Global Information Assurance Certification Paper

Security Audit of "*Chicago Hinge and Hasp*[1]"

Executive Summary

The company's network shows a high degree of attention to security issues in most areas as would be expected based on its line of business. There are, however, several areas that require improvement. In particular, password management, especially ageing and reuse, and backup/recovery practices need to be improved.

Especially noteworthy is the use of "defense in depth". Individual "core" machines are configured with security in mind, within the limitations of the particular operating system(s) in use on each machine. Applications with known exploitable "features" are configured with the features disabled. Network services that are required of internal operations are configured to restrict access to authorized machines or users. With the exception of identd (auth), no services are offered to the Internet. Remote logging, network time synchronization, and packet filtering firewalls are used. UPS units are used for all core systems and "shutdown" scripts are enabled. Machines used for testing and education, non-core machines, start with "out of the box" installation with "hardening" coming later in the process.

The system administrator has run nmap scans against all network devices (including a 3Com ISDN LanModem and a HP Print Server). The scans are run from "inside" and from the Internet. Additional vulnerable scans using the current version of SAINT have been run internally. The r* services have been deleted from the core systems and OpenSSH in is use. The administrator is subscribed to security alerting services for the Linux and the UNIX systems. They regularly monitor other news groups and security related websites. On the UNIX and unix-like systems, log and system configuration monitoring tools are applied. Security updates are applied quickly, as appropriate.

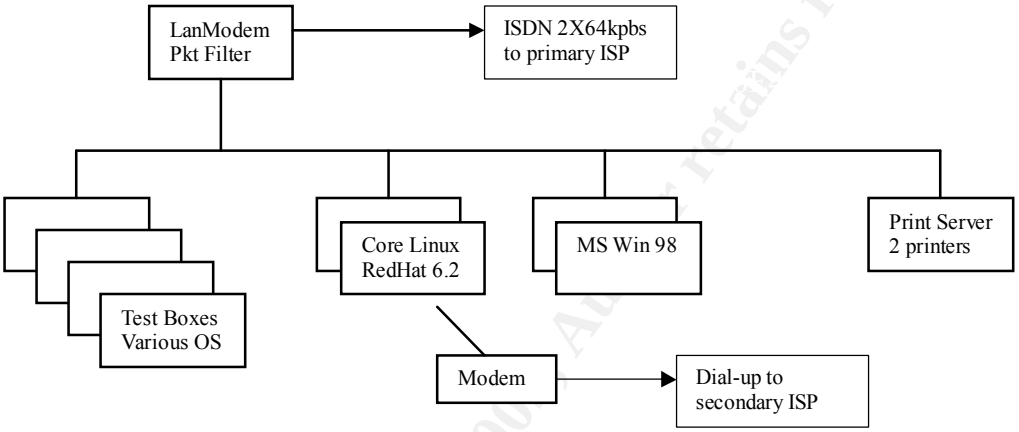
All of these good security practices, notwithstanding, password discipline is lax. While good password selection techniques are in use and public key technology is applied when using ssh, some passwords have not been changed for years and are used across multiple systems. Note that, because of physical access restrictions to the consoles and since no external access is allowed (except for dial-in FAX), the exposure has been considered acceptable. Critical data is backed up to separate drives, floppies, or 4mm tape, but only on a casual basis. The establishment of a DMZ to further isolate the test machines is proposed.

[1] *Chicago Hinge and Hasp* is a fictitious business operation that selectively accepts special consulting assignments in a wide variety of information technology areas. It is a class C corporation, but functions as a small office/home office (SOHO) enterprise. It operates an internal networking and operating systems laboratory with an average of eight computers and four to five different operating systems.

Configuration

Network

The *Chicago Hinge and Hasp* lab network consists of a collection of Intel x86 based PC, mostly custom built over a period of ten years. Two machines, a Sony laptop and an HP Pavilion are the exceptions. All machines are networked on a mix of thin-ethernet and cat-5. Two printers are connected to an HP Jetdirect 500X print server. The principal connection to the Internet is via a 3Com 3C892 ISDN LanModem to the primary ISP.



The LanModem provides for demand call setup, network address translation (NAT), dynamic host control server (DHCPD), and domain name services (DNS). See the discussion below on the DNS server configurations. The 3C892 can be, but is not, configured to accept dial-in or dial-in with call back. Incoming connection requests are ignored. The unit is configured to ignore SMB packets so that the ISDN link will not be brought up unnecessarily. It is also configured to allow incoming connection request from a source IP address only if a NAT table entry exists. The feature, called "Intelligent NAT" is useful, for example when an external message transport agent, such as sendmail, sends an ident (auth) request back during simple mail transport protocol (SMTP) handshaking. All other "unsolicited incoming packets" are dropped. As shown below in the external nmap scan of the LanModem, this filtering is quite effective. The DHCP function has been tested and works as advertised, but is not used as it would interfere with the DNS trials discussed later.

