kismet.conf

Original

The original config file can be found at /etc/kismet.conf (ArchLinux) or /etc/kismet/kismet.conf (Debian/Ubuntu), it depends on your OS. Other options apply too.

<sxh bash; title:Original kismet.conf;> # Kismet config file # Most of the "static" configs have been moved to here – the command line # config was getting way too crowded and cryptic. We want functionality, # not continually reading –help!

Version of Kismet config version=2009-newcore

Name of server (Purely for organizational purposes) # If commented out, defaults to host name of system # servername=Kismet Server

Prefix of where we log (as used in the logtemplate later) # logprefix=/some/path/to/logs

Do we process the contents of data frames? If this is enabled, data # frames will be truncated to the headers only immediately after frame type # detection. This will disable IP detection, etc, however it is likely # safer (and definitely more polite) if monitoring networks you do not own. # hidedata=true

Do we allow plugins to be used? This will load plugins from the system # and user plugin directiories when set to true (See the README for the default # plugin locations). allowplugins=true

See the README for full information on the new source format # ncsource=interface:options # for example: # ncsource=wlan0 # ncsource=wifi0:type=madwifi # ncsource=wlan0:name=intel,hop=false,channel=11

Comma-separated list of sources to enable. This is only needed if you defined # multiple sources and only want to enable some of them. By default, all defined # sources are enabled. # For example, if sources with name=prismsource and name=ciscosource are defined, # and you only want to enable those two: # enablesources=prismsource,ciscosource

Control which channels we like to spend more time on. By default, the list # of channels is pulled from the driver automatically. By setting preferred channels, # if they are present in the channel list, they'll be set with a timing delay so that # more time is spent on them. Since 1, 6, 11 are the common default channels, it makes # sense to spend more time monitoring them. # For finer control, see further down in the config for the channellist= directives. preferredchannels=1,6,11

How many channels per second do we hop? (1-10) channelvelocity=3

By setting the dwell time for channel hopping we override the channelvelocity # setting above and dwell on each channel for the given number of seconds. #channeldwell=10

Channels are defined as: # channellist=name:ch1,ch2,ch3 # or # channellist=name:range-start-end-width-offset,ch,range,ch,... # # Channels may be a numeric channel or a frequency #

Channels may specify an additional wait period. For common default channels, # an additional wait period can be useful. Wait periods delay for that number # of times per second - so a configuration hopping 10 times per second with a # channel of 6:3 would delay 3/10ths of a second on channel 6. # # Channel lists may have up to 256 channels and ranges (combined). For power # users scanning more than 256 channels with a single card, ranges must be used. # # Ranges are meant for "power users" who wish to define a very large number of # channels. A range may specify channels or frequencies, and will automatically # sort themselves to cover channels in a non-overlapping fashion. An example # range for the normal 802.11b/g spectrum would be: # # range-1-11-3-1 # # which indicates starting at 1, ending at 11, a channel width of 3 channels, # incrementing by one. A frequency based definition would be: # # range-2412-2462-22-5 # # since 11g channels are 22 mhz wide and 5 mhz apart. # # Ranges have the flaw that they cannot be shared between sources in a non-overlapping # way, so multiple sources using the same range may hop in lockstep with each other # and duplicate the coverage. # # channellist=demo:1:3,6:3,11:3,range-5000-6000-20-10

Default channel lists # These channel lists MUST BE PRESENT for Kismet to work properly. While it is # possible to change these, it is not recommended. These are used when the supported # channel list can not be found for the source; to force using these instead of # the detected supported channels, override with channellist= in the source defintion # # IN GENERAL, if you think you want to modify these, what you REALLY want to do is # copy them and use channellist= in the packet source. channellist=IEEE80211b:1:3,6:3,11:3,2,7,3,8,4,9,5,10 channellist=IEEE80211a:36,40,44,48,52,56,60,64,149,153,157,161,165 channellist=IEEE80211ab:1:3,6:3,11:3,2,7,3,8,4,9,5,10,36,40,44,48,52,56,60,64,149,153,157,161,165

