Bluetooth Sniffing with BloosCloos

Andrew White, Michael Stegeman, Jef Harkay, Ben Anglin
{whitea1, mastege, jef3, jama1}@umbc.edu

20 May 2011

Abstract

Packet sniffing on Bluetooth networks has proven to be a difficult task, and due to this fact, companies typically charge large sums of money for equipment to help analyze and secure a Bluetooth network. The motivation behind this paper is to demonstrate that Bluetooth sniffing and analysis can be done without exorbitant cost and with similar ease to typical Wi-Fi analysis.

Our work is not without precedence. At ShmooCon 2011, Michael Ossmann [**?**] demonstrated that it is possible to create custom Bluetooth hardware and interface it with open source software to perform basic Bluetooth traffic analysis. Though Mr. Ossmann's approach was commendable, its reliance on custom hardware creates unnecessary complications in implementation. To remedy this, BloosCloos, in part, combines Mr. Ossmann's research with research conducted by Max Moser [**?**], who has demonstrated that commodity hardware with a Cambridge Silicon Radio (CSR) chipset can be used to sniff Bluetooth traffic when flashed with modified firmware.

BloosCloos is a Bluetooth sniffing package that contains a Bluetooth dongle with monitor mode capabilities and custom libpcap and Wireshark builds providing techniques for packet sniffing. The dongle included in the BloosCloos package is an ordinary "off the shelf" Bluetooth dongle running modified firmware. The tools included in the package help capture and analyze Bluetooth network traffic. Since the device requires very little alteration, and the tools are made using free, open source software, the overall cost of the package is minimal for a security researcher. Customers that purchase this package are able to analyze and secure their Bluetooth networks for approximately $15.

Keywords

Bluetooth, BloosCloos, sniffing, Wireshark, dongle, scanning, flashing, firmware, libpcap

1 Introduction

Sniffing network traffic is not a new problem, but tools to do so are a necessary component of network protocol development and security research. These tools are used every day by security professionals to secure their networks. Historically, Bluetooth was designed as a proprietary protocol and was developed to prevent access to its lower level implementations in an effort to prevent disclosure of its design. Now that the protocol has become an open standard, this difficulty should no longer be an issue but still discourages developers. This has led to a lull in the development of tools to monitor Bluetooth network traffic.

Sniffing a Bluetooth network requires much more effort than Wi-Fi sniffing due to the frequency-hopping spread spectrum (FHSS) technique used by Bluetooth devices. The FHSS provides a challenge because there are 79 channels that the devices can jump around on. If a researcher has 79 Bluetooth sniffers all on separate channels, then this is not an issue. However, this would require a large sum of money, and the 79 sniffers all must share information with each other, which is both difficult and confusing to implement.

In order to follow the frequency hopping pattern, a user must know the Bluetooth piconet master's address (BD_ADDR). Although sniffing Bluetooth networks is difficult, the user connected should not feel completely secure because there are methods of discovering the BD_ADDR. Regardless, Bluetooth users should be interested in and aware of their network's security.

The knowledge of the master Bluetooth device's address is a hard problem because the addresses do not normally show up in the packets exchanged. This is not a problem if the Bluetooth master is in discoverable mode because the Linux "hcitool scan" command will reveal the address. However, if the master is in hidden mode, the user must find a more creative way of determining the addresses.

One way to discover the master's address is to observe the frequency hop synchronization frames in the connection between the master and slave devices. Another possibility is to observe when the master and slave devices switch roles, revealing the slave's address. Both of these methods only provide the master's address, but that is all that is needed to carry out sniffing on the desired piconet [**?**].

This paper presents a package that can gather information from a Bluetooth network with cheap, commodity hardware and free, open source software. The first section will give an overview of relevant work, followed by the methods section, which will describe how we went about accomplishing this paper's objective. The next two sections will show and discuss the results of our experiment, and the fifth section will be our conclusions.

2 Previous Work

At ShmooCon 2011, Michael Ossmann presented a piece of hardware he created called *Ubertooth* [**?**]. This device is a custom-built Bluetooth sniffer that costs around $60 to implement. In addition to creating this device, a plugin for Kismet was developed to perform the actual sniffing. Unlike Ubertooth, BloosCloos uses commodity hardware as the creation of custom hardware is prohibitive to many otherwise excellent security researchers. Rather than using Kismet, we decided to use Wireshark and libpcap for packet sniffing due to their popularity, intuitive GUI and ease of use.

Max Moser wrote a paper in 2007 entitled *Busting the Bluetooth Myth - Getting RAW Access* [**?**]. This paper describes several Cambridge Silicon Radio (CSR) Bluetooth chipsets that can be flashed with custom firmware, giving them sniffing capabilities. We used one of the chipsets and the firmware flashing techniques described for our project. Mr. Moser's project used commercial

software to perform the sniffing, but our project is significantly different because we perform Bluetooth sniffing with a version of the open source software Wireshark that we modified to have these capabilities.

In 2001, Markus Jakobsson presented a paper entitled *Security Weaknesses in Bluetooth* [**?**]. As the name suggests, the paper detailed several vulnerabilities in what was then the current Bluetooth standard (1.0B) along with possible exploits. Among the cited exploits were attacks against PIN length, methods to glean information concerning a user's location, and spoofing and intercepting real-time communication.

For the last exploit, it was noted that an attacker would need to be able to defeat the channel-hopping security measure used by the Bluetooth protocol. Towards this end, Mr. Jakobsson suggested the creation of devices which would simply scan all 79 channels used by Bluetooth, thus allowing an attacker to sniff communications regardless of the channel. However, Mr. Jakobsson left this, and other exploits, as theory. Our project took his idea a step further and implemented the sniffing capability necessary to provide real-time traffic analysis. Our research is not the first such implementation of Bluetooth sniffing hardware. However, with our work, users now do not need lots of money or low level knowledge of hardware to do this research. Our project has taken what has thus far been an activity reserved for a privileged few and enabled it for the masses.

Gregory Lam, et al.'s 'Unit Key Stealing' technique [**?**] takes advantage of the shared link keys between the two devices during authentication and the fact that the PIN shared between the two devices is usually 4 digits that are generated by the device itself. Typically, a device will reuse this private information on every connection it initiates, so any device which legitimately connects to it will have this 'private' information. The initial handshake which sets up the encryption channel between the two devices is unencrypted and can be fully monitored. The only information another device needs to impersonate a device in this communication is the frequency-hopping scheme. BloosCloos complements their work because it utilizes commodity tools which allows researchers to demonstrate some of the techniques described. Future research on this topic can take advantage of our tool to show that these attacks are in fact major problems with the current implementations of the Bluetooth standard.

3 Methods

3.1 Obtaining a Bluetooth Dongle

In order to monitor Bluetooth traffic, we needed a Bluetooth dongle providing RAW capabilities. To achieve this, we had to find a dongle that could be flashed with custom firmware, but we quickly learned that there were very few chipsets that are rewritable. Luckily, we found the BlueCore4-EXT chipset, which is part of the Cambridge Silicon Radio (CSR) family.

After finding a flashable chipset, we had to locate a Bluetooth dongle that used this chipset. The cheapest and easiest one to purchase was the D-Link DBT-120 Revision C1 dongle. We bought two used DBT-120 dongles on eBay for $15 each.

3.2 Flashing the Bluetooth Dongle

For the flashing process, we had to acquire an older firmware version that supported RAW access. This was a little tricky because the vendor stopped providing free firmware updates. After scouring the Internet for an acceptable firmware version, we came across the airsnifferdev46bc4.dfu firmware. We then flashed this firmware to the device using 'dfutool'. Finally, we verified that we had RAW access using the 'hcitool' and 'hciconfig' tools.

NOTE: dfutool, hcitool, and hciconfig are all part of the BlueZ Bluetooth stack for Linux.

3.3 Bluetooth Sniffer Interface

The next step was to create an easy to use interface that allows a user to sniff Bluetooth traffic. For this, we decided to use Wireshark, a packet analysis tool, and libpcap, the underlying packet capture library. This open source software tends to be popular with network security researchers, so it provides BloosCloos users a familiar interface to work with.

3.3.1 Wireshark and libpcap

The first software problem we addressed was getting our Bluetooth devices to be recognized by Wireshark. After reading through the Wireshark and libpcap source code, we determined that libpcap already had the ability to detect Bluetooth interfaces. However, we did not see any of our Bluetooth hardware listed in the program. Through further research, we discovered that none of the Linux distributions we were using provided libpcap libraries with this Bluetooth detection enabled. As such, we recompiled libpcap, enabling Bluetooth detection, and linked Wireshark against this new library. Wireshark then displayed our Bluetooth devices, and we could perform packet captures using regular Bluetooth dongles, as long as they were connected to a piconet.

We also found that Wireshark has the ability to manually detect capture interfaces, should the installed libpcap not have the ability to do so. To ensure that Bluetooth devices were still detected in this case, we ported the Bluetooth interface detection from libpcap to Wireshark. After this initial capture interface detection, Wireshark handles devices the same way as if libpcap detected them.

3.3.2 Frontline

We then began to deal with the RAW access capabilities. For this, we turned to the Frontline program, which is an open source project that allows users with CSR-based Bluetooth dongles to sniff on a Bluetooth network, given the piconet master's MAC address. When this MAC address is provided, Frontline automatically finds the master's clock and syncs the RAW device's clock with that of the piconet master's. After this happens, we can sniff any traffic on the network.

Because this code already performs well, we ported a large portion of the Frontline code into libpcap. When a Wireshark user selects a Bluetooth interface with RAW support, an interface is displayed in Wireshark that asks the user for the master's MAC address (BD_ADDR). When the user enters the BD_ADDR, our code attempts to find the device with the specified address and sync the clocks. With the clocks synced, it is possible for the RAW device to hop along with the rest of the piconet allowing Wireshark to see all of the traffic sent on this piconet through the sniffer.

3.3.3 GTK

Next, we looked into adding the new functionality into the existing user interface. We originally intended to add this through a plug-in, but due to time constraints we decided to just fork the Wireshark code and add our modifications into the existing GTK GUI.

To design this new interface we first looked into using the automated Glade Interface Designer to allow us to quickly prototype and create this code. Unfortunately, this ended up not working since the newest versions of the software changed how it generated the GTK code. Wireshark uses an older GTK model which is created by directly invoking GTK commands to generate the various parts of the GUI. The newer model relies on parsing XML documents to create the interface, which would have required a complete rewrite of the entire Wireshark interface to use Glade effectively. As such, we decided to manually generate the GTK code which created the interface we desired.

After testing with a hard coded master MAC address, we began to add the capability to dynamically change the MAC address. We added a section in the interface options dialog where a user could enter the desired MAC address. Similarly, we added a toolbar button (Figure 1) that allows the user to change this MAC address during a capture, which will restart the capture. When the user starts a new capture on the RAW interface, the provided MAC address is passed to a new libpcap function to set the master MAC before the capture begins.

Figure 1: Bluetooth MAC address toolbar button (in the orange square).

Figure 2: Bluetooth MAC address box after clicking the button.

We also added a global preference into Wireshark (Figure 3) that allows the user to set a default master MAC address. This preference is persistent across restarts of Wireshark, and will automatically be used if the user does not enter a new MAC address via the toolbar button or interface options. To add this preference, we traced similar Wireshark preferences and added new function calls where necessary.

Figure 3: Black rectangle shows the new master MAC address box in the BloosCloos Wireshark options dialog box.