Secondary Internet connectivity is provided through a dial-up link to a secondary ISP. This service is slower than the ISDN link, but has no usage charges. It is used for long downloads for economic reasons. The analog modem is connected to a machine currently running RedHat Linux 6.2 that does not support packet forwarding either at the kernel level or through the ipchains[2] "firewall", that is to say, the default policy for the forward chain is "DENY" and no other rules for the forward chain exist in the rule set.

[2] Detailed information on ipchains (and its replacement iptables) can be found at netfilter.kernelnotes.org in *Linux 2.4 Packet Filtering HOWTO.*

The network uses 192.168.1.0/24 addressing internally.

Operating Systems

Two of the machines run Microsoft Windows 98 as their exclusive OS. The laptop is configured for dual boot to either Windows 98, which came with the unit or, at the moment a 7.1beta version of the Mandrake Linux distribution. The two "core" machines both run RedHat Linux 6.2. They are called "core" because their configurations are stable, at least relative to the remaining machines. One serves as a master DNS, network time protocol (NTP) server for the remainder of the network, and as a NFS server. The other core machine hosts a slave DNS, a peer NTP daemon, the uninterruptible power supply (UPS) master controller, and a VMWare host of Microsoft Windows 2000 Professional edition.

Four other machines run a variety of different operating systems. At the time of this audit, one machine was running Mandrake Linux 7.1, dual booting with OpenBSD 2.7. Another machine was running RedHat 7.0b (pinstripe). The third runs OpenBSD 2.7 only and the last machine is targeted to run Sun Solaris 2.8 when it gets a new motherboard that can support a fast processor. Other operating systems that are available for study, but were not installed at the time of the audit are FreeBSD 4.0, BeOS, and older version of Solaris, FreeBSD, OpenBSD, and other Linux distributions.

At the time of the audit all Microsoft Windows "critical patches[3]" had been applied and all Linux updates (both security and other) had been applied to the core systems. Non-core systems generally remain unpatched for testing reasons.

It is particularly interesting to note that the Mandrake distribution (as of 7.1) has a system security scanning script run out of cron very much like that of OpenBSD. A number of world writeable files and directories, both system files in /tmp and quite a few application files. Some of these applications files were for system utilities others were for third party applications.

Security Scans

As part of the audit, nmap[4] scans were run against all networked devices including the LanModem and the Print Server. Here are some of the results.

LanModem from inside

```
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
 Interesting ports on lanmodem.chandh.inc (192.168.1.1):
(The 1521 ports scanned but not shown below are in state: closed)
Port       State       Service
23/tcp     open        telnet
```

[3] Microsoft provides updates to its consumer operating system and applications at http://windowsupdate.microsoft.com/

[4] Information and a current copy of nmap can be found at http://www.insecure.org/nmap/. The current version as of 8/13/2000 is 2.53.

```
80/tcp     open        http

TCP Sequence Prediction: Class=64K rule
                         Difficulty=1 (Trivial joke)
Remote OS guesses: Extreme Gigabit switch (unknown version),
Router/Switch/Printer (LanPlex 2500/Cisco Catalyst 5505/CISCO
6509/Trancell Webramp/Xylan Omni Switch)/Epson Stylus (100BTX-NIC),
SunOS 4.1.1 - 4.1.4 (or derivative), VxWorks 5.3.x bases system (usually
an ethernet hub or switch), VAX/VMS 5.3 on a MicroVAX II, Xylan
OmniSwitch 5x/9x ethernet switch, Annex3 Comm server R10.0, or Hitach
HI-UX/WE2

Nmap run completed -- 1 IP address (1 host up) scanned in 24 seconds
```

The ports reported are as expected since the LanModem can be accessed for configuration via its internet http server or via telnet.

What is more interesting is the contrast between a fresh install of RedHat 6.2 and one of 7.0beta.