Client/server listen config listen=tcp://127.0.0.1:2501 # People allowed to connect, comma seperated IP addresses or network/mask # blocks. Netmasks can be expressed as dotted quad (/255.255.255.0) or as # numbers (/24) allowedhosts=127.0.0.1 # Maximum number of concurrent GUI's maxclients=5 # Maximum backlog before we start throwing out or killing clients. The # bigger this number, the more memory and the more power it will use. maxbacklog=5000

Server + Drone config options. To have a Kismet server export live packets # as if it were a drone, uncomment these. # dronelisten=tcp://127.0.0.1:3501 # droneallowedhosts=127.0.0.1 # dronemaxclients=5 # droneringlen=65535

OUI file, expected format 00:11:22<tab>manufname # IEEE OUI file used to look up manufacturer info. We default to the # wireshark one since most people have that. ouifile=/etc/manuf ouifile=/usr/share/wireshark/wireshark/manuf ouifile=/usr/share/wireshark/manuf ouifile=/Applications/Wireshark.app/Contents/Resources/share/wireshark/manuf

Do we have a GPS? gps=true # Do we use a locally serial attached GPS, or use a gpsd server, or # use a fixed virtual gps? # (Pick only one) gpstype=gpsd # Host:port that GPSD is running on. This can be localhost OR remote! gpshost=localhost:2947

gpstype=serial # What serial device do we look for the GPS on? # gpsdevice=/dev/rfcomm0

gpstype=virtual # gpsposition=100,-50 # gpsaltitude=1234

Do we lock the mode? This overrides coordinates of lock "0", which will # generate some bad information until you get a GPS lock, but it will # fix problems with GPS units with broken NMEA that report lock 0 gpsmodelock=false # Do we try to reconnect if we lose our link to the GPS, or do we just # let it die and be disabled? gpsreconnect=true

Do we export packets over tun/tap virtual interfaces? tuntap_export=false # What virtual interface

do we use tuntap_device=kistap0

Packet filtering options: # filter_tracker - Packets filtered from the tracker are not processed or # recorded in any way. # filter_export - Controls what packets influence the exported CSV, network, # xml, gps, etc files. # All filtering options take arguments containing the type of address and # addresses to be filtered. Valid address types are 'ANY', 'BSSID', # 'SOURCE', and 'DEST'. Filtering can be inverted by the use of '!' before # the address. For example, # filter_tracker=ANY(!"00:00:DE:AD:BE:EF") # has the same effect as the previous mac_filter config file option. # filter_tracker=... # filter_dump=... # filter_export=... # filter_netclient=...

Alerts to be reported and the throttling rates. # alert=name,throttle/unit,burst # The throttle/unit describes the number of alerts of this type that are # sent per time unit. Valid time units are second, minute, hour, and day. # Burst describes the number of alerts sent before throttling takes place. # For example: # alert=FOO,10/min,5 # Would allow 5 alerts through before throttling is enabled, and will then # limit the number of alerts to 10 per minute. # A throttle rate of 0 disables throttling of the alert. # See the README for a list of alert types. alert=ADHOCCONFLICT,5/min,1/sec alert=AIRJACKSSID,5/min,1/sec alert=APSPOOF,10/min,1/sec alert=BCASTDISCON,5/min,2/sec alert=BSSTIMESTAMP,5/min,1/sec alert=CHANCHANGE,5/min,1/sec alert=CRYPTODROP,5/min,1/sec alert=DISASSOCTRAFFIC,10/min,1/sec alert=DEAUTHFLOOD,5/min,2/sec alert=DEAUTHCODEINVALID,5/min,1/sec alert=DISCONCODEINVALID,5/min,1/sec alert=DHCPNAMECHANGE,5/min,1/sec alert=DHCPOSCHANGE,5/min,1/sec alert=DHCPCLIENTID,5/min,1/sec alert=DHCPCONFLICT,10/min,1/sec alert=NETSTUMBLER,5/min,1/sec alert=LUCENTTEST,5/min,1/sec alert=LONGSSID,5/min,1/sec alert=MSFBCOMSSID,5/min,1/sec alert=MSFDLINKRATE,5/min,1/sec alert=MSFNETGEARBEACON,5/min,1/sec alert=NULLPROBERESP,5/min,1/sec #alert=PROBENOJOIN,5/min,1/sec

