.0.1 -exec /usr/libexec/rlogind -a
netacl-rlogind: permit-hosts YOURADDRESS 198.6.73.2 -exec
   /usr/libexec/rlogind
# if the next line is uncommented, the rlogin proxy is available
netacl-rlogind: permit-hosts * -exec /usr/local/etc/rlogin-gw
```

86

Netperm-table *(cont)*

```
#
# to enable finger service uncomment these 2 lines
netacl-fingerd: permit-hosts YOURNET.* -exec /usr/libexec/fingerd
netacl-fingerd: permit-hosts * -exec /bin/cat /usr/local/etc/finger.txt

# Example smap rules:
# -------------------
smap, smapd:    userid 6
smap, smapd:    directory /var/spool/smap
smapd:          sendmail /usr/sbin/sendmail
smap:           timeout 3600
```

87

Netperm-table *(cont)*

```
# Example ftp gateway rules:
# --------------------------
#ftp-gw:        denial-msg      /usr/local/etc/ftp-deny.txt
#ftp-gw:        welcome-msg     /usr/local/etc/ftp-welcome.txt
#ftp-gw:        help-msg        /usr/local/etc/ftp-help.txt
ftp-gw:         timeout 3600
# uncomment the following line if you want internal users to be
# able to do FTP with the internet
ftp-gw:                 permit-hosts YOURNET.*
# uncomment the following line if you want external users to be
# able to do FTP with the internal network using authentication
ftp-gw:                 permit-hosts * -authall -log { retr stor }

# Example http-gw rules:
# ----------------------
http-gw:                permit-hosts YOURNET.*
```

88

Netperm-table *(cont)*

```
#tn-gw:         denial-msg      /usr/local/etc/tn-deny.txt
#tn-gw:         welcome-msg     /usr/local/etc/tn-welcome.txt
#tn-gw:         help-msg        /usr/local/etc/tn-help.txt
tn-gw:          timeout 3600
tn-gw:          permit-hosts YOURNET.* -passok -xok
# if this line is uncommented incoming traffic is permitted WITH
# authentication required
#tn-gw:         permit-hosts * -auth

#rlogin-gw:     denial-msg      /usr/local/etc/rlogin-deny.txt
#rlogin-gw:     welcome-msg     /usr/local/etc/rlogin-welcome.txt
#rlogin-gw:     help-msg        /usr/local/etc/rlogin-help.txt
rlogin-gw:      timeout 3600
rlogin-gw:      permit-hosts YOURNET.* -passok -xok
#rlogin-gw:     permit-hosts * -auth -xok
```

89

Netperm-table *(cont)*

```
# Example auth server and client rules
# -----------------------------------
authsrv:        hosts 127.0.0.1
authsrv:        database /usr/local/etc/fw-authdb
authsrv:        badsleep 1200
authsrv:        nobogus true

# clients using the auth server
*:              authserver 127.0.0.1 7777

# X-forwarder rules
tn-gw, rlogin-gw:       xforwarder /usr/local/etc/x-gw
```

90

Reporting

- Toolkit includes some summarizer scripts in **tools/admin/reporting**
  - ftp-summ.sh
  - http-summ.sh
  - tn-gw-summ.sh
  - weekly-report.sh
- Useful to run from crontab to mail admins

91

Testing Procedures

- Check what network services are available
- Check (from outside) that inside systems are not reachable
- Test authentication system
  - Attempt an operation that requires authentication

92

Testing Procedures *(cont)*

- Test each component by practical means
  - Telnet out through telnet proxy
  - Rlogin through rlogin proxy
  - FTP through FTP proxy
  - Send mail

93

Maintaining the Software

- Firewall toolkit software intended to be “base line” and not change other than bug fixes
- “If it works, don’t fix it”
- *fwall-users@tis.com* mailing list for notification of major bugs and releases
- *fwall-support@tis.com* for limited support for users

94

Installing Patches

- Format of netperm-table will not change (preserve backward compatibility)
- Format of auth database should not need to change
- If a new release is installed it should install “on top of” existing release without problems if the new release is configured the same way

95

Summary

- Toolkit is a useful set of components for building firewalls or secured systems
- Some assembly required
- Requires O/S specific configuration

96