RedHat 6.2 with ssh added

```
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
 Interesting ports on minerva.chandh.inc (192.168.1.14):
(The 1505 ports scanned but not shown below are in state: closed)
Port       State       Service
21/tcp     open        ftp
22/tcp     open        ssh
23/tcp     open        telnet
25/tcp     open        smtp
53/tcp     open        domain
79/tcp     open        finger
80/tcp     open        http
98/tcp     open        linuxconf
111/tcp    open        sunrpc
113/tcp    open        auth
513/tcp    open        login
514/tcp    open        shell
515/tcp    open        printer
968/tcp    open        unknown
1024/tcp   open        kdm
1031/tcp   open        iad2
6000/tcp   open        X11
7000/tcp   open        afs3-fileserver

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=1564095 (Good luck!)
Remote operating system guess: Linux 2.1.122 - 2.2.14

Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
```

RedHat 7.0beta without ssh

```
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
 Interesting ports on palis.chandh.inc (192.168.1.2):
```

```
(The 1518 ports scanned but not shown below are in state: closed)
Port       State       Service
111/tcp    open        sunrpc
113/tcp    open        auth
515/tcp    open        printer
957/tcp    open        unknown
6000/tcp   open        X11

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=3235600 (Good luck!)
Remote operating system guess: Linux 2.1.122 - 2.2.14

Nmap run completed -- 1 IP address (1 host up) scanned in 2 seconds
```

Note that the default "workstation" install includes many fewer services.

To demonstrate the effectiveness of the LanModem as a packet filter, here is the nmap scan from the Internet towards the machine (minerva.chandh.inc) that was running RedHat 6.2 with ssh.

```
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Warning:  No TCP ports found open on this machine, OS detection will be
MUCH less reliable
All 1523 scanned ports on slip-32-101-208-83.dc.us.prserv.net
(32.101.208.83) are: filtered
Too many fingerprints match this host for me to give an accurate OS
guess
Nmap run completed -- 1 IP address (1 host up) scanned in 220 seconds
[root@sparta /root]#
```

Note that since no session had been established from the target (minerva) to the machine running nmap, even the auth (identd, port 113) was not visible.

Two other scans are of interest. First, an internal scan of the core machine sparta.chandh.inc (RedHat 6.2) that provides ntp, DNS, web, and NFS services among others. Following that is the scan of the same machine from the Internet with the ipchains "firewall" inplace.

Internal scan of sparta with X running.

```
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
 Interesting ports on sparta.chandh.inc (192.168.1.8):
(The 1510 ports scanned but not shown below are in state: closed)
Port       State       Service
21/tcp     open        ftp
22/tcp     open        ssh
25/tcp     open        smtp
53/tcp     open        domain
80/tcp     open        http
111/tcp    open        sunrpc
113/tcp    open        auth
515/tcp    open        printer
896/tcp    open        unknown
```

```
901/tcp    open        samba-swat
972/tcp    open        unknown
1024/tcp   open        kdm
2003/tcp   open        cfingerd

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=2370772 (Good luck!)
Remote operating system guess: Linux 2.1.122 - 2.2.14

Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
```

External scan of sparta over dial-up ppp connection with ipchains "firewall".

```
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
 Warning:  No TCP ports found open on this machine, OS detection will be
MUCH less reliable
All 975 scanned ports on sparta.chandh.inc (192.168.1.8) are: filtered
Too many fingerprints match this host for me to give an accurate OS
guess
Nmap run completed -- 1 IP address (1 host up) scanned in 1413 seconds
```

Note that the internal scan of Sparta took 1 second to find 13 ports (it did not detect port 123, the NTP port) while the external scan took 1413 seconds.

ipchains firewall rules

The "firewall" rules focus on blocking incoming requests and logging them. The rules also enforce both ingress and egress filtering. The systems administrator derived the rules in current use from a model created by Robert L. Ziegler with modifications based on suggestion found in the messages in the comp.os.linux.security news group[5]. The listing is long, but is presented here (as an Attachment) in hopes it will be of some use to other SOHO system administrators.

"Firewall" log analysis

The systems administrator provided the following documentation of a probe that was picked up on June 17th. It appears to be someone on a dial-up connection in Hungary looking for an open POP2 server.

```
Jun 17 10:33:10 sparta kernel: Packet log: input DENY ppp0 PROTO=6
145.236.213.226:109 209.8.42.80:109 L=40 S=0x00 I=39426 F=0x0000 T=30
SYN (#26)

[root@sparta log]# grep 109\/ /etc/services
pop2        109/tcp          pop-2 postoffice  # POP version 2
pop2        109/udp          pop-2
kpop        1109/tcp               # Pop with Kerberos
```