Controls behavior of the APSPOOF alert. SSID may be a literal match (ssid=) or # a regex (ssidregex=) if PCRE was available when kismet was built. The allowed # MAC list must be comma-separated and enclosed in quotes if there are multiple # MAC addresses allowed. MAC address masks are allowed. apspoof=Foo1:ssidregex="(?i:foobar)",validmacs=00:11:22:33:44:55 apspoof=Foo2:ssid="Foobar",validmacs="00:11:22:33:44:55,aa:bb:cc:dd:ee:ff"

Known WEP keys to decrypt, bssid,hexkey. This is only for networks where # the keys are already known, and it may impact throughput on slower hardware. # Multiple wepkey lines may be used for multiple BSSIDs. # wepkey=00:DE:AD:C0:DE:00,FEEDFACEDEADBEEF0102030405060708090 0

Is transmission of the keys to the client allowed? This may be a security # risk for some. If you disable this, you will not be able to query keys from # a client. allowkeytransmit=true

How often (in seconds) do we write all our data files (0 to disable) writeinterval=300

Do we use sound? # Not to be confused with GUI sound parameter, this controls wether or not the # server itself will play sound. Primarily for headless or automated systems. enablesound=false # Path to sound player soundbin=play

sound=newnet,true sound=newcryptnet,true sound=packet,true sound=gpslock,true sound=gpslost,true sound=alert,true

Does the server have speech? (Again, not to be confused with the GUI's speech) enablespeech=false # Binary used for speech (if not in path, full path must be specified) speechbin=flite # Specify raw or festival; Flite (and anything else that doesn't need formatting

around the string to speak) is 'raw', festival requires the string be wrapped in # SayText("...") speechtype=raw

How do we speak? Valid options: # speech Normal speech # nato NATO spellings (alpha, bravo, charlie) # spell Spell the letters out (aye, bee, sea) speechencoding=nato

speech=new,"New network detected s.s.i.d. %1 channel %2" speech=alert,"Alert %1" speech=gpslost,"G.P.S. signal lost" speech=gpslock,"G.P.S. signal O.K."

How many alerts do we backlog for new clients? Only change this if you have # a -very- low memory system and need those extra bytes, or if you have a high # memory system and a huge number of alert conditions. alertbacklog=50

File types to log, comma seperated. Built-in log file types: # alert Text file of alerts # gpsxml XML per-packet GPS log # nettxt Networks in text format # netxml Networks in XML format # pcapdump tcpdump/wireshark compatible pcap log file # string All strings seen (increases CPU load) logtypes=pcapdump,gpsxml,netxml,nettxt,alert

Format of the pcap dump (PPI or 80211) pcapdumpformat=ppi # pcapdumpformat=80211

Default log title logdefault=Kismet

logtemplate - Filename logging template. # This is, at first glance, really nasty and ugly, but you'll hardly ever # have to touch it so don't complain too much. # # %p is replaced by the logging prefix + '/' # %n is replaced by the logging instance name # %d is replaced by the starting date as Mon-DD-YYYY # %D is replaced by the current date as YYYYMMDD # %t is replaced by the starting time as HH-MM-SS # %i is replaced by the increment log in the case of multiple logs # %l is replaced by the log type (pcapdump, strings, etc) # %h is replaced by the home directory

logtemplate=%p%n-%D-%t-%i.%l

Where state info, etc, is stored. You shouldnt ever need to change this. # This is a directory. configdir=%h/.kismet/ </sxh>

Shortened

The following kismet.conf is a modifided and shortened version of the one above. It has been adapted to the most common use.

