

Kismet 2009-06-R1

Original von: **Mike Kershaw**

<http://www.kismetwireless.net>

Übersetzt von: Yanneck Schleese

<http://www.moutfish.de>

1. Was ist eigentlich Kismet +
2. Von einer früheren Version upgraden +
3. Schnellstart +
4. Suidroot & Sicherheit
5. Empfangsquellen einstellen
6. Vorsichtsmaßnahmen & Eigenarten für und von speziellen Treibern
7. Unterstützte Aufnahmequellen
8. Plugins
9. GPS
10. Einstellungsmöglichkeiten für die Log-Datei
11. Filtermöglichkeiten
12. Warnmeldungen & IDS
13. Server Konfiguration
14. Kismet UI - Benutzeroberfläche
15. Kismet Drone
16. Mit Kismet kommunizieren
17. Problemlösungen
18. FAQ – Häufig gestellte Fragen

1. Was ist eigentlich Kismet

Kismet kann 802.11 Funknetzwerke aufspüren, ist ein passiver Sniffer und zugleich ein Intrusion Detection System. Kismet arbeitet mit jeder W-LAN Karte, die den Monitor Mode unterstützt. Es ist in der Lage 802.11b, 802.11a, 802.11g und 802.11n Datenverkehr mit zuschneiden(wenn es das Gerät und die Treiber erlauben).

Kismet bietet zudem eine Plugin Schnittstelle, die es erlaubt zusätzliche, nicht 802.11 Protokolle zu entschlüsseln.

Kismet identifiziert Netzwerke durch passives Sammeln von Paketen und spürt auch Netzwerke auf die unsichtbar sind. Auch Funknetzwerke die keine Beacons senden, werden durch das Erkennen von Datenverkehr erkannt und angezeigt. Unvollständig

2a. Aktualisieren von den neusten Versionen

In der Version 2009-06-R1 gibt es grundlegende Änderungen in der Verwendung von multi-vap fähigen Karten(zum Beispiel, moderne im Linux-Kernel enthaltende Treiber). Sollte es möglich sein, erstellt Kismet einen neuen virtuellen Access Point und rekonfigurieren ihn, anstatt das bestehende Interface zu modifizieren. Will man das alte Verfahren aufrecht erhalten, kann man in der Konfigurationsdatei die Zeile `'forcevap=false'` angeben.

2b. Upgraden von Kismet-old Versionen

Dieser Release bringt bedeutende Änderungen, wie Kismet arbeitet und wie es konfiguriert wird. Während viele Aspekte gleich geblieben sind, haben sich zum Beispiel der Client und die Konfiguration von Paketquellen und Kanälen sehr stark geändert.

Um die neuen Features zu nutzen, musst du deine existierende Konfigurationsdatei auf den neusten Stand bringen. Dazu ersetzt du die alte Datei durch die mitgelieferte neue.

Die wichtigsten Änderungen:

- * Quellen werden nun anders definiert. Siehe „Empfangsquellen einstellen“.
- * Alle Benutzerinterface Einstellungen werden nun im Kismet Client verarbeitet und der Speicherort der Konfigurationsdatei ist „users home directory in ~/.kismet/kismet_ui.conf“.
- * Viele Situationen, die früher eine Absturzursache für Kismet waren sind behoben.
- * Neue Filtermöglichkeiten
- * Neue Alarmoptionen
- * Komplettes neues Benutzerinterface
- * Überarbeitetes Netzwerkprotokoll
- * Spürbar weniger CPU Auslastung bei einer großen Anzahl von Netzwerken
- * Plugins

Dadurch, dass dieser Release eigentlich alles vorangegangene über den Haufen wirft, eröffnet er neue Möglichkeiten für einfachere Upgrades und grundlegende Erweiterungen von Features.