[5] Other news groups that the systems administrator follows include: comp.protocols.dns.bind and comp.protocols.time.ntp. The CERT/CC website (http://www.cert.org) is also frequently visited.

```
[root@sparta log]# dig -x 145.236.213.226

; <<>> DiG 8.2 <<>> -x
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 3
;; QUERY SECTION:
;;    226.213.236.145.in-addr.arpa, type = ANY, class = IN

;; ANSWER SECTION:
226.213.236.145.in-addr.arpa.  4D IN PTR  line-213-226.dial.matav.net.

;; AUTHORITY SECTION:
236.145.in-addr.arpa.   4D IN NS    ns1.matav.net.
236.145.in-addr.arpa.   4D IN NS    ns1.elender.hu.
236.145.in-addr.arpa.   4D IN NS    ns0.matav.net.

;; ADDITIONAL SECTION:
ns1.matav.net.          1D IN A           145.236.224.249
ns1.elender.hu.         1H IN A           212.108.200.66
ns0.matav.net.          1D IN A           145.236.224.248

;; Total query time: 6726 msec
;; FROM: sparta.chandh.inc to SERVER: default -- 192.168.1.8
;; WHEN: Sat Jun 17 10:36:19 2000
;; MSG SIZE  sent: 46  rcvd: 219
```

Domain Name Services

Chicago Hinge and Hasp runs an internal name server service. This was done primarly to provide for education and testing of new releases of Bind, but also to prevent reverse queries for 192.168.0.0/16 in-addr.arpa from leaking into the Internet through the LanModem. The current version of LanModem software provides for reverse lookup. Bind[6] version 8.2.2p5 is in use for the slave servers and 9.0.0rc1[7] was in use as the master at the time of the audit. Named is not run as root, but it is not chroot'd. Access controls to restrict zone transfers and external queries are in place as shown in the named.conf file[8]:

```
options {
      directory "/var/named";
};
```

[6] The Internet Software Consortium (ISC) is the home of the reference implementation of Bind (http://www.isc.org).

[7] Note that the Release Candidates for version 9.0.0, both rc1 and rc2, fail to run when started with '-u <username>' on Linux kernels older than 2.3.99-pre3. They will run with out the –u option.

[8] Excellent information in securing DNS can be found at Cricket Liu's web site, which is http://www.acmebw.com. (Note that Network Solutions Registery has acquired Acme Byte & Wire.) Look for two things: one, follow the Papers and Presentations link to Securing Your Name Server; and two, http://www.acmebw.com/askmrdns/bind-messages.htm provides an annotated list of named error messages.

```
acl to_secondaries { 192.168.1.14; 192.168.1.3; };

zone "." {
      type hint;
      file "named.ca";
};

zone "chandh.inc."{
      type master;
      file "chandh.inc.";
      allow-transfer{
            to_secondaries;
      };
      allow-query{
            192.168.1.0/24 ;
      };
      also-notify{
            192.168.1.14;
            192.168.1.3;
      };
};

zone "0.0.127.in-addr.arpa"{
      type master;
      file "named.local";
};

zone "1.168.192.IN-ADDR.ARPA"{
      type master;
      file "192.168.1.0";
      allow-transfer{
            to_secondaries;
      };
      allow-query{
            192.168.1.0/24 ;
      };
      also-notify{
            192.168.1.14;
            192.168.1.3;
      };
};
```


Policy

The organization has no written policy, but a well understood set of operating practices founded on "be suspicious", no remote access, no external services (e.g. web server, DNS, NTP, etc.), and "deny all, except what is allowed". This was justified by the fact that there are only two individuals involved with over 35 years of working together and about 35 years of combined computer security practice. SMB shares are restricted, "mobile code" (e.g., java and active-X) services are generally disabled, and Microsoft applications (e.g., Outlook/Outlook Express, Internet Explorer, Word, etc.) are generally avoided. Files can only be moved between the Microsoft Windows and the "Unix" domains through a non-anonymous ftp server that runs on a "core" machine".

In discussions with the *Chicago Hinge and Hasp* staff, it became clear that while password selection practices were consistent with best practices, password reuse and ageing were not.

Continuity of Operations

The two Microsoft Windows boxes each have their own UPS. The two "core" and two of the test systems (four boxes) share an UPS and through a KVM switch, they share a display, keyboard, and mouse. Spare equipment is available. SSH (OpenSSH on the Linux boxes and F-Secure on one of the Microsoft Windows boxes) provides for terminal and tunneled X access as well.

Personal

Only two individuals have unmonitored access to all systems. In person "Guests" are only allowed access to one particular Microsoft Windows machine. As noted above, in the Policy paragraph, these individuals have worked together and in the computer security field for 35 years. Both have had background investigations and both have worked for or on government IT systems. One of the individuals would be an excellent instructor if SANS elected to broaden their offering to include "main frame" security.

Site Security

All equipment is housed at a single location with physical security appropriate for the type of laboratory operations. A fire retardant safe is available of storage of removable media and a two-function paper shredder is used to dispose of sensitive documents. An incinerator is also available.