<sxh bash; title:Shortened kismet.conf;> # Kismet config file # # Optimized for the #WarBox

Version of Kismet config version=2009-newcore

Plugins # i.e. Ubertooth (https://www.kismetwireless.net/links.shtml) allowplugins=true

Name of server (Purely for organizational purposes) # If commented out, defaults to host name of system # servername=Kismet Server

Prefix of where we log (as used in the logtemplate later) # logprefix=/some/path/to/logs

Do we process the contents of data frames? If this is enabled, data # frames will be truncated to the headers only immediately after frame type # detection. This will disable IP detection, etc, however it is likely # safer (and definitely more polite) if monitoring networks you do not own. # hidedata=true

Do we allow plugins to be used? This will load plugins from the system # and user plugin directiories when set to true (See the README for the default # plugin locations). allowplugins=true

See the README for full information on the new source format # ncsource=interface:options # for example: # ncsource=wlan0 # ncsource=wifi0:type=madwifi # ncsource=wlan0:name=intel,hop=false,channel=11

Comma-separated list of sources to enable. This is only needed if you defined # multiple sources and only want to enable some of them. By default, all defined # sources are enabled. # For example, if sources with name=prismsource and name=ciscosource are defined, # and you only want to enable those two: # enablesources=prismsource,ciscosource

Control which channels we like to spend more time on. By default, the list # of channels is pulled from the driver automatically. By setting preferred channels, # if they are present in the channel list, they'll be set with a timing delay so that # more time is spent on them. Since 1, 6, 11 are the common default channels, it makes # sense to spend more time monitoring them. # For finer control, see further down in the config for the channellist= directives. preferredchannels=1,6,11

How many channels per second do we hop? (1-10) channelvelocity=3

By setting the dwell time for channel hopping we override the channelvelocity # setting above and dwell on each channel for the given number of seconds. #channeldwell=10

Channels are defined as: # channellist=name:ch1,ch2,ch3 # or # channellist=name:range-start-end-width-offset,ch,range,ch,... # # Channels may be a numeric channel or a frequency # # Channels may specify an additional wait period. For common default channels, # an additional wait period can be useful. Wait periods delay for that number # of times per second - so a configuration hopping 10 times per second with a # channel of 6:3 would delay 3/10ths of a second on channel 6. # # Channel lists may have up to 256 channels and ranges (combined). For power # users scanning more than 256 channels with a single card, ranges must be used. # # Ranges are meant for "power users" who wish to define a very large number of # channels. A range may specify channels or frequencies, and will automatically # sort themselves to cover channels in a non-overlapping fashion. An example # range for the normal 802.11b/g spectrum would be: # # range-1-11-3-1 # # which indicates starting at 1, ending at 11, a channel width of 3 channels, # incrementing by one. A frequency based definition would be: # # range-2412-2462-22-5 # # since 11g channels are 22 mhz wide and 5 mhz apart. # # Ranges have the flaw that they cannot be shared between sources in a non-overlapping # way, so multiple sources using the same range may hop in lockstep with each other # and duplicate the coverage. # # channellist=demo:1:3,6:3,11:3,range-5000-6000-20-10

# Default channel lists # These channel lists MUST BE PRESENT for Kismet to work properly. While it is # possible to change these, it is not recommended. These are used when the supported # channel list can not be found for the source; to force using these instead of # the detected supported channels, override with channellist= in the source defintion # # IN GENERAL, if you think you want to modify these, what you REALLY want to do is # copy them and use channellist= in the packet source. channellist=IEEE80211b:1:3,6:3,11:3,2,7,3,8,4,9,5,10
channellist=IEEE80211a:36,40,44,48,52,56,60,64,149,153,157,161,165
channellist=IEEE80211ab:1:3,6:3,11:3,2,7,3,8,4,9,5,10,36,40,44,48,52,56,60,64,149,153,157,161,165