3. Schnellstart

Es ist besser, wenn du die gesamte Anleitung liest, aber für die Ungeduldigen gibt es hier das Minimum an Informationen um Kismet zu benutzen:

- * Lade Kismet von <http://www.kismetwireless.net/download.shtml> herunter
- * Führe „./configure“ aus. Achte auf die Ausgabe. Falls Kismet nicht alle Header und Libraries finden kann, die es braucht wird die Funktionalität stark eingeschränkt sein. Solltest du Kismet selber kompilieren wollen, brauchst du die nötigen Entwicklerpakete und Header. Normalerweise enden die Pakete auf -dev oder -devel.
- * Mit „make“ kompilierst du Kismet.
- * Du kannst Kismet mit „make install“ oder mit „make suidinstall“ installieren.
LIES DIR „Suidroot & Sicherheit“ IN DER README DURCH, FALLS DU DIE ZWEITE VARIANTE WÄHLST, ANSONSTEN KANN DEIN SYSTEM UNSICHER WERDEN.
- * Solltest du Kismet als suid-root installiert haben, füge dein Benutzerkonto zu der Kismet-Gruppe hinzu.
- * Führe „kismet“ aus. Solltest du es nicht als suid-root installiert haben, musst du es als root starten. Es wird jedoch nicht empfohlen, weil diese Variante unsicherer ist, als der privsep Modus, in dem die Paketverarbeitung auch ohne Adminrechte möglich ist.
- * Wenn du gefragt wirst, ob du den Kismet-Server starten willst, dann wähle „yes“
- * Bei der Frage, welches capture interface du hinzufügen möchtest, wähle dein wireless interface. Kismet erkennt in der Regel den Gerätetyp und die unterstützten Kanäle. Falls nicht, musst du die Einstellungen manuell vornehmen, dazu aber später mehr.
- * Die Logs werden in dem Ordner gespeichert, aus dem Kismet gestartet wurde. Es sei denn, der Speicherort wurde durch den „logprefix“, die Konfigurationsdatei, oder die „--log-prefix“ Startoption geändert.
- * LIES DEN REST DER README. Kismet hat so viele Funktionen und eine Fülle an Konfigurationsmöglichkeiten, die dazu dienen das Beste aus Kismet herauszuholen. Es empfiehlt sich also wirklich den Rest der Dokumentation zu lesen.

3b. Windows Schnellstart

- * Zum aktuellen Zeitpunkt ist die upgedatete CACE Installation nicht verfügbar. Benutzer die also in den Genuss der newcore Funktionen kommen wollen, müssen also Kismet mit Hilfe von Cygwin selber kompilieren.

Benutzung des CACE Pakets:

- * Lade das hier <http://www.kismetwireless.net/download.shtml> verlinkte Win32/Cygwin Setup von CACE herunter.
 - * Führe die Installation aus.
 - * Starte Kismet
 - * Wähle deine AirPcap, oder Kismet Drone Quellen aus.
- *LIES DEN REST DER README

Selber kompilieren:

- * Lade das Cygwin-Setup-Tool herunter (<http://www.cygwin.org>)
- * Installiere Cygwin mit make, GCC, libncurses, libncurses-dev
- * Lade das Airpcap_Devpack von CACE herunter.
- * Packe das Airpcap_Devpack und Libpcap_Devpack in den Quellcodeordner von Kismet
- * Führe „./configure“ aus.
- * Kompiliere Kismet mit „make“
- * Installiere Kismet durch die Eingabe von „make install“.

ACHTUNG: KISMET WIRD ****NUR**** MIT CACE AIRPCAP GERÄTEN, GESPEICHERTEN PCAP DATEIN, -ODER- ENTFERNTEN KISMET DRONES ARBEITEN, DIE AUF EINER UNTERSTÜTZTEN PLATTFORM LAUFEN. MOMENTAN WIRD KEINE ANDERE HARDWARE UNTER WINDOWS UNTERSTÜTZT. WINDOWS TREIBER UNTERSTÜTZEN NICHT DIE VON KISMET BENÖTIGTE WIFI-MONITORING FUNKTION. ES GIBT AUCH KEINE MÖGLICHKEIT DAS ZU ÄNDERN.