Vulnerabilities

Given the basically closed operating environment coupled with the apparent strength of the LanModem's and the ipchains "firewall" packet filtering, the *Chicago Hinge and Hasp* network appears to be well masked from external threats. Vigilance is required, however, since one of the purposes of the network is to function as a test and training laboratory. The security of "out-of-the-box" systems is not generally good (with the exception of OpenBSD). Down loading, compiling, and installing beta versions of system daemons have risks that unknown vulnerabilities will be added to the network. The systems administrator does run additional security tools such as SAINT, nessus, and snort from time to time. They should continue to do so with each new system installation and after any material configuration change.

Recommendations and Cost Estimate

Password practices should be improved to reduce reuse across systems and appropriate ageing rules put in place. Estimate 2 staff days.

Backup and recovery practices should be reviewed and amended as appropriate. Estimate 1 staff day.

Greater use of journaling file systems should be adopted and RAID storage for important directory/files should be considered. Estimate 1 staff week and US$6,000.00 for cabinet, controller, and drives. If software RAID is adequate the cost estimate is US$4,000.00.

If future business operations should require a visible web presence, then a DMZ should be introduced to separate the "test laboratory" further from the Internet. Estimate 1 staff week and US$1,000.00 for an Ethernet switch.

Additional Resource for Securing Linux

Linux System Security, The Administrator's Guide to Open Source Security Tools, by Scott Mann and Ellen L. Mitchell, (ISBN 0-13-015807-0) is recommended.