Client/server listen config listen=tcp://127.0.0.1:2501 # People allowed to connect, comma seperated IP addresses or network/mask # blocks. Netmasks can be expressed as dotted quad (/255.255.255.0) or as # numbers (/24) allowedhosts=127.0.0.1 # Maximum number of concurrent GUI's maxclients=5 # Maximum backlog before we start throwing out or killing clients. The # bigger this number, the more memory and the more power it will use. maxbacklog=5000

Server + Drone config options. To have a Kismet server export live packets # as if it were a drone, uncomment these. # dronelisten=tcp://127.0.0.1:3501 # droneallowedhosts=127.0.0.1 # dronemaxclients=5 # droneringlen=65535

OUI file, expected format 00:11:22<tab>manufname # IEEE OUI file used to look up manufacturer info. We default to the # wireshark one since most people have that. ouifile=/etc/manuf ouifile=/usr/share/wireshark/wireshark/manuf ouifile=/usr/share/wireshark/manuf ouifile=/Applications/Wireshark.app/Contents/Resources/share/wireshark/manuf

Do we have a GPS? gps=true # Do we use a locally serial attached GPS, or use a gpsd server, or # use a fixed virtual gps? # (Pick only one) gpstype=gpsd # Host:port that GPSD is running on. This can be localhost OR remote! gpshost=localhost:2947

gpstype=serial # What serial device do we look for the GPS on? # gpsdevice=/dev/rfcomm0

gpstype=virtual # gpsposition=100,-50 # gpsaltitude=1234

Do we lock the mode? This overrides coordinates of lock "0", which will # generate some bad information until you get a GPS lock, but it will # fix problems with GPS units with broken NMEA that report lock 0 gpsmodelock=false # Do we try to reconnect if we lose our link to the GPS, or do we just # let it die and be disabled? gpsreconnect=true

Do we export packets over tun/tap virtual interfaces? tuntap_export=false # What virtual interface do we use tuntap_device=kistap0

Packet filtering options: # filter_tracker - Packets filtered from the tracker are not processed or # recorded in any way. # filter_export - Controls what packets influence the exported CSV, network, # xml, gps, etc files. # All filtering options take arguments containing the type of address and # addresses to be filtered. Valid address types are 'ANY', 'BSSID', # 'SOURCE', and 'DEST'. Filtering can be inverted by the use of '!' before # the address. For example, # filter_tracker=ANY(!"00:00:DE:AD:BE:EF") # has the same effect as the previous mac_filter config file option. # filter_tracker=... # filter_dump=... # filter_export=... # filter_netclient=...

Alerts to be reported and the throttling rates. # alert=name,throttle/unit,burst # The throttle/unit describes the number of alerts of this type that are # sent per time unit. Valid time units are second, minute, hour, and day. # Burst describes the number of alerts sent before throttling takes place. # For example: # alert=FOO,10/min,5 # Would allow 5 alerts through before throttling is enabled, and will then # limit the number of alerts to 10 per minute. # A throttle rate of 0 disables throttling of the alert. # See the README for a list of alert types. alert=ADHOCCONFLICT,5/min,1/sec alert=AIRJACKSSID,5/min,1/sec alert=APSPOOF,10/min,1/sec alert=BCASTDISCON,5/min,2/sec alert=BSSTIMESTAMP,5/min,1/sec alert=CHANCHANGE,5/min,1/sec alert=CRYPTODROP,5/min,1/sec alert=DISASSOCTRAFFIC,10/min,1/sec alert=DEAUTHFLOOD,5/min,2/sec alert=DEAUTHCODEINVALID,5/min,1/sec alert=DISCONCODEINVALID,5/min,1/sec alert=DHCPNAMECHANGE,5/min,1/sec alert=DHCPOSCHANGE,5/min,1/sec alert=DHCPCLIENTID,5/min,1/sec alert=DHCPCONFLICT,10/min,1/sec alert=NETSTUMBLER,5/min,1/sec alert=LUCENTTEST,5/min,1/sec alert=LONGSSID,5/min,1/sec