3c. OSX/Darwin Schnellstart

- * Lade Kismet von <http://www.kismetwireless.net/download.shtml>
- * Führe „./configure“ aus. Achte auf die Ausgabe! Falls Kismet nicht alle Header und Libraries finden kann, die es braucht wird die Funktionalität stark eingeschränkt sein. Solltest du Kismet selber kompilieren wollen, brauchst du die nötigen Entwicklerpakete und Header. Es kann sein, dass du libpcap manuell installieren musst.

Die libpcap die in OSX enthalten ist, unterstützt kein PPI logging. Kismet wird also nicht in der Lage sein PPI richtig zu loggen (das bedeutet, es logt 802.11 Pakete ohne per-Paket Header)

„./configure“ wird OSX automatisch erkennen und eine Suidroot Installation vornehmen. Durch die Option „--with-suidgroup“ kann dies verhindert werden.

- * Mit „make“ kompilierst du Kismet.
- * Du kannst Kismet mit „make install“, oder mit „make suidinstall“ installieren.
LIES DIR „*Suidroot & Sicherheit*“ IN DER README DURCH, FALLS DU DIE ZWEITE VARIANTE WÄHLST, ANSONSTEN KANN DEIN SYSTEM UNSICHER WERDEN.
- * Solltest du Kismet als suid-root installiert haben, füge dein Benutzerkonto zu der „staff“-Gruppe hinzu, falls das noch nicht geschehen ist.
- * Führe „kismet“ aus. Solltest du es nicht als suid-root installiert haben, musst du es als root starten. Es wird jedoch nicht empfohlen, weil diese Variante unsicherer ist, als der privsep Modus, in dem die Paketverarbeitung auch ohne Adminrechte möglich ist.
- * Wenn du gefragt wirst, ob du den Kismet-Server starten willst, dann wähle „yes“
- * Bei der Frage, welches capture interface du hinzufügen möchtest, wähle dein wireless interface. Kismet erkennt in der Regel den Gerätetyp und die unterstützten Kanäle. Falls nicht, musst du die Einstellungen manuell vornehmen, dazu aber später mehr.
- * Bei vielen Macs ist das „en1“, öffne einfach ein Terminal und überprüfe die Ausgabe von „ifconfig -a“.
- * Die W-Lan Karte muss aktiviert sein, ansonsten funktioniert Kismet nicht.
- * Im Moment funktioniert Kismet NUR mit Airport W-LAN -Karten. USB-GERÄTE WERDEN NICHT UNTERSTÜTZT.
- * Die Logs werden in dem Ordner gespeichert, aus dem Kismet gestartet wurde. Es sei denn, der Speicherort wurde durch den „logprefix“, die Konfigurationsdatei, oder die „--log-prefix“ Startoption geändert.
- * LIES DEN REST DER README.

4. Suidroot & Sicherheit

Um die W-LAN Karte zu konfigurieren, den Monitor Mode zu aktivieren und Pakete aufzeichnen zu können, brauch Kismet Root-Rechte. Es gibt zwei Möglichkeiten dies zu erreichen: Entweder startet man Kismet als Root, oder man installiert es so, dass Kontrollkomponenten automatisch als Root starten.

Startet man Kismet als Root, bedeutet das, dass Kismet die ganze Zeit als Programm mit Root-Zugriff ausgeführt wird. In der Theorie stellt das keine zusätzliche Gefährdung dar. Sollte es jedoch

unerwartete Mängel in Kismets Paketverarbeitungscode geben, könnte es möglich sein bösartigen Code ins System zu schleusen, der dann mit Root-Rechten ausgeführt wird. Außerdem werden fremde Plugins als Root ausgeführt, was ebenfalls unsicher sein kann.