Attachment – rc.firewall script

This script is used to secure a dial-up ppp connection for a single machine on a small office/home office (SOHO) network. As written, it does not support packet forwarding or network address translation. It is a modification of an original work of Robert L. Ziegler.

```
#!/bin/sh

# ------------------------------------------------------------------
# Copyright (C) 1997, 1998, 1999, 2000  Robert L. Ziegler
#
#  Permission to use, copy, modify, and distribute this software and its
#  documentation for educational, research, private and non-profit
purposes,
#  without fee, and without a written agreement is hereby granted.
#  This software is provided as an example and basis for individual
firewall
#  development.  This software is provided without warranty.
#
#  Any material furnished by Robert L. Ziegler is furnished on an
#  "as is" basis.  He makes no warranties of any kind, either expressed
#  or implied as to any matter including, but not limited to, warranty
#  of fitness for a particular purpose, exclusivity or results obtained
#  from use of the material.
# ------------------------------------------------------------------

#  /etc/rc.d/rc.firewall
#  Invoked from /etc/ppp/ip-up, or
#  from /sbin/ifup-local, or
#  from /etc/sysconfig/network-scripts/ifup-post.

echo "Starting firewalling... "

# ------------------------------------------------------------------
#  Some definitions for easy maintenance.
#  EDIT THESE TO SUIT YOUR SYSTEM AND ISP.

EXTERNAL_INTERFACE="ppp0"           # Internet connected interface
LOOPBACK_INTERFACE="lo"             # or your local naming convention
INTERNAL_INTERFACE="eth0"           # and internal network connecton

# the cut -c 21-35 did not always work with short IP addresses
IPADDR=$(/sbin/ifconfig | grep -A 4 ppp0 | awk '/inet/ { print $2 } ' |
sed -e s/addr://)
# so I commented this line out  (2000/06/17 jag)
# IPADDR=$(/sbin/ifconfig | /bin/grep P-t-P | /usr/bin/cut -c 21-35)

ANYWHERE="any/0"              # match any IP address

DHCP_SERVER="any/0"
NAMESERVER_1="any/0"                # everyone must have at least one
NAMESERVER_2="any/0"                # or more
NAMESERVER_3="any/0"                #
```

```
NETWORK_PRINTER="hqprinters.chandh.inc"   # Remote print server

SMTP_SERVER="any/0"                 # Your ISP mail gateway. Your relay.
POP_SERVER="cpcug.org"         # Your ISP pop mail server.
WEB_PROXY_SERVER="127.0.0.1"        # Your ISP web proxy server

LOOPBACK="127.0.0.0/8"              # reserved loopback address range
CLASS_A="10.0.0.0/8"                # class A private networks
CLASS_B="172.16.0.0/12"             # class B private networks
CLASS_C="192.168.0.0/16"            # class C private networks
BROADCAST_SRC="0.0.0.0"             # broadcast source address
BROADCAST_DEST="255.255.255.255"    # broadcast destination address
PRIVPORTS="0:1023"                  # well known, privileged port range
UNPRIVPORTS="1024:65535"            # unprivileged port range

# ---------------------------------------------------------------------

NFS_PORT="2049"                     # (TCP/UDP) NFS
SOCKS_PORT="1080"             # (TCP) Socks

# X Windows port allocation begins at 6000 and increments to 6063
# for each additional server running.
XWINDOW_PORTS="6000:6063"           # (TCP) X windows

# The SSH client starts at 1023 and works down to 513 for each
# additional simultaneous connection originating from a privileged port.
# Clients can optionally be configured to use only unprivileged ports.
SSH_REMOTE_PORTS="1022:65535"       # port range for local clients
SSH_LOCAL_PORTS="513:65535"         # port range for remote clients

# traceroute usually uses -S 32769:65535 -D 33434:33523
TRACEROUTE_SRC_PORTS="32769:65535"
TRACEROUTE_DEST_PORTS="33434:33523"

# ---------------------------------------------------------------------
# Default policy is DENY
# Explicitly accept desired INCOMING & OUTGOING connections

    # Remove all existing rules belonging to this filter
    ipchains -F

    # Set the default policy of the filter to deny.
    ipchains -P input  DENY
    ipchains -P output REJECT
    ipchains -P forward DENY

# ---------------------------------------------------------------------

    # Enable TCP SYN Cookie Protection
    echo 1 > /proc/sys/net/ipv4/tcp_syncookies

    # Enable always defragging Protection
    echo 1 > /proc/sys/net/ipv4/ip_always_defrag

    # Enable broadcast echo  Protection
    echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
```

```
    # Enable bad error message  Protection
    echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses

    # Enable IP spoofing protection
    # turn on Source Address Verification
    for f in /proc/sys/net/ipv4/conf/*/rp_filter; do
        echo 1 > $f
    done

    # Disable ICMP Redirect Acceptance
    for f in /proc/sys/net/ipv4/conf/*/accept_redirects; do
        echo 0 > $f
    done

    # Disable Source Routed Packets
    for f in /proc/sys/net/ipv4/conf/*/accept_source_route; do
        echo 0 > $f
    done

    # Log Spoofed Packets, Source Routed Packets, Redirect Packets
    for f in /proc/sys/net/ipv4/conf/*/log_martians; do
        echo 1 > $f
    done


# ------------------------------------------------------------------
# LOOPBACK

    # Unlimited traffic on the loopback interface.

    ipchains -A input  -i $LOOPBACK_INTERFACE  -j ACCEPT
    ipchains -A output -i $LOOPBACK_INTERFACE  -j ACCEPT

# ------------------------------------------------------------------
# INTERNAL LAN

    # Unlimited traffic on the internal (eth0) interface.

    ipchains -A input  -i $INTERNAL_INTERFACE  -j ACCEPT
    ipchains -A output -i $INTERNAL_INTERFACE  -j ACCEPT

# ------------------------------------------------------------------
# SPOOFING & BAD ADDRESSES
# Refuse spoofed packets.
# Ignore blatantly illegal source addresses.
# Protect yourself from sending to bad addresses.

    # Refuse incoming packets pretending to be from the external
address.
    ipchains -A input -i $EXTERNAL_INTERFACE  -s $IPADDR -j DENY -l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s $CLASS_A -j DENY -l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s $CLASS_B -j DENY -l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s $CLASS_C -j DENY -l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s 224.0.0.0/4 -j DENY -l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s 240.0.0.0/5 -j DENY -l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s 1.0.0.0/7 -j DENY -l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s 2.0.0.0/8 -j DENY -l
```

```
    ipchains -A input -i $EXTERNAL_INTERFACE  -s 5.0.0.0/8 -j DENY -l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s 7.0.0.0/8 -j DENY -l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s 23.0.0.0/8 -j DENY -l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s 27.0.0.0/8 -j DENY -l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s 31.0.0.0/8 -j DENY -l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s 37.0.0.0/8 -j DENY -l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s 39.0.0.0/8 -j DENY -l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s 41.0.0.0/8 -j DENY -l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s 42.0.0.0/8 -j DENY -l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s 58.0.0.0/7 -j DENY -l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s 60.0.0.0/8 -j DENY -l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s 65.0.0.0/8 -j DENY -l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s 66.0.0.0/7 -j DENY -l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s 68.0.0.0/6 -j DENY -l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s 72.0.0.0/5 -j DENY -l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s 80.0.0.0/4 -j DENY -l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s 96.0.0.0/3 -j DENY -l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s 169.254.0.0/16 -j DENY
-l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s 192.0.2.0/24 -j DENY -l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s 217.0.0.0/8 -j DENY -l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s 218.0.0.0/7 -j DENY -l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s 220.0.0.0/6 -j DENY -l
    ipchains -A input -i $EXTERNAL_INTERFACE  -s 248.0.0.0/5 -j DENY -l

# ------------------------------------------------------------------
# NOTE:
#      The symbolic names used in /etc/services for the port numbers
vary by
#      supplier.  Using them is less error prone and more meaningful,
though.

# ------------------------------------------------------------------
# TCP UNPRIVILEGED PORTS
# Avoid ports subject to protocol & system administration problems.

    # NFS: establishing a TCP connection
    ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp -y \
             --destination-port $NFS_PORT -j DENY -l
    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp -y \
             --destination-port $NFS_PORT -j REJECT

    # Xwindows: establishing a connection
    ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp -y \
             --destination-port $XWINDOW_PORTS -j DENY -l
    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp -y \
             --destination-port $XWINDOW_PORTS -j REJECT

    # SOCKS: establishing a connection
    ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp -y \
             --destination-port $SOCKS_PORT -j DENY -l
    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp -y \
             --destination-port $SOCKS_PORT -j REJECT

# ------------------------------------------------------------------
# UDP UNPRIVILEGED PORTS
# Avoid ports subject to protocol & system administration problems.
```

```
    ipchains -A input  -i $EXTERNAL_INTERFACE -p udp  \
             --destination-port $NFS_PORT -j DENY -l

    # UDP INCOMING TRACEROUTE
    # traceroute usually uses -S 32769:65535 -D 33434:33523

    ipchains -A input  -i $EXTERNAL_INTERFACE -p udp  \
             --source-port $TRACEROUTE_SRC_PORTS \
             --destination-port $TRACEROUTE_DEST_PORTS -j DENY -l

# ------------------------------------------------------------------

    ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
             -d $IPADDR -j ACCEPT

# ------------------------------------------------------------------

# DNS server (53)
# ---------------

    # DNS: full server
    # ----------------

    # server/client to server query or response

    ipchains -A input  -i $EXTERNAL_INTERFACE -p udp  \
             --source-port $UNPRIVPORTS \
             -d $IPADDR 53 -j ACCEPT

    ipchains -A output -i $EXTERNAL_INTERFACE -p udp  \
             -s $IPADDR 53 \
             --destination-port $UNPRIVPORTS -j ACCEPT

    ipchains -A output -i $EXTERNAL_INTERFACE -p udp  \
             -s $IPADDR 53 \
             --destination-port 53 -j ACCEPT

    ipchains -A input  -i $EXTERNAL_INTERFACE -p udp  \
             --source-port 53 \
             -d $IPADDR 53 -j ACCEPT


    # DNS client (53)
    # ---------------
    ipchains -A output -i $EXTERNAL_INTERFACE -p udp  \
             -s $IPADDR $UNPRIVPORTS \
             -d $NAMESERVER_1 53 -j ACCEPT

    ipchains -A input  -i $EXTERNAL_INTERFACE -p udp  \
             -s $NAMESERVER_1 53 \
             -d $IPADDR $UNPRIVPORTS -j ACCEPT


    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR $UNPRIVPORTS \
             -d $NAMESERVER_1 53 -j ACCEPT
```

```
    # ------------------------------------------------------------------

    # HTTP client (80)
    # ----------------
    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR $UNPRIVPORTS \
             --destination-port 80 -j ACCEPT

    # ------------------------------------------------------------------

    # HTTPS client (443)
    # ------------------
    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR $UNPRIVPORTS \
             --destination-port 443 -j ACCEPT

    # ------------------------------------------------------------------

    # WWW-CACHE client
    # ----------------
    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR $UNPRIVPORTS \
             -d $WEB_PROXY_SERVER 8000 -j ACCEPT

    # ------------------------------------------------------------------

    # POP client (110)
    # ----------------
    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR $UNPRIVPORTS \
             -d $POP_SERVER 110 -j ACCEPT

    # ------------------------------------------------------------------

    # SMTP client (25)
    # ----------------
    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR $UNPRIVPORTS \
             -d $SMTP_SERVER 25 -j ACCEPT

    # ------------------------------------------------------------------

    # SSH client (22)
    # ---------------
    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR $SSH_LOCAL_PORTS \
             --destination-port 22 -j ACCEPT

    # ------------------------------------------------------------------

    # AUTH server (113)
    # -----------------

    # Accept incoming connections to identd but disable in.identd in
inetd.conf.
    ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp  \
```

```
            --source-port $UNPRIVPORTS \
            -d $IPADDR 113 -j ACCEPT

    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
            -s $IPADDR 113 \
            --destination-port $UNPRIVPORTS -j ACCEPT


    # AUTH client (113)
    # -----------------
    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp  \
            -s $IPADDR $UNPRIVPORTS \
            --destination-port 113 -j ACCEPT

    # ------------------------------------------------------------------

    # WHOIS client (43)
    # -----------------
    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp  \
            -s $IPADDR $UNPRIVPORTS \
            --destination-port 43 -j ACCEPT

    # ------------------------------------------------------------------

    # FTP client (21)
    # ---------------

    # outgoing request
    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp  \
            -s $IPADDR $UNPRIVPORTS \
            --destination-port 21 -j ACCEPT


    # PORT mode data channel
    ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp  \
            --source-port 20 \
            -d $IPADDR $UNPRIVPORTS -j ACCEPT

    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
            -s $IPADDR $UNPRIVPORTS \
            --destination-port 20 -j ACCEPT


    # PASSIVE mode data channel creation
    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp  \
            -s $IPADDR $UNPRIVPORTS \
            --destination-port $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------
# UDP accept only on selected ports
# ---------------------------------

    # ------------------------------------------------------------------

    # NTP TIME clients (123)
    # ----------------------
    ipchains -A output -i $EXTERNAL_INTERFACE -p udp  \
```

```
            -s $IPADDR  123 \
            -d 0.0.0.0/0 123 -j ACCEPT

    ipchains -A input  -i $EXTERNAL_INTERFACE -p udp  \
            -s 0.0.0.0/0 123 \
            -d $IPADDR 123 -j ACCEPT

    # ------------------------------------------------------------------

    # OUTGOING TRACEROUTE
    # -------------------
    ipchains -A output -i $EXTERNAL_INTERFACE -p udp  \
            -s $IPADDR $TRACEROUTE_SRC_PORTS \
            --destination-port $TRACEROUTE_DEST_PORTS -j ACCEPT -l

# ------------------------------------------------------------------
# ICMP

    #    To prevent denial of service attacks based on ICMP bombs,
filter
    #    incoming Redirect (5) and outgoing Destination Unreachable (3).
    #    Note, however, disabling Destination Unreachable (3) is not
    #    advisable, as it is used to negotiate packet fragment size.

    # For bi-directional ping.
    #     Message Types:  Echo_Reply (0),  Echo_Request (8)
    #     To prevent attacks, limit the src addresses to your ISP range.
    #
    # For outgoing traceroute.
    #     Message Types:  INCOMING Dest_Unreachable (3), Time_Exceeded
(11)
    #     default UDP base: 33434 to base+nhops-1
    #
    # For incoming traceroute.
    #     Message Types:  OUTGOING Dest_Unreachable (3), Time_Exceeded
(11)
    #     To block this, deny OUTGOING 3 and 11

    #  0: echo-reply (pong)
    #  3: destination-unreachable, port-unreachable, fragmentation-
needed, etc.
    #  4: source-quench
    #  5: redirect
    #  8: echo-request (ping)
    # 11: time-exceeded
    # 12: parameter-problem

    ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp  \
            --icmp-type echo-reply \
            -d $IPADDR -j ACCEPT

    ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp  \
            --icmp-type destination-unreachable \
            -d $IPADDR -j ACCEPT

    ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp  \
            --icmp-type source-quench \
```

```
            -d $IPADDR -j ACCEPT

    ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp  \
            --icmp-type time-exceeded \
            -d $IPADDR -j ACCEPT

    ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp  \
            --icmp-type parameter-problem \
            -d $IPADDR -j ACCEPT


    ipchains -A output -i $EXTERNAL_INTERFACE -p icmp  \
            -s $IPADDR fragmentation-needed -j ACCEPT

    ipchains -A output -i $EXTERNAL_INTERFACE -p icmp  \
            -s $IPADDR source-quench -j ACCEPT

    ipchains -A output -i $EXTERNAL_INTERFACE -p icmp  \
            -s $IPADDR echo-request -j ACCEPT

    ipchains -A output -i $EXTERNAL_INTERFACE -p icmp  \
            -s $IPADDR parameter-problem -j ACCEPT

# ------------------------------------------------------------------
# Enable logging for selected denied packets

    ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp  -j DENY -l

    ipchains -A input  -i $EXTERNAL_INTERFACE -p udp  \
            --destination-port $PRIVPORTS -j DENY -l

    ipchains -A input  -i $EXTERNAL_INTERFACE -p udp  \
            --destination-port $UNPRIVPORTS -j DENY -l


    ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp  \
            --icmp-type 5 -j DENY -l
    ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp  \
            --icmp-type 13:255 -j DENY -l

    ipchains -A output -i $EXTERNAL_INTERFACE  -j REJECT -l

# ------------------------------------------------------------------

echo "done"

exit 0
```

Last Updated: July 20th, 2017

Upcoming Training

SANSFIRE 2017	**Washington, DC**	**Jul 22, 2017 - Jul 29, 2017**	**Live Event**
SANS Network Security 2017	**Las Vegas, NV**	**Sep 10, 2017 - Sep 17, 2017**	**Live Event**
SANS OnDemand	**Online**	**Anytime**	**Self Paced**
SANS SelfStudy	**Books & MP3s Only**	**Anytime**	**Self Paced**