alert=MSFBCOMSSID,5/min,1/sec alert=MSFDLINKRATE,5/min,1/sec
alert=MSFNETGEARBEACON,5/min,1/sec alert=NULLPROBERESP,5/min,1/sec
#alert=PROBENOJOIN,5/min,1/sec

# Controls behavior of the APSPOOF alert. SSID may be a literal match (ssid=) or # a regex (ssidregex=) if PCRE was available when kismet was built. The allowed # MAC list must be comma-separated and enclosed in quotes if there are multiple # MAC addresses allowed. MAC address masks are allowed. apspoof=Foo1:ssidregex="(?i:foobar)",validmacs=00:11:22:33:44:55
apspoof=Foo2:ssid="Foobar",validmacs="00:11:22:33:44:55,aa:bb:cc:dd:ee:ff"

Known WEP keys to decrypt, bssid,hexkey. This is only for networks where # the keys are already known, and it may impact throughput on slower hardware. # Multiple wepkey lines may be used for multiple BSSIDs. # wepkey=00:DE:AD:C0:DE:00,FEEDFACEDEADBEEF01020304050607080900

Is transmission of the keys to the client allowed? This may be a security # risk for some. If you disable this, you will not be able to query keys from # a client. allowkeytransmit=true

How often (in seconds) do we write all our data files (0 to disable) writeinterval=300

Do we use sound? # Not to be confused with GUI sound parameter, this controls wether or not the # server itself will play sound. Primarily for headless or automated systems. enablesound=false # Path to sound player soundbin=play

sound=newnet,true sound=newcryptnet,true sound=packet,true sound=gpslock,true
sound=gpslost,true sound=alert,true

# Does the server have speech? (Again, not to be confused with the GUI's speech)
enablespeech=false # Binary used for speech (if not in path, full path must be specified)
speechbin=flite # Specify raw or festival; Flite (and anything else that doesn't need formatting # around the string to speak) is 'raw', festival requires the string be wrapped in # SayText("...")
speechtype=raw

How do we speak? Valid options: # speech Normal speech # nato NATO spellings (alpha, bravo, charlie) # spell Spell the letters out (aye, bee, sea) speechencoding=nato

speech=new,"New network detected s.s.i.d. %1 channel %2" speech=alert,"Alert %1"
speech=gpslost,"G.P.S. signal lost" speech=gpslock,"G.P.S. signal O.K."

How many alerts do we backlog for new clients? Only change this if you have # a -very- low memory system and need those extra bytes, or if you have a high # memory system and a huge number of alert conditions. alertbacklog=50

# File types to log, comma seperated. Built-in log file types: # alert Text file of alerts # gpsxml XML per-packet GPS log # nettxt Networks in text format # netxml Networks in XML format # pcapdump tcpdump/wireshark compatible pcap log file # string All strings seen (increases CPU load)
logtypes=pcapdump,gpsxml,netxml,nettxt,alert

Format of the pcap dump (PPI or 80211) pcapdumpformat=ppi # pcapdumpformat=80211

Logs

Default log title logdefault=kismet

Location of logs logprefix=/var/log/kismet/

Kismet Logs

logtemplate - Filename logging template. logtemplate=%p%n-%D-%t-%i.%l

Where state info, etc, is stored. You shouldn't ever need to change this. # This is a directory. configdir=%h/.kismet/ </sxh>

From:
https://wiki.c3l.lu/ - **Chaos Computer Club Lëtzebuerg**

Permanent link:
https://wiki.c3l.lu/doku.php?id=projects:security:warxing:kismet:kismet.conf

Last update: **2021/10/10 22:51**