Die `suid-root` Installation erstellt eine Binary mit eingeschränkter Funktionalität (`kismet_capture`) die nur durch Mitglieder der Gruppe „`kismet`“ gestartet werden kann. Kismet ist dadurch in der Lage die Karten zu konfigurieren und die Kanäle zu wechseln. Das Encodieren der Pakete erfordert keine besonderen Rechte. Durch diese Maßnahme wird die Angriffsfläche auf Kismet stark verringert.

Die Distributionen sind stark bemüht, diese Methode zu verbreiten, weil durch die normale Gruppenverwaltung geregelt werden kann, was welcher Benutzer darf. Kismet wird es also in diesem Fall gestattet, den Zustand des W-LAN Interfaces zu ändern.

Embedded Systems haben für gewöhnlich weniger Speicherplatz und Arbeitsspeicher. Außerdem ist meist keine Benutzer/Root trennung vorgesehen. Bei eingebetteten Systemen ist es eventuell erforderlich Kismet ohne die `kismet_capture` Binary zu installieren und es nur im Root-Modus auszuführen. Jedoch bleibt das weiter oben im Text erwähnte Risiko bestehen.

Unter keinen Umständen sollte die `kismet_server` Binary selbst als `suidroot` installiert werden, da dadurch jeder Sicherheitscheck umgangen wird.

5. Empfangsquellen

Alle Pakete die Kismet verarbeitet kommen von einer Paketquelle. Diese Paketquellen sind in der Regel Netzwerkkarten auf dem lokalen System. Es können aber auch zuvor aufgezeichnete Dateien, oder externe Systeme sein, auf denen eine Kismet Drone läuft.

In den meisten Fällen erkennt Kismet den richtigen Treiber und die unterstützten Kanäle, wenn man nur das Interface der Netzwerkkarte angibt. Obwohl dies für die meisten Benutzer genügen wird, gibt es viele Einstellungsmöglichkeiten für die Aufnahmequellen.

Kismet empfängt Pakete über die 802.11 Schicht. Dazu wird der Modus des Interfaces geändert, was eine Benutzung für den normalen Betrieb unmöglich macht. Es ist also in der Regel nicht möglich, weiter mit einem W-LAN verbunden zu sein, während Kismet auf das Gerät zugreift.

Aufnahmequellen können über das Kismet UI hinzugefügt werden. Dazu geht man auf „Add source“ und fügt die Interfaces hinzu. Hat man mehr als ein Interface, werden diese durch ein Komma getrennt.

Es gibt jedoch auch die Möglichkeit, die `kismet.conf` zu benutzen. Dazu verwendet man die „`ncsource=`“ Option. In etwa so: „`ncsource=wlan0:option1=foo,option2=bar`“.

Einstellungsmöglichkeiten für die Quelle:

<code>name=foo</code>	Eigener Name für die Quelle (sollte keiner angegeben werden, wird der Name des Aufnahmeinterface verwendet). Diese Option ist ohne weitere Bedeutung und dient lediglich zur leichteren Erkennung für den Benutzer.
<code>type=foo</code>	Quellen die nicht automatisch erkannt werden müssen hier definiert werden. Das ist zwar äußerst selten der Fall, aber es ist möglich. Eine Erklärung zu den verschiedenen Typen folgt weiter unten im Text.
<code>uuid=foo</code>	Benutzer die eine einmalige Kennung für ihre Quelle wünschen können dies hier einstellen. Für die meisten Anwender trifft dies nicht zu. Ein Kennung wird nach diesem Muster eingegeben: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXXX
<code>hop=true false</code>	Deaktiviert „channel hopping“ für diese Quelle. Channelhopping ist erforderlich um ein breites Spektrum an Kanälen abzudecken.
<code>Velocity=#</code>	Geschwindigkeit der Kanalwechsel (Anzahl der Kanäle pro Sekunde), Kismet kann zwischen 1-10 Kanäle pro Sekunde wechseln.
<code>Dwell=#</code>	Dwell ist die Zeit, die Kismet auf jedem Kanal wartet bevor es weiterspringt. Falls das Hopping eingeschaltet ist und eine Zeit angegeben wird, dann wird Kismet N Sekunden in diesem Kanal warten bevor es weiterspringt und nicht mehr N Kanäle