3.4 Command Line

The Wireshark source code also builds two other tools, namely 'dumpcap' and 'tshark'. These tools can be used from the command line to perform packet captures if a graphical environment is not available. Wireshark and Tshark both make calls to Dumpcap to perform their packet captures. As such, it was necessary to add an extra command line option to both programs that allows the user to set the desired MAC address. This MAC address can now be specified by passing the '-U' option to either program. For instance, to run Tshark on a RAW Bluetooth interface, a user could run 'tshark -U 01:23:45:67:89:ab -i hci0', where hci0 is the desired interface. Dumpcap can be run in a similar manner.

3.5 DAVE_CONFIG

We also created a script which configures and builds both libpcap and Wireshark with our modifications, called DAVE_CONFIG. It also installs flex, bison and libbluetooth-dev if they are not already installed on the system. For ease of installation, the user just has to run the DAVE_CONFIG command. The script can optionally try to flash firmware to a specified Bluetooth dongle, if desired. There are also instructions on how to properly backup and flash the Bluetooth dongle's firmware, as well as building libpcap and Wireshark, should DAVE_CONFIG fail.

3.6 Testing

Our packet capturing was completed in a relatively quiet air space, though not devoid of noise. We captured packets on our own piconet in a restaurant during lunch hours, so other Bluetooth enabled devices were present in the vicinity. An additional form of noise on the 2.4 GHz airspace was that of the restaurant's wifi network. Even with this noise, we managed to capture Bluetooth traffic from our piconet.

4 Results

After enabling Bluetooth support in libpcap, we were able to capture traffic using a standard Bluetooth dongle, but only if that dongle was connected to a piconet. This is similar to capturing on a Wi-Fi interface in non-promiscuous mode. This is useful for debugging existing Bluetooth sessions but does not give a user the desired sniffing capabilities. Listing 1 shows a few packets from a capture and Figure 4 shows the actual Wireshark capture obtained during a file transfer between the devices.

No.	Time	Protocol	Info
42	3.822042	SDP	Rcvd SDP_ServiceSearchAttributeResponse

```
43      3.825446    L2CAP       Sent Connection Request
44      3.828057    HCI_EVT     Number of Completed Packets
45      3.862040    L2CAP       Rcvd Connection Response
46      3.862052    L2CAP       Sent Configure Request
```

Listing 1: File transfer capture on normal Bluetooth dongle.

Figure 4: BloosCloos Wireshark displaying the results of a normal capture.

We were able to test if an individual Bluetooth interface had RAW capabilities in Wireshark. This gave us the ability to apply special policies to RAW devices that will not be activated for standard Bluetooth devices. This includes performing the necessary clock synchronization Frontline requires to allow the device to hop along with the piconet. Testing if a Bluetooth device has RAW access can be done as shown in Listing 2, using BlueZ library functionality.

```
static int bt_is_raw(int dev_id)
{
    struct hci_dev_info di;
    if (hci_devinfo(dev_id, &di) < 0)
        return 0;

    return hci_test_bit(HCI_RAW, &di.flags);
}
```

Listing 2: Detecting RAW Bluetooth device.

Once the Frontline code was integrated into libpcap, we were able to test with the RAW dongle. The BD_ADDR of the development machine's internal Bluetooth radio was hard-coded into libpcap for early testing. We paired the computer with a cell phone such that the development machine was designated as the piconet master. When a capture was started in Wireshark on the RAW device, it synced that device's clock with the clock of the internal Bluetooth adapter and started capturing packets on that piconet. However, all packets were displayed in Wireshark as either HCI_ACL or HCI_EVT packets, rather than the individual protocols, such as L2CAP or SDP. For instance, the following packets (shown in Listing 3 and Figure 5) were captured during a similar file transfer between the devices.

No.	Time	Protocol	Info
9185	22.711739	HCI_ACL	
9186	22.797738	HCI_EVT	Vendor-Specific

Listing 3: File transfer capture on RAW Bluetooth dongle.

Figure 5: BloosCloos Wireshark displaying the results of a RAW capture.

Included with this paper are the modified versions of Wireshark and libpcap that we created, in Appendices A and B, respectively. Additionally the code for DAVE_CONFIG is in Appendix C. These sections should be consulted for further details pertaining to our code implementation.

In total, we changed and added a total of 199 lines of code in libpcap and 537 in Wireshark. The DAVE_CONFIG script is 335 lines of code.

5 Discussion

The biggest limitation of BloosCloos is that it requires the BD_ADDR of the piconet master, which is obviously not ideal. This limitation can be easily overcome by either scanning for the master's address or reading the Bluetooth addresses off of the physical devices. There are other methods of finding Bluetooth devices, such as sequentially pinging MAC addresses, but this is not something that could be implemented in Wireshark. Even with this drawback, the capabilities that have been implemented in BloosCloos may enable researchers to develop better ways of finding BD_ADDRs, which could eventually be added into BloosCloos or implemented as a separate project.

One of the drawbacks to our project is that it requires a device with a very specific radio. The device we used has already been discontinued by the manufacturers. While the exact radio chipset we used is not necessarily required, we do not know of any other radio chipsets that give the same ability to flash custom firmware, allowing RAW access. Discovering another piece of hardware compatible with the modifications we used would be a very good place to continue work on this topic.

We believe that we completed our testing in a realistic environment even though we did not test it under the most adverse conditions. We felt that our testing was adequate for security analysis and that a slightly noisy environment is good enough to show results. Additional research under more adverse and more clear conditions may be good places to continue research.

Our results show that we managed to integrate basic RAW Bluetooth monitoring capabilities into Wireshark, but more work is necessary to create an ideal platform for Bluetooth security analysis. An ideal Bluetooth security analysis platform would capture and decode all packets and allow for the visualization of a network and its traffic.

Future work continuing this research should begin by updating Wireshark to reassemble the packets captured in RAW mode such that their protocols are identified as with normal Bluetooth devices. Once the packets are successfully reassembled, it should be possible to implement Bluetooth packet decryption in Wireshark in a similar manner as the built-in 802.11 WEP packet decryption.

We have found that our changes to Wireshark are finicky. Sometimes it works without a problem, but other times it does not capture any packets. When we run the clock syncing portion of the Frontline code while also running Wireshark, we seem to regularly get results. This leads us to believe that we have a design flaw in our code. Frontline resyncs the clocks

every 30 seconds, whereas BloosCloos syncs the clocks only on start. It may be necessary to add threading to libpcap to accomplish this constant resyncing feature.

While modifying libpcap, we ran into an issue with a Linux kernel upgrade that broke the Bluetooth interface detection. This forced us to rework the code such that it was more compliant with the BlueZ Bluetooth stack implementation, rather than the very low level methods that libpcap usually implements. After modifying the code, the interface detection was all back to normal in both Wireshark and libpcap.

6 Conclusions

With this project, we attempted to perform Bluetooth network sniffing in a simple, cost-effective manner. Previous projects have been able to sniff Bluetooth networks, but have either required large budgets or circuit design experience, as they used either expensive or custom-built hardware. BloosCloos instead uses "off-the-shelf" hardware and open source software, allowing a wide variety of researchers the ability to analyze Bluetooth network traffic on a budget.

Retrospectively, it appears that we should have limited our project's scope a little more than we did. While we can sniff any connection for which we have the proper BD_ADDR, the idea that we can scan all channels looking for active connections seems less practical, as it requires significantly more advanced hardware. As such, this would be a good place to continue research.

This limitation aside, we developed the Wireshark and libpcap Bluetooth detection functionality beyond their previous capabilities. They also now have the ability to handle RAW devices differently, specifically allowing RAW devices to sync to a given piconet's frequency hopping pattern. With this functionality, we have shown that it is possible to use commodity hardware to sniff Bluetooth connections inside of Wireshark. As this was the primary goal of our project, we believe that we have achieved overall success.

