Creating server parameter sysctl.conf for WebServers / FTP Server.

`sysctl` is an interface that allows you to make changes to a running Linux kernel. With /etc/sysctl.conf you can configure various Linux networking and system settings such as:

- Limit network-transmitted configuration for IPv4
- Limit network-transmitted configuration for IPv6
- Turn on execshield protection
- Prevent against the common `syn flood attack`
- Turn on source IP address verification
- Prevents a cracker from using a spoofing attack against the IP address of the server.
- Logs several types of suspicious packets, such as spoofed packets, source-routed packets, and redirects.

IMPROVE SYSTEM MEMORY MANAGEMENT

1. `fs.file-max` Increase size of file handles and inode cache

```
# Increase size of file handles and inode cache
fs.file-max = 2097152
```

1. `vm.dirty_ratio` setting virtual memory ratio.
2. `vm.swappiness` How often swap should be used. 0 is least, 60 default. We set it to 10.
3. `vm.dirty_background_ratio` contains 10, which is a percentage of total system memory, the number of pages at which the `pdflush` background `writeback` daemon will start writing out dirty data. However, for fast RAID based disk system this may cause large flushes of dirty memory pages. If you increase this value from 10 to 20 (a large value) will result into less frequent flushes. We can start with 2, being frequent flushes, this is what we need for a web/nginx system.

```
# Do less swapping
vm.dirty_ratio = 60
vm.swappiness = 10
vm.dirty_background_ratio = 2
```

GENERAL NETWORK SECURITY OPTIONS

1. `net.ipv4.tcp_sack` Disable select acknowledgments
2. `net.ipv4.tcp_dsack` Allows TCP to send “duplicate” SACKs.
3. `net.ipv4.tcp_keepalive_time` How often TCP sends out keepalive messages when keepalive is enabled. Default: 2hours.
4. `net.ipv4.tcp_keepalive_probes` How many keepalive probes TCP sends out, until it decides that the connection is broken. Default value: 9.
5. `net.ipv4.tcp_keepalive_intvl` How frequently the probes are send out. Multiplied by `tcp_keepalive_probes` it is time to kill not responding connection, after probes started. Default value: 75sec i.e. connection will be aborted after ~11 minutes of retries.

6. `net.ipv4.tcp_fin_timeout` Time to hold socket in state FIN-WAIT-2, if it was closed by our side. Peer can be broken and never close its side, or even died unexpectedly. Default value is 60sec. Usual value used in 2.2 was 180 seconds, you may restore it, but remember that if your machine is even underloaded WEB server, you risk to overflow memory with kilotons of dead sockets, FIN-WAIT-2 sockets are less dangerous than FIN-WAIT-1, because they eat maximum 1.5K of memory, but they tend to live longer. `Cf.tcp_max_orphans`.
7. `net.ipv4.tcp_rmem` The three values setting the minimum, initial, and maximum size of the Memory Receive Buffer per connection. They define the actual memory usage, not just TCP window size.
8. `net.ipv4.tcp_wmem` The same as tcp_rmem, but just for Memory Send Buffer per connection.
9. `net.ipv4.tcp_retries2` This value influences the timeout of an alive TCP connection, when RTO retransmissions remain unacknowledged. Given a value of N, a hypothetical TCP connection following exponential backoff with an initial RTO of `TCP_RTO_MIN` would retransmit N times before killing the connection at the (N+1)th RTO. The default value of 15 yields a hypothetical timeout of 924.6 seconds and is a lower bound for the effective timeout. TCP will effectively time out at the first RTO which exceeds the hypothetical timeout. RFC 1122 recommends at least 100 seconds for the timeout, which corresponds to a value of at least 8.
10. `net.ipv4.tcp_synack_retries` Number of times SYNACKs for a passive TCP connection attempt will be retransmitted. Should not be higher than 255. Default value is 5, which corresponds to ~180seconds.
11. `net.ipv4.ip_local_port_range` - 2 INTEGERS Defines the local port range that is used by TCP and UDP to choose the local port. The first number is the first, the second the last local port number. Default value depends on amount of memory available on the system: 1. `128Mb 32768-61000`, 2. `128Mb 1024-4999 or even less.` This number defines number of active connections, which this system can issue simultaneously to systems not supporting TCP extensions (timestamps). With `tcp_tw_recycle` enabled (i.e. by default) range 1024-4999 is enough to issue up to 2000 connections per second to systems supporting timestamps.
12. `net.ipv4.tcp_rfc1337` - BOOLEAN If set, the TCP stack behaves conforming to RFC1337. If unset, we are not conforming to RFC, but prevent TCP `TIME_WAIT` assassination. Default: 0
13. `net.ipv4.tcp_fin_timeout` - INTEGER Time to hold socket in state FIN-WAIT-2, if it was closed by our side. Peer can be broken and never close its side, or even died unexpectedly. Default value is 60sec. Usual value used in 2.2 was 180 seconds, you may restore it, but remember that if your machine is even underloaded WEB server, you risk to overflow memory with kilotons of dead sockets, FIN-WAIT-2 sockets are less dangerous than FIN-WAIT-1, because they eat maximum 1.5K of memory, but they tend to live longer. Cf. `tcp_max_orphans`.
14. `net.core.somaxconn` - INTEGER Limit of socket listen() backlog, known in userspace as SOMAXCONN. Defaults to 128. See also `tcp_max_syn_backlog` for additional tuning for TCP sockets.

```
# Number of times SYNACKs for passive TCP connection.
net.ipv4.tcp_synack_retries = 2

# Allowed local port range
net.ipv4.ip_local_port_range = 2000 65535

# Protect Against TCP Time-Wait
net.ipv4.tcp_rfc1337 = 1

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

# Decrease the time default value for connections to keep alive
```

```
net.ipv4.tcp_keepalive_time = 300
net.ipv4.tcp_keepalive_probes = 5
net.ipv4.tcp_keepalive_intvl = 15

### TUNING NETWORK PERFORMANCE ###

# Default Socket Receive Buffer - NOTE: This will be overridden by tcp_rmem
net.core.rmem_default = 31457280

# Maximum Socket Receive Buffer - NOTE: This value will NOT be Overridden by tcp_rmem
net.core.rmem_max = 12582912

# Default Socket Send Buffer - NOTE: This will be overridden by tcp_wmem
net.core.wmem_default = 31457280

# Maximum Socket Send Buffer - NOTE: This value will NOT be Overridden by tcp_wmem
net.core.wmem_max = 12582912

# Increase number of incoming connections
net.core.somaxconn = 65536

# Increase number of incoming connections backlog
net.core.netdev_max_backlog = 65536

# Increase the maximum amount of option memory buffers
net.core.optmem_max = 25165824

# Increase the maximum total buffer-space allocatable
# This is measured in units of pages (4096 bytes)
net.ipv4.tcp_mem = 65536 131072 262144
net.ipv4.udp_mem = 65536 131072 262144

# Increase the read-buffer space allocatable
net.ipv4.tcp_rmem = 8192 87380 16777216
net.ipv4.udp_rmem_min = 16384

# Increase the write-buffer-space allocatable
net.ipv4.tcp_wmem = 8192 65536 16777216
net.ipv4.udp_wmem_min = 16384

# Increase the tcp-time-wait buckets pool size to prevent simple DOS attacks
net.ipv4.tcp_max_tw_buckets = 1440000
net.ipv4.tcp_tw_recycle = 1
net.ipv4.tcp_tw_reuse = 1
```