	pro Sekunde wechseln.
channellist=name	Benutze einer alternative Kanalliste und nicht die automatisch erkannte. Dazu muss die Liste angegeben werden.
split=true false	Falls mehrere Quellen die selbe Kanalliste verwenden, (entweder automatisch erkannte, oder durch channellist= option festgelegte) teilt Kismet diese, so dass sich die Kanäle sich nicht überschneiden und zwei Interfaces zu der selben Zeit im gleichen Kanal sind. Dieses Verhalten kann jedoch durch das setzen von „false“ außer Kraft gesetzt werden.
retry=true false	Im Falle eines Fehlers versucht Kismet die Quelle wieder zu öffnen. Wenn man dies nicht möchte, setzt man die Option auf „false“.
vap=interface	Erstellt ein weiteres Interface zum aufnehmen, anstatt das existierende zu benutzen. Das macht eigentlich nur Sinn für Treiber die das mac80211 Interface unter Linux nutzen. Benutzer die Kismet+Managed, oder Kismet+Injektion verwenden wollen, sollten die vap Option nutzen.
forcevap=t f	True/False. Erzwingt die Erstellung eines Monitor-Mode VAP, falls dies möglich ist. (alle auf Linux mac80211 basierenden Treiber unterstützen dies). Standardeinstellung ist „true“, ein VAP wird in seinem Namen ein 'mon', ie 'wlan0mon', 'wlan1mon' enthalten und alle Aufzeichnungen laufen über diesen VAP. Dieses Verhalten kann mit 'forcevap=false' verhindert werden.
wpa_scan=time	Falls ein mac80211 VAP verwendet wird, kann Kismet wpa_supplicant auf einem Managed Interface aktiviert werden um Hardware unterstützte Scans zu verwenden. Es ist also möglich auf anderen Kanälen zu horchen, ohne die Verbindung zum Netzwerk zu verlieren . Ein Vorschlag für die Zeit sind 15 Sekunden.
validatefcs=t f	True/False. Kismet bemüht sich normalerweise nicht die FCS Checksumme von ankommendem Packeten zu überprüfen, weil die meisten Treiber nur die gültig Frames anzeigen. Paketquellen die ungültige Frames anzeigen, werden automatisch diese Option aktivieren. Wenn das Gerät manuell konfiguriert wurde, sollte man dieses Feature aktivieren um zu verhindern, dass Kismet ungültige Pakete verarbeitet.
fcs=true false	Erzwingt das Bearbeiten von FCS Bytes einer Paketquelle. Standard ist "false", was eine ursprünglich FCS Bearbeitung bedeutet. Quellen die „per-packet headers“ beinhalten, wie zum Beispiel Radioapparate oder PPI werden diese Einstellungen ignorieren, da der FCS im Funk-Header encodiert wird. Bei Quellen wie zum Beispiel vorher aufgezeichnete Dateien, oder roh 802.11 Aufnahmedateien ohne Header sollte diese Funktion aktiviert werden.
fcsfail=true	Zwingt einen mac80211 VAP Pakete mit falscher FCS (Paket Checksumme) zu melden. Diese Option funktioniert nur unter Linux und bei Verwendung von mac80211 Treibern. Die Einstellung MUSS hinter 'vap=' stehen, oder sie wird ignoriert. Wenn 'fcsfail' eingeschaltet wird, dann wird 'validatefcs' automatisch aktiviert. Die 'fcsfail'-Option sollte nur eingeschaltet werden, wenn man man zu einer PPI logt. Beim auslesen von normalen PCAP-Dateien wird die FCS nicht erhalten und die Ausgabe wird unlesbar. WARNUNG: Bei manchen Treiberversionen führt das aktivieren zu OOPS Warnungen des Kernels und das Interface reagiert nicht mehr, wenn die Aufzeichnung gestoppt wird. Diese Option ist nur für Experten und sollte am besten in Ruhe gelassen werden.
plcpfail=true	Zwingt einen mac80211 VAP Pakete die nicht den PLCP Check bestehen zu melden. (falls das auf dem Interface möglich ist). Es herrschen die selben Bedingungen und Warnungen wie bei 'fcsfail'. Diese Option ist nur für Experten und sollte am besten in Ruhe gelassen werden.