7 Bibliography

Appendix A: Wireshark Source Code Modifications

The following is a patch to Wireshark, version 1.4.4.

```
diff -rupN -x .gitignore -x configure -x svn_version.xml -x 'grammar.[ch]' -x '
    dtd_grammar.[ch]' -x '*wslua.[ch]' -x faq.txt -x Info.plist -x 'mate_grammar.[ch]' -
    x ps.c -x svnversion.h -x idl2wrs.sh wireshark-1.4.4/Makefile.am stegemanitons/
    wireshark/Makefile.am
--- wireshark-1.4.4/Makefile.am 2011-03-01 12:06:08.000000000 -0500
+++ stegemanitons/wireshark/Makefile.am 2011-04-08 09:48:40.478196339 -0400
@@ -327,7 +327,8 @@ wireshark_LDADD = \
        @LIBGCRYPT_LIBS@                \
        @LIBGNUTLS_LIBS@                \
        @LIBSMI_LDFLAGS@                \
-       @PORTAUDIO_LIBS@
+       @PORTAUDIO_LIBS@                \
+       @BLUETOOTH_LIBS@
 wireshark_CFLAGS = $(AM_CLEAN_CFLAGS) $(py_dissectors_dir)

 if ENABLE_STATIC
@@ -354,7 +355,8 @@ tshark_LDADD = \
        @CORESERVICES_FRAMEWORKS@       \
        @LIBGCRYPT_LIBS@                \
        @LIBGNUTLS_LIBS@                \
-       @LIBSMI_LDFLAGS@
+       @LIBSMI_LDFLAGS@                \
+       @BLUETOOTH_LIBS@
 tshark_CFLAGS = $(AM_CLEAN_CFLAGS) $(py_dissectors_dir)

 if ENABLE_STATIC
@@ -381,7 +383,8 @@ rawshark_LDADD = \
        @CORESERVICES_FRAMEWORKS@       \
        @LIBGCRYPT_LIBS@                \
        @LIBGNUTLS_LIBS@                \
-       @LIBSMI_LDFLAGS@
+       @LIBSMI_LDFLAGS@                \
+       @BLUETOOTH_LIBS@
 rawshark_CFLAGS = $(AM_CLEAN_CFLAGS) $(py_dissectors_dir)

 # Libraries with which to link text2pcap.
@@ -458,7 +461,8 @@ dumpcap_LDADD = \
        @SOCKET_LIBS@                   \
        @NSL_LIBS@                      \
        @CORESERVICES_FRAMEWORKS@       \
-       @LIBCAP_LIBS@
+       @LIBCAP_LIBS@           \
+       @BLUETOOTH_LIBS@
 dumpcap_CFLAGS = $(AM_CLEAN_CFLAGS) $(py_dissectors_dir)

 # Common headers
diff -rupN -x .gitignore -x configure -x svn_version.xml -x 'grammar.[ch]' -x '
    dtd_grammar.[ch]' -x '*wslua.[ch]' -x faq.txt -x Info.plist -x 'mate_grammar.[ch]' -
    x ps.c -x svnversion.h -x idl2wrs.sh wireshark-1.4.4/Makefile.in stegemanitons/
    wireshark/Makefile.in
--- wireshark-1.4.4/Makefile.in 2011-03-01 12:07:55.000000000 -0500
+++ stegemanitons/wireshark/Makefile.in 2011-05-13 08:14:18.416669996 -0400
@@ -325,6 +325,7 @@ AUTOCONF = @AUTOCONF@
 AUTOHEADER = @AUTOHEADER@
 AUTOMAKE = @AUTOMAKE@
 AWK = @AWK@
+BLUETOOTH_LIBS = @BLUETOOTH_LIBS@
 CC = @CC@
 CCDEPMODE = @CCDEPMODE@
 CC_FOR_BUILD = @CC_FOR_BUILD@
@@ -1097,7 +1098,8 @@ wireshark_LDADD = \
        @LIBGCRYPT_LIBS@                \
        @LIBGNUTLS_LIBS@                \
        @LIBSMI_LDFLAGS@                \
-       @PORTAUDIO_LIBS@
+       @PORTAUDIO_LIBS@                \
```

```
+       @BLUETOOTH_LIBS@

 wireshark_CFLAGS = $(AM_CLEAN_CFLAGS) $(py_dissectors_dir)
 @ENABLE_STATIC_FALSE@tshark_LDFLAGS = -export-dynamic
@@ -1121,7 +1123,8 @@ tshark_LDADD = \
        @CORESERVICES_FRAMEWORKS@       \
        @LIBGCRYPT_LIBS@                \
        @LIBGNUTLS_LIBS@                \
-       @LIBSMI_LDFLAGS@
+       @LIBSMI_LDFLAGS@                \
+       @BLUETOOTH_LIBS@

 tshark_CFLAGS = $(AM_CLEAN_CFLAGS) $(py_dissectors_dir)
 @ENABLE_STATIC_FALSE@rawshark_LDFLAGS = -export-dynamic
@@ -1145,7 +1148,8 @@ rawshark_LDADD = \
        @CORESERVICES_FRAMEWORKS@       \
        @LIBGCRYPT_LIBS@                \
        @LIBGNUTLS_LIBS@                \
-       @LIBSMI_LDFLAGS@
+       @LIBSMI_LDFLAGS@                \
+       @BLUETOOTH_LIBS@

 rawshark_CFLAGS = $(AM_CLEAN_CFLAGS) $(py_dissectors_dir)

@@ -1229,7 +1233,8 @@ dumpcap_LDADD = \
        @SOCKET_LIBS@                   \
        @NSL_LIBS@                      \
        @CORESERVICES_FRAMEWORKS@       \
-       @LIBCAP_LIBS@
+       @LIBCAP_LIBS@           \
+       @BLUETOOTH_LIBS@

 dumpcap_CFLAGS = $(AM_CLEAN_CFLAGS) $(py_dissectors_dir)

diff -rupN -x .gitignore -x configure -x svn_version.xml -x 'grammar.[ch]' -x '
    dtd_grammar.[ch]' -x '*wslua.[ch]' -x faq.txt -x Info.plist -x 'mate_grammar.[ch]' -
    x ps.c -x svnversion.h -x idl2wrs.sh wireshark-1.4.4/acinclude.m4 stegemanitons/
    wireshark/acinclude.m4
--- wireshark-1.4.4/acinclude.m4        2011-03-01 12:06:08.000000000 -0500
+++ stegemanitons/wireshark/acinclude.m4        2011-04-08 09:48:40.478196339 -0400
@@ -1681,6 +1681,36 @@ AC_DEFUN([AC_WIRESHARK_GEOIP_CHECK],
        fi
 ])

+#
+# AC_WIRESHARK_BLUETOOTH_CHECK
+#
+AC_DEFUN([AC_WIRESHARK_BLUETOOTH_CHECK],
+[
+       want_bluetooth=defaultyes
+
+       if test "x$want_bluetooth" = "xdefaultyes"; then
+               want_bluetooth=yes
+               if test "x$ac_cv_enable_usr_local" = "xyes" ; then
+                       withval=/usr/local
+                       if test -d "$withval"; then
+                               AC_WIRESHARK_ADD_DASH_L(LDFLAGS, ${withval}/lib)
+                       fi
+               fi
+       fi
+
+       if test "x$want_bluetooth" = "xyes"; then
+               AC_CHECK_LIB(bluetooth, hci_devid,
+                 [
+                   BLUETOOTH_LIBS=-lbluetooth
+               AC_DEFINE(HAVE_BLUETOOTH_H, 1, [Define to use BlueZ library])
+               have_good_bluetooth=yes
+                 ],,
+               )
+       else
```

```
+                AC_MSG_RESULT(not required)
+        fi
+])
+
 #AC_WIRESHARK_GCC_LDFLAGS_CHECK
 #
 # $1 : ldflag(s) to test
diff -rupN -x .gitignore -x configure -x svn_version.xml -x 'grammar.[ch]' -x '
    dtd_grammar.[ch]' -x '*wslua.[ch]' -x faq.txt -x Info.plist -x 'mate_grammar.[ch]' -
    x ps.c -x svnversion.h -x idl2wrs.sh wireshark-1.4.4/capture-pcap-util-unix.c
    stegemanitons/wireshark/capture-pcap-util-unix.c
--- wireshark-1.4.4/capture-pcap-util-unix.c     2011-03-01 12:06:08.000000000 -0500
+++ stegemanitons/wireshark/capture-pcap-util-unix.c     2011-04-18 09:06:03.963742762
    -0400
@@ -58,10 +58,21 @@ struct rtentry;
 # include <sys/sockio.h>
 #endif

+#ifdef HAVE_BLUETOOTH_H
+#include <bluetooth/bluetooth.h>
+#include <bluetooth/hci.h>
+#include <bluetooth/hci_lib.h>
+
+#define BT_IFACE "hci"
+#endif
+
 #include "capture_ifinfo.h"
 #include "capture-pcap-util.h"
 #include "capture-pcap-util-int.h"

+// Used to test local code
+// #undef HAVE_PCAP_FINDALLDEVS
+
 #ifndef HAVE_PCAP_FINDALLDEVS
 struct search_user_data {
        char     *name;
@@ -282,6 +293,60 @@ get_interface_list(int *err, char **err_
        }
 #endif

+#ifdef HAVE_BLUETOOTH_H
+    struct hci_dev_list_req *dl;
+    struct hci_dev_req *dr;
+    int i, sk;
+    struct hci_dev_info di;
+    int dd;
+    char dev_name[20], dev_descr[40];
+
+    sk = socket(AF_BLUETOOTH, SOCK_RAW, BTPROTO_HCI);
+    if (sk < 0)
+        goto no_bt;
+
+    dl = malloc(HCI_MAX_DEV * sizeof(*dr) + sizeof(*dl));
+    if (!dl)
+        goto close_bt;
+
+    memset(dl, 0, HCI_MAX_DEV * sizeof(*dr) + sizeof(*dl));
+    dl->dev_num = HCI_MAX_DEV;
+
+    if (ioctl(sk, HCIGETDEVLIST, dl) < 0)
+        goto free_bt;
+
+    dr = dl->dev_req;
+    for (i = 0; i < dl->dev_num; i++, dr++) {
+        if (hci_devinfo(dr->dev_id, &di) < 0)
+            continue;
+
+        if (!hci_test_bit(HCI_UP, &di.flags))
+            continue;
+
```

```
+        dd = hci_open_dev(di.dev_id);
+        if (dd < 0)
+            continue;
+        hci_close_dev(dd);
+
+        snprintf(dev_name, 20, BT_IFACE"%d", dr->dev_id);
+        if (hci_test_bit(HCI_RAW, &di.flags))
+            snprintf(dev_descr, 40, "RAW Bluetooth adapter number %d", i);
+        else
+            snprintf(dev_descr, 40, "Bluetooth adapter number %d", i);
+
+        if_info = if_info_new(dev_name, dev_descr);
+        if_info->loopback = FALSE;
+        il = g_list_insert(il, if_info, nonloopback_pos);
+        nonloopback_pos++;
+    }
+
+free_bt:
+    free(dl);
+close_bt:
+    close(sk);
+no_bt:
+#endif
+
        g_free(ifc.ifc_buf);
        close(sock);

diff -rupN -x .gitignore -x configure -x svn_version.xml -x 'grammar.[ch]' -x '
    dtd_grammar.[ch]' -x '*wslua.[ch]' -x faq.txt -x Info.plist -x 'mate_grammar.[ch]' -
    x ps.c -x svnversion.h -x idl2wrs.sh wireshark-1.4.4/capture_opts.c stegemanitons/
    wireshark/capture_opts.c
--- wireshark-1.4.4/capture_opts.c      2011-03-01 12:06:11.000000000 -0500
+++ stegemanitons/wireshark/capture_opts.c      2011-05-11 11:36:34.225507466 -0400
@@ -55,6 +55,7 @@ static gboolean capture_opts_output_to_p
 void
 capture_opts_init(capture_options *capture_opts, void *cf)
 {
+  memset(capture_opts->master_mac, 0, 6);
   capture_opts->cf                    = cf;
   capture_opts->cfilter               = g_strdup("");     /* No capture filter string
       specified */
   capture_opts->iface                 = NULL;             /* Default is "pick the
      first interface" */
@@ -177,6 +178,9 @@ capture_opts_log(const char *log_domain,
 #ifdef _WIN32
     g_log(log_domain, log_level, "SignalPipeWrite     : %d", capture_opts->
        signal_pipe_write_fd);
 #endif
+    g_log(log_domain, log_level, "Bluetooth master MAC: %02x:%02x:%02x:%02x:%02x:%02x",
+            capture_opts->master_mac[0], capture_opts->master_mac[1], capture_opts->
    master_mac[2],
+            capture_opts->master_mac[3], capture_opts->master_mac[4], capture_opts->
    master_mac[5]);
 }

 /*
@@ -473,6 +477,12 @@ capture_opts_add_opt(capture_options *ca
     case 'I':        /* Capture in monitor mode */
         capture_opts->monitor_mode = TRUE;
         break;
+    case 'U':
+        sscanf(optarg_str_p, "%02x:%02x:%02x:%02x:%02x:%02x",
+                &capture_opts->master_mac[0], &capture_opts->master_mac[1],
+                &capture_opts->master_mac[2], &capture_opts->master_mac[3],
+                &capture_opts->master_mac[4], &capture_opts->master_mac[5]);
+        break;
 #endif
     case 'k':        /* Start capture immediately */
         *start_capture = TRUE;
```

```
diff -rupN -x .gitignore -x configure -x svn_version.xml -x 'grammar.[ch]' -x '
    dtd_grammar.[ch]' -x '*wslua.[ch]' -x faq.txt -x Info.plist -x 'mate_grammar.[ch]' -
    x ps.c -x svnversion.h -x idl2wrs.sh wireshark-1.4.4/capture_opts.h stegemanitons/
    wireshark/capture_opts.h
--- wireshark-1.4.4/capture_opts.h      2011-03-01 12:06:11.000000000 -0500
+++ stegemanitons/wireshark/capture_opts.h      2011-05-06 13:59:52.188717587 -0400
@@ -161,6 +161,7 @@ typedef struct capture_options_tag {
     uid_t owner;                    /**< owner of the cfile */
     gid_t group;                    /**< group of the cfile */
 #endif
+    unsigned char master_mac[6];
 } capture_options;

 /* initialize the capture_options with some reasonable values */
diff -rupN -x .gitignore -x configure -x svn_version.xml -x 'grammar.[ch]' -x '
    dtd_grammar.[ch]' -x '*wslua.[ch]' -x faq.txt -x Info.plist -x 'mate_grammar.[ch]' -
    x ps.c -x svnversion.h -x idl2wrs.sh wireshark-1.4.4/capture_sync.c stegemanitons/
    wireshark/capture_sync.c
--- wireshark-1.4.4/capture_sync.c      2011-03-01 12:06:07.000000000 -0500
+++ stegemanitons/wireshark/capture_sync.c      2011-05-13 08:13:10.620584275 -0400
@@ -334,6 +334,7 @@ sync_pipe_start(capture_options *capture
 #ifdef HAVE_PCAP_SETSAMPLING
     char ssampling[ARGV_NUMBER_LEN];
 #endif
+    char master_mac[18];
 #if defined(_WIN32) || defined(HAVE_PCAP_CREATE)
     char buffer_size[ARGV_NUMBER_LEN];
 #endif
@@ -468,6 +469,16 @@ sync_pipe_start(capture_options *capture
     }
 #endif

+    unsigned char tmp[6] = {0, 0, 0, 0, 0, 0};
+    if (memcmp(capture_opts->master_mac, tmp, 6)) {
+        argv = sync_pipe_add_arg(argv, &argc, "-U");
+        g_snprintf(master_mac, sizeof(master_mac), "%02x:%02x:%02x:%02x:%02x:%02x",
+                capture_opts->master_mac[0], capture_opts->master_mac[1],
+                capture_opts->master_mac[2], capture_opts->master_mac[3],
+                capture_opts->master_mac[4], capture_opts->master_mac[5]);
+        argv = sync_pipe_add_arg(argv, &argc, master_mac);
+    }
+
     /* dumpcap should be running in capture child mode (hidden feature) */
 #ifndef DEBUG_CHILD
     argv = sync_pipe_add_arg(argv, &argc, "-Z");
diff -rupN -x .gitignore -x configure -x svn_version.xml -x 'grammar.[ch]' -x '
    dtd_grammar.[ch]' -x '*wslua.[ch]' -x faq.txt -x Info.plist -x 'mate_grammar.[ch]' -
    x ps.c -x svnversion.h -x idl2wrs.sh wireshark-1.4.4/config.h.in stegemanitons/
    wireshark/config.h.in
--- wireshark-1.4.4/config.h.in 2011-03-01 12:06:50.000000000 -0500
+++ stegemanitons/wireshark/config.h.in 2011-05-13 08:14:57.317210409 -0400
@@ -27,6 +27,9 @@
 /* Define to 1 if you have the <arpa/nameser.h> header file. */
 #undef HAVE_ARPA_NAMESER_H

+/* Define to use BlueZ library */
+#undef HAVE_BLUETOOTH_H
+
 /* Define to use c-ares library */
 #undef HAVE_C_ARES

diff -rupN -x .gitignore -x configure -x svn_version.xml -x 'grammar.[ch]' -x '
    dtd_grammar.[ch]' -x '*wslua.[ch]' -x faq.txt -x Info.plist -x 'mate_grammar.[ch]' -
    x ps.c -x svnversion.h -x idl2wrs.sh wireshark-1.4.4/configure.in stegemanitons/
    wireshark/configure.in
--- wireshark-1.4.4/configure.in        2011-03-01 12:06:07.000000000 -0500
+++ stegemanitons/wireshark/configure.in        2011-05-13 08:13:10.660564275 -0400
@@ -99,7 +99,33 @@ else
 fi
```

```
 AC_PATH_PROG(LEX, flex)
-AC_PATH_PROG(PYTHON, python)
+
+# Try to make sure we're using Python 2
+AC_PATH_PROG(PYTHON, python2)
+if test "x$PYTHON" == x
+then
+    AC_PATH_PROG(PYTHON, python2.7)
+    if test "x$PYTHON" == x
+    then
+        AC_PATH_PROG(PYTHON, python2.6)
+        if test "x$PYTHON" == x
+        then
+            AC_PATH_PROG(PYTHON, python2.5)
+            if test "x$PYTHON" == x
+            then
+                AC_PATH_PROG(PYTHON, python2.4)
+                if test "x$PYTHON" == x
+                then
+                    AC_PATH_PROG(PYTHON, python2.3)
+                    if test "x$PYTHON" == x
+                    then
+                        AC_PATH_PROG(PYTHON, python)
+                    fi
+                fi
+            fi
+        fi
+    fi
+fi

 AC_SUBST(PERL)
 AC_SUBST(POD2MAN)
@@ -1518,6 +1544,31 @@ else
 fi
 AC_SUBST(GEOIP_LIBS)

+dnl BLUETOOTH Check
+BLUETOOTH_LIBS=''
+AC_MSG_CHECKING(whether to use the BlueZ library if available)
+
+AC_ARG_WITH(bluetooth,
+  AC_HELP_STRING( [--with-bluetooth@<:@=DIR@:>@],
+                  [use BlueZ (located in directory DIR, if supplied).   @<:@default=yes
    , if present@:>@]),
+[
+if   test "x$withval" = "xno";  then
+       want_bluetooth=no
+elif test "x$withval" = "xyes"; then
+       want_bluetooth=yes
+elif test -d "$withval"; then
+       want_bluetooth=yes
+       AC_WIRESHARK_ADD_DASH_L(LDFLAGS, ${withval}/lib)
+fi
+])
+if test "x$want_bluetooth" = "xno"; then
+       AC_MSG_RESULT(no)
+else
+       AC_MSG_RESULT(yes)
+       AC_WIRESHARK_BLUETOOTH_CHECK
+fi
+AC_SUBST(BLUETOOTH_LIBS)
+
 dnl Python devel Check
 AC_ARG_WITH(python,
     AC_HELP_STRING( [--with-python@<:@=DIR@:>@],
diff -rupN -x .gitignore -x configure -x svn_version.xml -x 'grammar.[ch]' -x '
    dtd_grammar.[ch]' -x '*wslua.[ch]' -x faq.txt -x Info.plist -x 'mate_grammar.[ch]' -
    x ps.c -x svnversion.h -x idl2wrs.sh wireshark-1.4.4/doc/dumpcap.pod stegemanitons/
    wireshark/doc/dumpcap.pod
--- wireshark-1.4.4/doc/dumpcap.pod      2011-03-01 12:02:28.000000000 -0500
```

```
+++ stegemanitons/wireshark/doc/dumpcap.pod     2011-05-11 09:07:50.600669208 -0400
@@ -22,6 +22,7 @@ S<[ B<-p> ]>
 S<[ B<-q> ]>
 S<[ B<-s> E<lt>capture snaplenE<gt> ]>
 S<[ B<-S> ]>
+S<[ B<-U> E<lt>MAC addressE<gt> ]>
 S<[ B<-v> ]>
 S<[ B<-w> E<lt>outfileE<gt> ]>
 S<[ B<-y> E<lt>capture link typeE<gt> ]>
@@ -240,6 +241,12 @@ memory, or saved to disk.  A value of 0

 Print statistics for each interface once every second.

+=item -U E<lt>MAC addressE<gt>
+
+Set the Bluetooth piconet master MAC address to the specified address. This is
+only useful when capturing on a Bluetooth interface that has RAW access, such
+as a CSR chipset with with custom firmware.
+
 =item -v

 Print the version and exit.
diff -rupN -x .gitignore -x configure -x svn_version.xml -x 'grammar.[ch]' -x '
    dtd_grammar.[ch]' -x '*wslua.[ch]' -x faq.txt -x Info.plist -x 'mate_grammar.[ch]' -
    x ps.c -x svnversion.h -x idl2wrs.sh wireshark-1.4.4/doc/tshark.pod stegemanitons/
    wireshark/doc/tshark.pod
--- wireshark-1.4.4/doc/tshark.pod      2011-03-01 12:02:27.000000000 -0500
+++ stegemanitons/wireshark/doc/tshark.pod      2011-05-11 09:03:20.215926333 -0400
@@ -34,6 +34,7 @@ S<[ B<-s> E<lt>capture snaplenE<gt> ]>
 S<[ B<-S> ]>
 S<[ B<-t> ad|a|r|d|dd|e ]>
 S<[ B<-T> pdml|psml|ps|text|fields ]>
+S<[ B<-U> E<lt>MAC addressE<gt>]>
 S<[ B<-v> ]>
 S<[ B<-V> ]>
 S<[ B<-w> E<lt>outfileE<gt>|- ]>
@@ -599,6 +600,11 @@ form specified by the B<-E> option. For
 would generate comma-separated values (CSV) output suitable for importing
 into your favorite spreadsheet program.

+=item -U E<lt>MAC addressE<gt>
+
+Set the Bluetooth piconet master MAC address to the specified address. This is
+only useful when capturing on a Bluetooth interface that has RAW access, such
+as a CSR chipset with with custom firmware.

 =item -v

diff -rupN -x .gitignore -x configure -x svn_version.xml -x 'grammar.[ch]' -x '
    dtd_grammar.[ch]' -x '*wslua.[ch]' -x faq.txt -x Info.plist -x 'mate_grammar.[ch]' -
    x ps.c -x svnversion.h -x idl2wrs.sh wireshark-1.4.4/docbook/wsug_src/WSUG_app_tools
    .xml stegemanitons/wireshark/docbook/wsug_src/WSUG_app_tools.xml
--- wireshark-1.4.4/docbook/wsug_src/WSUG_app_tools.xml 2011-03-01 12:02:28.000000000
    -0500
+++ stegemanitons/wireshark/docbook/wsug_src/WSUG_app_tools.xml 2011-05-11
    08:58:48.488521778 -0400
@@ -101,6 +101,7 @@ Capture interface:
   -L                       print list of link-layer types of iface and exit
   -S                       print statistics for each interface once every second
   -M                       for -D, -L, and -S produce machine-readable output
+  -U &lt;MAC address&gt;          use MAC address as Bluetooth piconet master

 Stop conditions:
   -c &lt;packet count&gt;         stop after n packets (def: infinite)
diff -rupN -x .gitignore -x configure -x svn_version.xml -x 'grammar.[ch]' -x '
    dtd_grammar.[ch]' -x '*wslua.[ch]' -x faq.txt -x Info.plist -x 'mate_grammar.[ch]' -
    x ps.c -x svnversion.h -x idl2wrs.sh wireshark-1.4.4/dumpcap.c stegemanitons/
    wireshark/dumpcap.c
--- wireshark-1.4.4/dumpcap.c    2011-03-01 12:06:07.000000000 -0500
+++ stegemanitons/wireshark/dumpcap.c    2011-05-11 09:20:49.404414158 -0400
```

```
@@ -362,6 +362,7 @@ print_usage(gboolean print_ver) {
    fprintf(output, "  -L                       print list of link-layer types of iface
        and exit\n");
    fprintf(output, "  -S                       print statistics for each interface once
        every second\n");
    fprintf(output, "  -M                       for -D, -L, and -S produce machine-
        readable output\n");
+   fprintf(output, "  -U <MAC address>         use MAC address as Bluetooth piconet
     master\n");
    fprintf(output, "\n");
 #ifdef HAVE_PCAP_REMOTE
    fprintf(output, "\nRPCAP options:\n");
@@ -1992,6 +1993,7 @@ capture_loop_open_input(capture_options
      ld->pcap_h = pcap_create(capture_opts->iface, open_err_str);
      if (ld->pcap_h != NULL) {
        pcap_set_snaplen(ld->pcap_h, capture_opts->has_snaplen ? capture_opts->snaplen :
            WTAP_MAX_PACKET_SIZE);
+       pcap_set_master_bd_addr(ld->pcap_h, capture_opts->master_mac);
        pcap_set_promisc(ld->pcap_h, capture_opts->promisc_mode);
        pcap_set_timeout(ld->pcap_h, CAP_READ_TIMEOUT);

@@ -3242,11 +3244,13 @@ main(int argc, char *argv[])

 #ifdef HAVE_PCAP_CREATE
 #define OPTSTRING_I "I"
+#define OPTSTRING_U "U:"
 #else
 #define OPTSTRING_I ""
+#define OPTSTRING_U ""
 #endif

-#define OPTSTRING "a:" OPTSTRING_A "b:" OPTSTRING_B "c:Df:hi:" OPTSTRING_I "L"
    OPTSTRING_m "Mnpq" OPTSTRING_r "Ss:" OPTSTRING_u "vw:y:Z:"
+#define OPTSTRING "a:" OPTSTRING_A "b:" OPTSTRING_B "c:Df:hi:" OPTSTRING_I "L"
    OPTSTRING_m "Mnpq" OPTSTRING_r "Ss:" OPTSTRING_u OPTSTRING_U "vw:y:Z:"

 #ifdef DEBUG_CHILD_DUMPCAP
    if ((debug_log = ws_fopen("dumpcap_debug_log.tmp","w")) == NULL) {
@@ -3534,6 +3538,7 @@ main(int argc, char *argv[])
 #endif /* _WIN32 or HAVE_PCAP_CREATE */
 #ifdef HAVE_PCAP_CREATE
        case 'I':        /* Monitor mode */
+       case 'U':        /* Set master MAC address */
 #endif
          status = capture_opts_add_opt(&global_capture_opts, opt, optarg, &start_capture
              );
          if(status != 0) {
diff -rupN -x .gitignore -x configure -x svn_version.xml -x 'grammar.[ch]' -x '
    dtd_grammar.[ch]' -x '*wslua.[ch]' -x faq.txt -x Info.plist -x 'mate_grammar.[ch]' -
    x ps.c -x svnversion.h -x idl2wrs.sh wireshark-1.4.4/epan/prefs.c stegemanitons/
    wireshark/epan/prefs.c
--- wireshark-1.4.4/epan/prefs.c        2011-03-01 12:05:50.000000000 -0500
+++ stegemanitons/wireshark/epan/prefs.c        2011-05-10 11:33:51.813545192 -0400
@@ -1220,6 +1220,7 @@ init_prefs(void) {
    prefs.capture_real_time             = TRUE;
    prefs.capture_auto_scroll           = TRUE;
    prefs.capture_show_info             = FALSE;
+   prefs.capture_master_mac            = NULL;

  /* set the default values for the name resolution dialog box */
    prefs.name_resolve                  = RESOLV_ALL ^ RESOLV_NETWORK;
@@ -1257,6 +1258,7 @@ prefs_reset(void)
    g_free(prefs.capture_devices_descr);
    g_free(prefs.capture_devices_hide);
    g_free(prefs.capture_devices_monitor_mode);
+   g_free(prefs.capture_master_mac);

    uat_unload_all();
    oids_cleanup();
@@ -1771,6 +1773,7 @@ prefs_capture_device_monitor_mode(const
```

```
 #define PRS_CAP_REAL_TIME              "capture.real_time_update"
 #define PRS_CAP_AUTO_SCROLL            "capture.auto_scroll"
 #define PRS_CAP_SHOW_INFO              "capture.show_info"
+#define PRS_CAP_MASTER_MAC             "capture.master_mac"

 #define RED_COMPONENT(x)   (guint16) (((((x) >> 16) & 0xff) * 65535 / 255))
 #define GREEN_COMPONENT(x) (guint16) (((((x) >>  8) & 0xff) * 65535 / 255))
@@ -2251,6 +2254,9 @@ set_pref(gchar *pref_name, gchar *value,
     prefs.capture_auto_scroll = ((g_ascii_strcasecmp(value, "true") == 0)?TRUE:FALSE);
   } else if (strcmp(pref_name, PRS_CAP_SHOW_INFO) == 0) {
     prefs.capture_show_info = ((g_ascii_strcasecmp(value, "true") == 0)?TRUE:FALSE);
+  } else if (strcmp(pref_name, PRS_CAP_MASTER_MAC) == 0) {
+    g_free(prefs.capture_master_mac);
+    prefs.capture_master_mac = g_strdup(value);

 /* handle the global options */
   } else if (strcmp(pref_name, PRS_NAME_RESOLVE) == 0 ||
@@ -3165,6 +3171,12 @@ write_prefs(char **pf_path_return)
   fprintf(pf, PRS_CAP_SHOW_INFO ": %s\n",
           prefs.capture_show_info == TRUE ? "TRUE" : "FALSE");

+  if (prefs.capture_master_mac != NULL) {
+    fprintf(pf, "\n# Bluetooth piconet master MAC\n");
+    fprintf(pf, "# Ex: 01:23:45:67:89:ab\n");
+    fprintf(pf, PRS_CAP_MASTER_MAC ": %s\n", prefs.capture_master_mac);
+  }
+
   fprintf (pf, "\n######## Printing ########\n");

   fprintf (pf, "\n# Can be one of \"text\" or \"postscript\".\n"
@@ -3307,6 +3319,7 @@ copy_prefs(e_prefs *dest, e_prefs *src)
   dest->capture_real_time = src->capture_real_time;
   dest->capture_auto_scroll = src->capture_auto_scroll;
   dest->capture_show_info = src->capture_show_info;
+  dest->capture_master_mac = g_strdup(src->capture_master_mac);
   dest->name_resolve = src->name_resolve;
   dest->name_resolve_concurrency = src->name_resolve_concurrency;
   dest->display_hidden_proto_items = src->display_hidden_proto_items;
@@ -3364,6 +3377,10 @@ free_prefs(e_prefs *pr)
     g_free(pr->capture_devices_monitor_mode);
     pr->capture_devices_monitor_mode = NULL;
   }
+  if (pr->capture_master_mac != NULL) {
+    g_free(pr->capture_master_mac);
+    pr->capture_master_mac = NULL;
+  }
 }

 static void
diff -rupN -x .gitignore -x configure -x svn_version.xml -x 'grammar.[ch]' -x '
    dtd_grammar.[ch]' -x '*wslua.[ch]' -x faq.txt -x Info.plist -x 'mate_grammar.[ch]' -
    x ps.c -x svnversion.h -x idl2wrs.sh wireshark-1.4.4/epan/prefs.h stegemanitons/
    wireshark/epan/prefs.h
--- wireshark-1.4.4/epan/prefs.h        2011-03-01 12:05:50.000000000 -0500
+++ stegemanitons/wireshark/epan/prefs.h        2011-05-10 07:58:25.729682744 -0400
@@ -159,6 +159,7 @@ typedef struct _e_prefs {
   gboolean capture_real_time;
   gboolean capture_auto_scroll;
   gboolean capture_show_info;
+  gchar    *capture_master_mac;
   guint    rtp_player_max_visible;
   guint    tap_update_interval;
   gboolean display_hidden_proto_items;
diff -rupN -x .gitignore -x configure -x svn_version.xml -x 'grammar.[ch]' -x '
    dtd_grammar.[ch]' -x '*wslua.[ch]' -x faq.txt -x Info.plist -x 'mate_grammar.[ch]' -
    x ps.c -x svnversion.h -x idl2wrs.sh wireshark-1.4.4/gtk/capture_dlg.c stegemanitons
    /wireshark/gtk/capture_dlg.c
--- wireshark-1.4.4/gtk/capture_dlg.c   2011-03-01 12:03:49.000000000 -0500
+++ stegemanitons/wireshark/gtk/capture_dlg.c   2011-05-13 11:13:49.445242861 -0400
@@ -94,6 +94,13 @@
```

```
 #define E_CAP_PCAP_NG_KEY                "cap_pcap_ng"
 #define E_CAP_FILT_KEY                   "cap_filter_te"
 #define E_CAP_FILE_TE_KEY                "cap_file_te"
+/*Bluetooth*/
+#define E_CAP_BLUE_TE1_KEY               "cap_blue_te1"
+#define E_CAP_BLUE_TE2_KEY               "cap_blue_te2"
+#define E_CAP_BLUE_TE3_KEY               "cap_blue_te3"
+#define E_CAP_BLUE_TE4_KEY               "cap_blue_te4"
+#define E_CAP_BLUE_TE5_KEY               "cap_blue_te5"
+#define E_CAP_BLUE_TE6_KEY               "cap_blue_te6"
 #define E_CAP_MULTI_FILES_ON_CB_KEY      "cap_multi_files_on_cb"
 #define E_CAP_RING_FILESIZE_CB_KEY       "cap_ring_filesize_cb"
 #define E_CAP_RING_FILESIZE_SB_KEY       "cap_ring_filesize_sb"
@@ -278,6 +285,12 @@ set_if_capabilities(gboolean monitor_mod
   guint num_supported_link_types;
   GtkWidget *linktype_om = (GtkWidget *) g_object_get_data(G_OBJECT(cap_open_w),
       E_CAP_LT_OM_KEY);
   GtkWidget *linktype_lb = g_object_get_data(G_OBJECT(linktype_om),
       E_CAP_LT_OM_LABEL_KEY);
+  GtkWidget *blue_te1 = (GtkWidget *) g_object_get_data(G_OBJECT(cap_open_w),
+      E_CAP_BLUE_TE1_KEY);
+  GtkWidget *blue_te2 = (GtkWidget *) g_object_get_data(G_OBJECT(cap_open_w),
+      E_CAP_BLUE_TE2_KEY);
+  GtkWidget *blue_te3 = (GtkWidget *) g_object_get_data(G_OBJECT(cap_open_w),
+      E_CAP_BLUE_TE3_KEY);
+  GtkWidget *blue_te4 = (GtkWidget *) g_object_get_data(G_OBJECT(cap_open_w),
+      E_CAP_BLUE_TE4_KEY);
+  GtkWidget *blue_te5 = (GtkWidget *) g_object_get_data(G_OBJECT(cap_open_w),
+      E_CAP_BLUE_TE5_KEY);
+  GtkWidget *blue_te6 = (GtkWidget *) g_object_get_data(G_OBJECT(cap_open_w),
+      E_CAP_BLUE_TE6_KEY);
   GtkWidget *if_ip_lb;
   GString *ip_str = g_string_new("IP address: ");
   int ips = 0;
@@ -329,6 +342,14 @@ set_if_capabilities(gboolean monitor_mod
   }
 #endif

+  /* Set Bluetooth master MAC box sensitivity */
+  gtk_widget_set_sensitive(blue_te1, strstr(if_name, "hci") == if_name);
+  gtk_widget_set_sensitive(blue_te2, strstr(if_name, "hci") == if_name);
+  gtk_widget_set_sensitive(blue_te3, strstr(if_name, "hci") == if_name);
+  gtk_widget_set_sensitive(blue_te4, strstr(if_name, "hci") == if_name);
+  gtk_widget_set_sensitive(blue_te5, strstr(if_name, "hci") == if_name);
+  gtk_widget_set_sensitive(blue_te6, strstr(if_name, "hci") == if_name);
+
   /*
    * If the interface name is in the list of known interfaces, get
    * its list of link-layer types and set the option menu to display it.
@@ -1527,6 +1548,11 @@ capture_prep_cb(GtkWidget *w _U_, gpoint
                 *file_fr, *file_vb,
                 *file_hb, *file_bt, *file_lb, *file_te,
+
+                *blue_fr,*blue_vb,
+                *blue_hb,*blue_lb,*blue_te1,
+                *blue_te2,*blue_te3,*blue_te4,*blue_te5,*blue_te6,
+
                 *multi_tb, *multi_files_on_cb,
                 *ring_filesize_cb, *ring_filesize_sb, *ring_filesize_cbx,
                 *file_duration_cb, *file_duration_sb, *file_duration_cbx,
@@ -1567,6 +1593,8 @@ capture_prep_cb(GtkWidget *w _U_, gpoint
   guint32        value;
   gchar          *cap_title;
   gchar          *if_device;
+  gchar          **mac_tokens;
+  int            mac_token_len = -1;

   if (cap_open_w != NULL) {
     /* There's already a "Capture Options" dialog box; reactivate it. */
```

```
@@ -1974,6 +2002,95 @@ capture_prep_cb(GtkWidget *w _U_, gpoint
    gtk_container_set_border_width(GTK_CONTAINER(right_vb), 0);
    gtk_box_pack_start(GTK_BOX(main_hb), right_vb, FALSE, FALSE, 0);

+   /* Bluetooth master options box */
+   blue_fr = gtk_frame_new("Bluetooth");
+   gtk_container_add(GTK_CONTAINER(left_vb), blue_fr);
+
+   blue_vb = gtk_vbox_new(FALSE, 3);
+   gtk_container_set_border_width(GTK_CONTAINER(blue_vb), 5);
+   gtk_container_add(GTK_CONTAINER(blue_fr), blue_vb);
+
+   blue_hb = gtk_hbox_new(FALSE, 3);
+   gtk_box_pack_start(GTK_BOX(blue_vb), blue_hb, FALSE, FALSE, 0);
+
+   blue_lb = gtk_label_new("Master Address:");
+   gtk_box_pack_start(GTK_BOX(blue_hb), blue_lb, FALSE, FALSE, 3);
+
+   if (prefs.capture_master_mac != NULL) {
+     mac_tokens = g_strsplit(prefs.capture_master_mac, ":", 6);
+     mac_token_len = g_strv_length(mac_tokens);
+   }
+
+   blue_te1 = gtk_entry_new();
+   gtk_entry_set_max_length(blue_te1, 2);
+   gtk_entry_set_width_chars(blue_te1, 2);
+   gtk_tooltips_set_tip(tooltips, blue_te1,
+     "Enter the Master MAC Address. "
+     "If you don't enter something here, RAW capture will not be possible.",
+     NULL);
+   if (mac_token_len >= 1)
+     gtk_entry_set_text(GTK_ENTRY(blue_te1), mac_tokens[0]);
+   gtk_box_pack_start(GTK_BOX(blue_hb), blue_te1, TRUE, TRUE, 3);
+
+   blue_te2 = gtk_entry_new();
+   gtk_entry_set_max_length(blue_te2, 2);
+   gtk_entry_set_width_chars(blue_te2, 2);
+   gtk_tooltips_set_tip(tooltips, blue_te2,
+     "Enter the Master MAC Address. "
+     "If you don't enter something here, RAW capture will not be possible.",
+     NULL);
+   if (mac_token_len >= 2)
+     gtk_entry_set_text(GTK_ENTRY(blue_te2), mac_tokens[1]);
+   gtk_box_pack_start(GTK_BOX(blue_hb), blue_te2, TRUE, TRUE, 3);
+
+   blue_te3 = gtk_entry_new();
+   gtk_entry_set_max_length(blue_te3, 2);
+   gtk_entry_set_width_chars(blue_te3, 2);
+   gtk_tooltips_set_tip(tooltips, blue_te3,
+     "Enter the Master MAC Address. "
+     "If you don't enter something here, RAW capture will not be possible.",
+     NULL);
+   if (mac_token_len >= 3)
+     gtk_entry_set_text(GTK_ENTRY(blue_te3), mac_tokens[2]);
+   gtk_box_pack_start(GTK_BOX(blue_hb), blue_te3, TRUE, TRUE, 3);
+
+   blue_te4 = gtk_entry_new();
+   gtk_entry_set_max_length(blue_te4, 2);
+   gtk_entry_set_width_chars(blue_te4, 2);
+   gtk_tooltips_set_tip(tooltips, blue_te4,
+     "Enter the Master MAC Address. "
+     "If you don't enter something here, RAW capture will not be possible.",
+     NULL);
+   if (mac_token_len >= 4)
+     gtk_entry_set_text(GTK_ENTRY(blue_te4), mac_tokens[3]);
+   gtk_box_pack_start(GTK_BOX(blue_hb), blue_te4, TRUE, TRUE, 3);
+
+   blue_te5 = gtk_entry_new();
+   gtk_entry_set_max_length(blue_te5, 2);
+   gtk_entry_set_width_chars(blue_te5, 2);
```

```
+  gtk_tooltips_set_tip(tooltips, blue_te5,
+    "Enter the Master MAC Address. "
+    "If you don't enter something here, RAW capture will not be possible.",
+    NULL);
+  if (mac_token_len >= 5)
+    gtk_entry_set_text(GTK_ENTRY(blue_te5), mac_tokens[4]);
+  gtk_box_pack_start(GTK_BOX(blue_hb), blue_te5, TRUE, TRUE, 3);
+
+  blue_te6 = gtk_entry_new();
+  gtk_entry_set_max_length(blue_te6, 2);
+  gtk_entry_set_width_chars(blue_te6, 2);
+  gtk_tooltips_set_tip(tooltips, blue_te6,
+    "Enter the Master MAC Address. "
+    "If you don't enter something here, RAW capture will not be possible.",
+    NULL);
+  if (mac_token_len >= 6)
+    gtk_entry_set_text(GTK_ENTRY(blue_te6), mac_tokens[5]);
+  gtk_box_pack_start(GTK_BOX(blue_hb), blue_te6, TRUE, TRUE, 3);
+
+  if (mac_token_len >= 1)
+    g_strfreev(mac_tokens);
+  /*end*/
+
   /* Capture file-related options frame */
   file_fr = gtk_frame_new("Capture File(s)");
   gtk_container_add(GTK_CONTAINER(left_vb), file_fr);
@@ -2306,6 +2423,14 @@ capture_prep_cb(GtkWidget *w _U_, gpoint
   g_object_set_data(G_OBJECT(cap_open_w), E_CAP_PCAP_NG_KEY, pcap_ng_cb);
   g_object_set_data(G_OBJECT(cap_open_w), E_CAP_FILT_KEY,  filter_te);
   g_object_set_data(G_OBJECT(cap_open_w), E_CAP_FILE_TE_KEY,  file_te);
+  /*Bluetooth*/
+  g_object_set_data(G_OBJECT(cap_open_w), E_CAP_BLUE_TE1_KEY ,  blue_te1);
+  g_object_set_data(G_OBJECT(cap_open_w), E_CAP_BLUE_TE2_KEY ,  blue_te2);
+  g_object_set_data(G_OBJECT(cap_open_w), E_CAP_BLUE_TE3_KEY ,  blue_te3);
+  g_object_set_data(G_OBJECT(cap_open_w), E_CAP_BLUE_TE4_KEY ,  blue_te4);
+  g_object_set_data(G_OBJECT(cap_open_w), E_CAP_BLUE_TE5_KEY ,  blue_te5);
+  g_object_set_data(G_OBJECT(cap_open_w), E_CAP_BLUE_TE6_KEY ,  blue_te6);
+
   g_object_set_data(G_OBJECT(cap_open_w), E_CAP_MULTI_FILES_ON_CB_KEY,
       multi_files_on_cb);
   g_object_set_data(G_OBJECT(cap_open_w), E_CAP_RING_NBF_CB_KEY,  ringbuffer_nbf_cb);
   g_object_set_data(G_OBJECT(cap_open_w), E_CAP_RING_NBF_SB_KEY,  ringbuffer_nbf_sb);
@@ -2338,7 +2463,20 @@ capture_prep_cb(GtkWidget *w _U_, gpoint
   /* Set the sensitivity of various widgets as per the settings of other
      widgets. */
   capture_prep_adjust_sensitivity(NULL, cap_open_w);
-
+  if (global_capture_opts.iface != NULL) {
+    gtk_widget_set_sensitive(blue_te1,
+            strstr(global_capture_opts.iface, "hci") == global_capture_opts.iface);
+    gtk_widget_set_sensitive(blue_te2,
+            strstr(global_capture_opts.iface, "hci") == global_capture_opts.iface);
+    gtk_widget_set_sensitive(blue_te3,
+            strstr(global_capture_opts.iface, "hci") == global_capture_opts.iface);
+    gtk_widget_set_sensitive(blue_te4,
+            strstr(global_capture_opts.iface, "hci") == global_capture_opts.iface);
+    gtk_widget_set_sensitive(blue_te5,
+            strstr(global_capture_opts.iface, "hci") == global_capture_opts.iface);
+    gtk_widget_set_sensitive(blue_te6,
+            strstr(global_capture_opts.iface, "hci") == global_capture_opts.iface);
+  }
   /* Catch the "activate" signal on the text
      entries, so that if the user types Return there, we act as if the
     "OK" button had been selected, as happens if Return is typed if some
@@ -2678,7 +2816,8 @@ capture_dlg_prep(gpointer parent_w) {
             *ring_filesize_cb, *ring_filesize_sb, *ring_filesize_cbx,
             *file_duration_cb, *file_duration_sb, *file_duration_cbx,
             *stop_files_cb, *stop_files_sb,
-            *m_resolv_cb, *n_resolv_cb, *t_resolv_cb;
+            *m_resolv_cb, *n_resolv_cb, *t_resolv_cb,
```

```
+            *blue_te1, *blue_te2, *blue_te3, *blue_te4, *blue_te5, *blue_te6;
 #ifdef HAVE_PCAP_REMOTE
    GtkWidget *iftype_cbx;
 #endif
@@ -2693,6 +2832,33 @@ capture_dlg_prep(gpointer parent_w) {
    gchar *cf_name;
    gchar *dirname;
    gint32 tmp;
+   int tmp_hex;
+
+   /*Bluetooth*/
+   blue_te1 = (GtkWidget *) g_object_get_data(G_OBJECT(parent_w), E_CAP_BLUE_TE1_KEY);
+   blue_te2 = (GtkWidget *) g_object_get_data(G_OBJECT(parent_w), E_CAP_BLUE_TE2_KEY);
+   blue_te3 = (GtkWidget *) g_object_get_data(G_OBJECT(parent_w), E_CAP_BLUE_TE3_KEY);
+   blue_te4 = (GtkWidget *) g_object_get_data(G_OBJECT(parent_w), E_CAP_BLUE_TE4_KEY);
+   blue_te5 = (GtkWidget *) g_object_get_data(G_OBJECT(parent_w), E_CAP_BLUE_TE5_KEY);
+   blue_te6 = (GtkWidget *) g_object_get_data(G_OBJECT(parent_w), E_CAP_BLUE_TE6_KEY);
+
+   if (sscanf(gtk_entry_get_text(GTK_ENTRY(blue_te1)), "%02x", &tmp_hex) == 1)
+     global_capture_opts.master_mac[0] = (unsigned char)tmp_hex;
+
+   if (sscanf(gtk_entry_get_text(GTK_ENTRY(blue_te2)), "%02x", &tmp_hex) == 1)
+     global_capture_opts.master_mac[1] = (unsigned char)tmp_hex;
+
+   if (sscanf(gtk_entry_get_text(GTK_ENTRY(blue_te3)), "%02x", &tmp_hex) == 1)
+     global_capture_opts.master_mac[2] = (unsigned char)tmp_hex;
+
+   if (sscanf(gtk_entry_get_text(GTK_ENTRY(blue_te4)), "%02x", &tmp_hex) == 1)
+     global_capture_opts.master_mac[3] = (unsigned char)tmp_hex;
+
+   if (sscanf(gtk_entry_get_text(GTK_ENTRY(blue_te5)), "%02x", &tmp_hex) == 1)
+     global_capture_opts.master_mac[4] = (unsigned char)tmp_hex;
+
+   if (sscanf(gtk_entry_get_text(GTK_ENTRY(blue_te6)), "%02x", &tmp_hex) == 1)
+     global_capture_opts.master_mac[5] = (unsigned char)tmp_hex;

    if_cb       = (GtkWidget *) g_object_get_data(G_OBJECT(parent_w), E_CAP_IFACE_KEY);
 #ifdef HAVE_PCAP_REMOTE
diff -rupN -x .gitignore -x configure -x svn_version.xml -x 'grammar.[ch]' -x '
    dtd_grammar.[ch]' -x '*wslua.[ch]' -x faq.txt -x Info.plist -x 'mate_grammar.[ch]' -
    x ps.c -x svnversion.h -x idl2wrs.sh wireshark-1.4.4/gtk/capture_if_dlg.c
    stegemanitons/wireshark/gtk/capture_if_dlg.c
--- wireshark-1.4.4/gtk/capture_if_dlg.c        2011-03-01 12:03:49.000000000 -0500
+++ stegemanitons/wireshark/gtk/capture_if_dlg.c        2011-04-18 09:12:16.494054990
    -0400
@@ -408,7 +408,7 @@ GtkWidget * capture_get_if_icon(const if
    * XXX - this is for raw Bluetooth capture; what about IP-over-Bluetooth
    * devices?
    */
-  if ( strstr(if_info->name,"bluetooth") != NULL) {
+  if ( strstr(if_info->name,"hci") != NULL) {
     return pixbuf_to_widget(network_bluetooth_pb_data);
   }

diff -rupN -x .gitignore -x configure -x svn_version.xml -x 'grammar.[ch]' -x '
    dtd_grammar.[ch]' -x '*wslua.[ch]' -x faq.txt -x Info.plist -x 'mate_grammar.[ch]' -
    x ps.c -x svnversion.h -x idl2wrs.sh wireshark-1.4.4/gtk/filter_dlg.c stegemanitons/
    wireshark/gtk/filter_dlg.c
--- wireshark-1.4.4/gtk/filter_dlg.c    2011-03-01 12:03:48.000000000 -0500
+++ stegemanitons/wireshark/gtk/filter_dlg.c    2011-05-11 11:47:27.322136077 -0400
@@ -37,6 +37,9 @@
 #include "../filters.h"
 #include "../simple_dialog.h"

+#include "../capture_opts.h"
+#include "gtk/capture_globals.h"
+
 #include "gtk/main.h"
 #include "gtk/main_statusbar.h"
 #include "gtk/filter_dlg.h"
```

```
@@ -63,16 +66,26 @@
 #define E_FILT_DBLARG_KEY            "filter_dblarg"
 #define E_FILT_DBLACTIVATE_KEY       "filter_dblactivate"

+#define BLUE_TE1_KEY                "cap_blue_te1"
+#define BLUE_TE2_KEY                "cap_blue_te2"
+#define BLUE_TE3_KEY                "cap_blue_te3"
+#define BLUE_TE4_KEY                "cap_blue_te4"
+#define BLUE_TE5_KEY                "cap_blue_te5"
+#define BLUE_TE6_KEY                "cap_blue_te6"
+
 typedef struct _filter_cb_data {
   GList     *fl;
   GtkWidget *win;
 } filter_cb_data;

+static GtkWidget *mac_window();
+
 static GtkWidget *filter_dialog_new(GtkWidget *button, GtkWidget *filter_te,
                                     filter_list_type_t list_type,
                                     construct_args_t *construct_args);
 static void filter_dlg_dclick(GtkWidget *dummy, gpointer main_w_arg,
                   gpointer activate);
+static void mac_ok_cb(GtkWidget *ok_bt, GtkWidget *main_w);
 static void filter_dlg_ok_cb(GtkWidget *ok_bt, gpointer data);
 static void filter_dlg_apply_cb(GtkWidget *apply_bt, gpointer data);
 static void filter_apply(GtkWidget *main_w, gboolean destroy);
@@ -187,6 +200,12 @@ filter_button_destroy_cb(GtkWidget *butt
     }
 }

+static GtkWidget *global_mac_w;
+
+void mac_dialog_cb(GtkWidget *w _U_) {
+       global_mac_w = mac_window();
+}
+
 #ifdef HAVE_LIBPCAP
 static GtkWidget *global_cfilter_w;

@@ -338,6 +357,124 @@ clear_list(GtkWidget *main_w) {
 }
 #endif /* 0 */

+static GtkWidget *mac_window() {
+       GtkWidget *main_w, *blue_fr,*blue_vb, *blue_hb,*blue_lb,*blue_te1, *blue_te2, *
    blue_te3,
+                 *blue_te4, *blue_te5, *blue_te6, *bbox, *ok_bt;
+       GtkTooltips *tooltips;
+       gchar **mac_tokens;
+       int mac_token_len = -1;
+       tooltips = gtk_tooltips_new();
+       main_w = dlg_conf_window_new("Bluetooth");
+       gtk_window_set_default_size(GTK_WINDOW(main_w), 400, 100);
+
+       blue_fr = gtk_frame_new("Bluetooth");
+       gtk_container_add(GTK_CONTAINER(main_w), blue_fr);
+
+       blue_vb = gtk_vbox_new(FALSE, 3);
+       gtk_container_set_border_width(GTK_CONTAINER(blue_vb), 5);
+       gtk_container_add(GTK_CONTAINER(blue_fr), blue_vb);
+
+       blue_hb = gtk_hbox_new(FALSE, 3);
+       gtk_box_pack_start(GTK_BOX(blue_vb), blue_hb, FALSE, FALSE, 0);
+
+       blue_lb = gtk_label_new("Master Address:");
+       gtk_box_pack_start(GTK_BOX(blue_hb), blue_lb, FALSE, FALSE, 3);
+
+       if (prefs.capture_master_mac) {
+               mac_tokens = g_strsplit(prefs.capture_master_mac, ":", 6);
```

```
+                mac_token_len = g_strv_length(mac_tokens);
+        }
+
+        blue_te1 = gtk_entry_new();
+        gtk_entry_set_max_length(blue_te1, 2);
+        gtk_entry_set_width_chars(blue_te1, 2);
+        gtk_tooltips_set_tip(tooltips, blue_te1,
+                "Enter the Master MAC Address. "
+                "If you don't enter something here, RAW capture will not be possible.",
+                NULL);
+        if (mac_token_len >= 1)
+                gtk_entry_set_text(GTK_ENTRY(blue_te1), mac_tokens[0]);
+        gtk_box_pack_start(GTK_BOX(blue_hb), blue_te1, TRUE, TRUE, 3);
+
+        blue_te2 = gtk_entry_new();
+        gtk_entry_set_max_length(blue_te2, 2);
+        gtk_entry_set_width_chars(blue_te2, 2);
+        gtk_tooltips_set_tip(tooltips, blue_te2,
+                "Enter the Master MAC Address. "
+                "If you don't enter something here, RAW capture will not be possible.",
+                NULL);
+        if (mac_token_len >= 2)
+                gtk_entry_set_text(GTK_ENTRY(blue_te2), mac_tokens[1]);
+        gtk_box_pack_start(GTK_BOX(blue_hb), blue_te2, TRUE, TRUE, 3);
+
+        blue_te3 = gtk_entry_new();
+        gtk_entry_set_max_length(blue_te3, 2);
+        gtk_entry_set_width_chars(blue_te3, 2);
+        gtk_tooltips_set_tip(tooltips, blue_te3,
+                "Enter the Master MAC Address. "
+                "If you don't enter something here, RAW capture will not be possible.",
+                NULL);
+        if (mac_token_len >= 3)
+                gtk_entry_set_text(GTK_ENTRY(blue_te3), mac_tokens[2]);
+        gtk_box_pack_start(GTK_BOX(blue_hb), blue_te3, TRUE, TRUE, 3);
+
+        blue_te4 = gtk_entry_new();
+        gtk_entry_set_max_length(blue_te4, 2);
+        gtk_entry_set_width_chars(blue_te4, 2);
+        gtk_tooltips_set_tip(tooltips, blue_te4,
+                "Enter the Master MAC Address. "
+                "If you don't enter something here, RAW capture will not be possible.",
+                NULL);
+        if (mac_token_len >= 4)
+                gtk_entry_set_text(GTK_ENTRY(blue_te4), mac_tokens[3]);
+        gtk_box_pack_start(GTK_BOX(blue_hb), blue_te4, TRUE, TRUE, 3);
+
+        blue_te5 = gtk_entry_new();
+        gtk_entry_set_max_length(blue_te5, 2);
+        gtk_entry_set_width_chars(blue_te5, 2);
+        gtk_tooltips_set_tip(tooltips, blue_te5,
+                "Enter the Master MAC Address. "
+                "If you don't enter something here, RAW capture will not be possible.",
+                NULL);
+        if (mac_token_len >= 5)
+                gtk_entry_set_text(GTK_ENTRY(blue_te5), mac_tokens[4]);
+        gtk_box_pack_start(GTK_BOX(blue_hb), blue_te5, TRUE, TRUE, 3);
+
+        blue_te6 = gtk_entry_new();
+        gtk_entry_set_max_length(blue_te6, 2);
+        gtk_entry_set_width_chars(blue_te6, 2);
+        gtk_tooltips_set_tip(tooltips, blue_te6,
+                "Enter the Master MAC Address. "
+                "If you don't enter something here, RAW capture will not be possible.",
+                NULL);
+        if (mac_token_len >= 6)
+                gtk_entry_set_text(GTK_ENTRY(blue_te6), mac_tokens[5]);
+        gtk_box_pack_start(GTK_BOX(blue_hb), blue_te6, TRUE, TRUE, 3);
+
+        bbox = dlg_button_row_new(GTK_STOCK_OK, NULL);
```

```
+       gtk_box_pack_start(GTK_BOX(blue_vb), bbox, FALSE, FALSE, 5);
+       gtk_widget_show(bbox);
+
+       if (mac_token_len >= 1)
+               g_strfreev(mac_tokens);
+
+       g_object_set_data(G_OBJECT(main_w), BLUE_TE1_KEY, blue_te1);
+       g_object_set_data(G_OBJECT(main_w), BLUE_TE2_KEY, blue_te2);
+       g_object_set_data(G_OBJECT(main_w), BLUE_TE3_KEY, blue_te3);
+       g_object_set_data(G_OBJECT(main_w), BLUE_TE4_KEY, blue_te4);
+       g_object_set_data(G_OBJECT(main_w), BLUE_TE5_KEY, blue_te5);
+       g_object_set_data(G_OBJECT(main_w), BLUE_TE6_KEY, blue_te6);
+
+       ok_bt = g_object_get_data(G_OBJECT(bbox), GTK_STOCK_OK);
+       g_signal_connect(ok_bt, "clicked", G_CALLBACK(mac_ok_cb), main_w);
+       gtk_tooltips_set_tip(tooltips, ok_bt,
+               "Set the Bluetooth MAC address.", NULL);
+
+       gtk_widget_show_all(main_w);
+       window_present(main_w);
+       return main_w;
+}
+
 static GtkWidget *
 filter_dialog_new(GtkWidget *button, GtkWidget *parent_filter_te,
                   filter_list_type_t list_type, construct_args_t *construct_args)
@@ -712,6 +849,42 @@ filter_dlg_dclick(GtkWidget *filter_l, g
 }

 static void
+mac_ok_cb(GtkWidget *ok_bt, GtkWidget *main_w) {
+    GtkWidget *blue_te1, *blue_te2, *blue_te3, *blue_te4, *blue_te5, *blue_te6;
+       int tmp_hex;
+
+    blue_te1 = (GtkWidget *) g_object_get_data(G_OBJECT(main_w), BLUE_TE1_KEY);
+    blue_te2 = (GtkWidget *) g_object_get_data(G_OBJECT(main_w), BLUE_TE2_KEY);
+    blue_te3 = (GtkWidget *) g_object_get_data(G_OBJECT(main_w), BLUE_TE3_KEY);
+    blue_te4 = (GtkWidget *) g_object_get_data(G_OBJECT(main_w), BLUE_TE4_KEY);
+    blue_te5 = (GtkWidget *) g_object_get_data(G_OBJECT(main_w), BLUE_TE5_KEY);
+    blue_te6 = (GtkWidget *) g_object_get_data(G_OBJECT(main_w), BLUE_TE6_KEY);
+
+       if (sscanf(gtk_entry_get_text(GTK_ENTRY(blue_te1)), "%02x", &tmp_hex) == 1)
+               global_capture_opts.master_mac[0] = (unsigned char)tmp_hex;
+
+       if (sscanf(gtk_entry_get_text(GTK_ENTRY(blue_te2)), "%02x", &tmp_hex) == 1)
+               global_capture_opts.master_mac[1] = (unsigned char)tmp_hex;
+
+       if (sscanf(gtk_entry_get_text(GTK_ENTRY(blue_te3)), "%02x", &tmp_hex) == 1)
+               global_capture_opts.master_mac[2] = (unsigned char)tmp_hex;
+
+       if (sscanf(gtk_entry_get_text(GTK_ENTRY(blue_te4)), "%02x", &tmp_hex) == 1)
+               global_capture_opts.master_mac[3] = (unsigned char)tmp_hex;
+
+       if (sscanf(gtk_entry_get_text(GTK_ENTRY(blue_te5)), "%02x", &tmp_hex) == 1)
+               global_capture_opts.master_mac[4] = (unsigned char)tmp_hex;
+
+       if (sscanf(gtk_entry_get_text(GTK_ENTRY(blue_te6)), "%02x", &tmp_hex) == 1)
+               global_capture_opts.master_mac[5] = (unsigned char)tmp_hex;
+
+    window_destroy(GTK_WINDOW(main_w));
+
+    if (global_capture_opts.state != CAPTURE_STOPPED)
+        capture_restart(&global_capture_opts);
+}
+
+static void
 filter_dlg_ok_cb(GtkWidget *ok_bt, gpointer data)
 {
     filter_list_type_t list_type = *(filter_list_type_t *)data;
```

```
diff -rupN -x .gitignore -x configure -x svn_version.xml -x 'grammar.[ch]' -x '
    dtd_grammar.[ch]' -x '*wslua.[ch]' -x faq.txt -x Info.plist -x 'mate_grammar.[ch]' -
    x ps.c -x svnversion.h -x idl2wrs.sh wireshark-1.4.4/gtk/filter_dlg.h stegemanitons/
    wireshark/gtk/filter_dlg.h
--- wireshark-1.4.4/gtk/filter_dlg.h    2011-03-01 12:03:48.000000000 -0500
+++ stegemanitons/wireshark/gtk/filter_dlg.h    2011-05-11 11:49:01.395077076 -0400
@@ -73,6 +73,8 @@ void filter_button_destroy_cb(GtkWidget
  */
 void cfilter_dialog_cb(GtkWidget *widget);

+void mac_dialog_cb(GtkWidget *widget);
+
 /** User requested the "Display Filter" dialog box by menu or toolbar.
  *
  * @param widget parent widget
diff -rupN -x .gitignore -x configure -x svn_version.xml -x 'grammar.[ch]' -x '
    dtd_grammar.[ch]' -x '*wslua.[ch]' -x faq.txt -x Info.plist -x 'mate_grammar.[ch]' -
    x ps.c -x svnversion.h -x idl2wrs.sh wireshark-1.4.4/gtk/main.c stegemanitons/
    wireshark/gtk/main.c
--- wireshark-1.4.4/gtk/main.c  2011-03-01 12:03:49.000000000 -0500
+++ stegemanitons/wireshark/gtk/main.c  2011-05-10 08:11:43.647199562 -0400
@@ -3689,6 +3689,15 @@ prefs_to_capture_opts(void)
     global_capture_opts.show_info       = prefs.capture_show_info;
     global_capture_opts.real_time_mode = prefs.capture_real_time;
     auto_scroll_live                    = prefs.capture_auto_scroll;
+    if (prefs.capture_master_mac != NULL) {
+        sscanf(prefs.capture_master_mac, "%02x:%02x:%02x:%02x:%02x:%02x",
+               &global_capture_opts.master_mac[0],
+               &global_capture_opts.master_mac[1],
+               &global_capture_opts.master_mac[2],
+               &global_capture_opts.master_mac[3],
+               &global_capture_opts.master_mac[4],
+               &global_capture_opts.master_mac[5]);
+    }
 #endif /* HAVE_LIBPCAP */

   /* Set the name resolution code's flags from the preferences. */
diff -rupN -x .gitignore -x configure -x svn_version.xml -x 'grammar.[ch]' -x '
    dtd_grammar.[ch]' -x '*wslua.[ch]' -x faq.txt -x Info.plist -x 'mate_grammar.[ch]' -
    x ps.c -x svnversion.h -x idl2wrs.sh wireshark-1.4.4/gtk/main_toolbar.c
    stegemanitons/wireshark/gtk/main_toolbar.c
--- wireshark-1.4.4/gtk/main_toolbar.c  2011-03-01 12:03:47.000000000 -0500
+++ stegemanitons/wireshark/gtk/main_toolbar.c  2011-05-11 11:50:52.043059920 -0400
@@ -58,6 +58,8 @@
 #include "gtk/keys.h"
 #include "gtk/recent.h"
 #include "gtk/packet_history.h"
+#include "../capture_opts.h"
+#include "gtk/capture_globals.h"

 #ifdef NEW_PACKET_LIST
 #include "gtk/new_packet_list.h"
@@ -77,7 +79,7 @@ static GtkToolItem *go_to_button, *go_to
 static GtkToolItem *display_filter_button;
 static GtkToolItem *zoom_in_button, *zoom_out_button, *zoom_100_button, *
     colorize_button;
 static GtkToolItem *resize_columns_button;
-static GtkToolItem *color_display_button, *prefs_button, *help_button;
+static GtkToolItem *color_display_button, *prefs_button, *help_button, *mac_button;

 #define SAVE_BUTTON_TOOLTIP_TEXT "Save this capture file..."
 #define SAVE_AS_BUTTON_TOOLTIP_TEXT "Save this capture file as..."
@@ -212,6 +214,10 @@ void set_toolbar_for_captured_packets(gb
                                 have_captured_packets);
         gtk_widget_set_sensitive(GTK_WIDGET(resize_columns_button),
                                 have_captured_packets);
+        gtk_widget_set_sensitive(GTK_WIDGET(mac_button),
+                !have_captured_packets ||
+                (global_capture_opts.iface &&
```

```
+                    strstr(global_capture_opts.iface, "hci") == global_capture_opts.
    iface));

         /* XXX - I don't see a reason why this should be done (as it is in the
          * menus) */
@@ -417,6 +423,9 @@ toolbar_new(void)

     toolbar_append_separator(main_tb);

+    toolbar_item(mac_button, main_tb,
+       WIRESHARK_STOCK_BLUETOOTH_MAC, tooltips, "Edit Bluetooth piconet master...",
    mac_dialog_cb, NULL);
+
     toolbar_item(help_button, main_tb,
        GTK_STOCK_HELP, tooltips, "Show some help...", topic_cb, GINT_TO_POINTER(
           HELP_CONTENT));

diff -rupN -x .gitignore -x configure -x svn_version.xml -x 'grammar.[ch]' -x '
    dtd_grammar.[ch]' -x '*wslua.[ch]' -x faq.txt -x Info.plist -x 'mate_grammar.[ch]' -
    x ps.c -x svnversion.h -x idl2wrs.sh wireshark-1.4.4/gtk/menus.c stegemanitons/
    wireshark/gtk/menus.c
--- wireshark-1.4.4/gtk/menus.c 2011-03-01 12:03:48.000000000 -0500
+++ stegemanitons/wireshark/gtk/menus.c 2011-05-06 13:59:24.205865346 -0400
@@ -734,6 +734,8 @@ static GtkItemFactoryEntry menu_items[]
                                0, "<StockItem>", WIRESHARK_STOCK_CAPTURE_RESTART,},
     {"/Capture/Capture _Filters...", NULL, GTK_MENU_FUNC(cfilter_dialog_cb),
                                0, "<StockItem>", WIRESHARK_STOCK_CAPTURE_FILTER,},
+    {"/Capture/_Bluetooth piconet master...", NULL, GTK_MENU_FUNC(mac_dialog_cb),
+                               0, "<StockItem>", WIRESHARK_STOCK_BLUETOOTH_MAC,},
 #endif /* HAVE_LIBPCAP */
     {"/_Analyze", NULL, NULL, 0, "<Branch>", NULL,},
     {"/Analyze/_Display Filters...", NULL, GTK_MENU_FUNC(dfilter_dialog_cb),
diff -rupN -x .gitignore -x configure -x svn_version.xml -x 'grammar.[ch]' -x '
    dtd_grammar.[ch]' -x '*wslua.[ch]' -x faq.txt -x Info.plist -x 'mate_grammar.[ch]' -
    x ps.c -x svnversion.h -x idl2wrs.sh wireshark-1.4.4/gtk/prefs_capture.c
    stegemanitons/wireshark/gtk/prefs_capture.c
--- wireshark-1.4.4/gtk/prefs_capture.c 2011-03-01 12:03:46.000000000 -0500
+++ stegemanitons/wireshark/gtk/prefs_capture.c 2011-05-10 11:41:14.855251571 -0400
@@ -55,6 +55,12 @@
 #define CAPTURE_REAL_TIME_KEY   "capture_real_time"
 #define AUTO_SCROLL_KEY                 "auto_scroll"
 #define SHOW_INFO_KEY            "show_info"
+#define BLUE_TE1_KEY            "blue_te1"
+#define BLUE_TE2_KEY            "blue_te2"
+#define BLUE_TE3_KEY            "blue_te3"
+#define BLUE_TE4_KEY            "blue_te4"
+#define BLUE_TE5_KEY            "blue_te5"
+#define BLUE_TE6_KEY            "blue_te6"

 #define CAPTURE_TABLE_ROWS 6

@@ -100,12 +106,15 @@ capture_prefs_show(void)
 {
         GtkWidget       *main_tb, *main_vb;
         GtkWidget       *if_cbxe, *if_lb, *promisc_cb, *pcap_ng_cb, *sync_cb, *
             auto_scroll_cb, *show_info_cb;
+        GtkWidget       *blue_te1, *blue_te2, *blue_te3, *blue_te4, *blue_te5, *blue_te6
    , *blue_hb, *blue_lb;
         GtkWidget       *ifopts_lb, *ifopts_bt;
         GList           *if_list, *combo_list;
         int             err;
         int             row = 0;
         GtkTooltips     *tooltips = gtk_tooltips_new();
         gchar           *tooltips_text;
+        gchar **mac_tokens = NULL;
+     int mac_token_len = -1;

         /* