Linux sysctl hardening and network tuning

Vložil/a cm3l1k1 [1], 26 Duben, 2011 - 22:06

- GNU/Linux a BSD [2]
- Networks & Protocols [3]
- Security [4]

This article will show you how you can tune-up system security and network stack via sysctl.conf file on Linux-based operating systems because by default are kernel parameters very general and as other applications need some tuning to achieve better performance and system security.

I hope you will find it useful because I have spent many hours searching on websites and testing. You can download file also as attachment.

Newest version should be found here: http://wiki.securix.org/doku.php/etc_sysctl.conf

Every feedback is appreciated. Thx

```
# Kernel sysctl configuration file for Securix GNU/Linux
# File: /etc/sysctl.conf
#
# Build with focus on security and network stack tuning
#
# This is part of Securix GNU/Linux (www.securix.org)
#
# Author: Martin Cmelik (cm3l1k1) - www.security-portal.cz
# Version: 0.1
# Credits: cyberciti.biz, fasterdata.es.net
# ..and next ±20 websites
#
# For binary values, 0 is disabled, 1 is enabled.  See sysctl(8) and
# sysctl.conf(5) for more details.


########## Kernel config START ###############

# Controls whether core dumps will append the PID to the core filename.
# Useful for debugging multi-threaded applications.
kernel.core_uses_pid = 1

# Kernel EXEC shield
kernel.exec-shield = 1
kernel.randomize_va_space = 2

# Reboot system when kernel panic occur, oops will wait 30 seconds untill call
panic()
kernel.panic = 30
kernel.panic_on_oops = 30

# Disables the magic-sysrq key
kernel.sysrq = 0

# No core dumps for SUID
```

```
fs.suid_dumpable = 0

# Set maximum amount of memory allocated to shm to 256MB
kernel.shmmax = 268435456

# Modify system limits for Ensim WEBppliance, check #cat /proc/sys/fs/file-nr where
is number of actual used files
fs.file-max = 65000

#Allow for more PIDs
kernel.pid_max = 65536

########## Kernel config END ###############

########## IPv4 networking START ###############

# Controls IP packet forwarding
net.ipv4.ip_forward = 0

# Disable Proxy ARP
net.ipv4.proxy_arp = 0

# Decrease the time default value for tcp_fin_timeout connection
net.ipv4.tcp_fin_timeout = 15

# Decrease the time default value for tcp_keepalive_time connection
net.ipv4.tcp_keepalive_time = 1800

# Enable tcp_window_scaling
net.ipv4.tcp_window_scaling = 1

# Turn off the tcp_sack
net.ipv4.tcp_sack = 0

# Turn off the tcp_timestamps
net.ipv4.tcp_timestamps = 0

# Enable ignoring broadcasts request
net.ipv4.icmp_echo_ignore_broadcasts = 1

# Enable bad error message protection
net.ipv4.icmp_ignore_bogus_error_responses = 1

# This removes an odd behavior in the 2.6 kernels, whereby the kernel stores
# the slow start threshold for a client between TCP sessions.
net.ipv4.tcp_no_metrics_save=1

# Prevent SYN attack
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_max_syn_backlog = 4096
net.ipv4.tcp_syn_retries = 5
net.ipv4.tcp_synack_retries = 2

# Disable send redirects
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0

# Enable IP spoofing protection, turn on source route verification
net.ipv4.conf.all.rp_filter = 1
```

```
net.ipv4.conf.default.rp_filter = 1

# Enable redirects from gateways known in our routing table
net.ipv4.conf.all.secure_redirects = 1
net.ipv4.conf.default.secure_redirects = 1

# Disable ICMP Redirect Acceptance
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.default.accept_redirects = 0

# Log packets with impossible addresses to kernel
net.ipv4.conf.all.log_martians = 1
net.ipv4.conf.default.log_martians = 1

# Disable IP source routing
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.default.accept_source_route = 0

# Buffer size autotuning - buffer size (and tcp window size) is dynamically updated
for each connection.
# This option is not present in kernels older then 2.4.27 or 2.6.7 - update your
kernel
# In that case tuning options net.ipv4.tcp_wmem and net.ipv4.tcp_rmem isnt
recommended
net.ipv4.tcp_moderate_rcvbuf = 1

# Increase the tcp-time-wait buckets pool size
net.ipv4.tcp_max_tw_buckets = 1440000

# Increase allowed local port range
net.ipv4.ip_local_port_range = 1024 64000

# Increase the maximum memory used to reassemble IP fragments
net.ipv4.ipfrag_high_thresh = 512000
net.ipv4.ipfrag_low_thresh = 446464

# Increase the maximum amount of option memory buffers
net.core.optmem_max = 57344

########## IPv4 networking END ###############

########## IPv6 networking START ###############

# Number of Router Solicitations to send until assuming no routers are present.
# This is host and not router
net.ipv6.conf.default.router_solicitations = 0

# Accept Router Preference in RA?
net.ipv6.conf.default.accept_ra_rtr_pref = 0

# Learn Prefix Information in Router Advertisement
net.ipv6.conf.default.accept_ra_pinfo = 0

# Setting controls whether the system will accept Hop Limit settings from a router
advertisement
net.ipv6.conf.default.accept_ra_defrtr = 0

# Router advertisements can cause the system to assign a global unicast address to an
interface
```

```
net.ipv6.conf.default.autoconf = 0

# How many neighbor solicitations to send out per address?
net.ipv6.conf.default.dad_transmits = 0

# How many global unicast IPv6 addresses can be assigned to each interface?
net.ipv6.conf.default.max_addresses = 1

########## IPv6 networking END ##############
```

URL článku: https://www.security-portal.cz/node/3553

Odkazy:
[1] https://www.security-portal.cz/users/cm3l1k1
[2] https://www.security-portal.cz/category/tagy/gnu/linux-bsd
[3] https://www.security-portal.cz/category/tagy/networks-protocols
[4] https://www.security-portal.cz/category/tagy/security