Beispiel Einstellungen (Diese werden in die Konfigurationsdatei geschrieben, sie funktionieren aber auch als Kommandozeilen-Optionen, z.B. "-c wlan0"):

Aufnahme auf wlan0, Kanal 6, keine Sprünge zwischen den Kanälen
ncsource=wlan0:hop=false,channel=6

Aufnahme nur auf wlan0, 802.11b Kanälen starten, auch dann wenn 5GHz-Kanäle unterstützt werden.

```
ncsource=wlan0:channellist=IEEE80211b
```

Erstellt einen VAP auf wlan0 nennt ihn wlan0mon und nutzt wpa_supplicant um andere Kanäle zu sehen, aber dennoch mit dem Netzwerk verbunden zu bleiben.

```
ncsource=wlan0:vap=wlan0mon,hop=false,wpa_scan=15
```

Lesen von einer zuvor aufgezeichneten Datei:

```
ncsource=/home/foo/old.pcap
```

Aufnahme durch das erste Aircap-Gerät von Windows

```
ncsource=airpcap
```

Aufnahme durch die Benutzung einer Drone

```
ncsource=drone:host=10.10.100.2,port=2502
```

Kanallisten:

Kanallisten bestimmen die Kanäle und die Abstände in den Kismet springt. Das ist dann der Fall, wenn Kismet die Kanäle automatisch erkennt, oder wenn der Benutzer wünscht sie aus einem bestimmten Grund zu überschreiben.

Normalerweise werden Kanäle als IEEE Kanäle angesprochen (11, 36, etc) oder als Frequenzen (2401, 5200), jedoch kann es sein dass manche Plattformen oder Treiber nicht die IEEE Standard Bandbreite einhalten.

```
channellist=name:channel,channel,channel
```

Es ist auch möglich, Kanälen mehr Zeit zuzusprechen. In diesem Beispiel wird dreimal so viel Zeit in Kanal 1, 6 und 11 verbracht.

```
channellist=foo:1:3,6:3,11:3,2,3,4,5,6,7,8,9,10
```

Bis zu 256 Kanäle können in einer Kanalliste angegeben werden. Für weitere Kanäle muss die Bandbreite angegeben werden.

Bereichen können entweder durch Kanäle oder durch Frequenzen definiert werden.

```
channellist=name:range-[Start]-[Ende]-[Überlappung]-[Wiederholung]
```

Kanäle zwischen Start und Ende, mit Wiederholung. Kismet wird nicht zwischen Kanälen springen, die sich direkt überschneiden.

```
channellist=foo:range-1-11-3-1
```

Es ist ebenfalls möglich Bereiche durch Frequenzen zu definieren (802.11 2.4GHz Kanäle sind ~20MHz breit; theoretisch 22 aber 20 reicht auch, und 5 MHz getrennt).

```
channellist=foo:range-2412-2462-20-5
```

Von-Bis Bereiche werden nicht unter den Karten aufgeteilt. Es kann also passieren, dass sich zwei Quellen zu der selben Zeit auf dem selben Kanal befinden. - in anderen Worten, Von-Bis Angaben werden wie ein einzelner Kanal behandelt.

Kanäle können hier entweder von-bis, oder getrennt durch Kommas angegeben werden.

```
channellist=foo:range-1-11-3-1,36,52
```

Übersetzungsfehler können nicht ausgeschlossen werden. Ich übernehme keine Garantie dafür, dass die Übersetzung zu 100% korrekt ist. Im Moment ist sie unvollständig. Kommentare sind erwünscht und Kritik wird berücksichtigt. [Status=2673/9455 Wörtern]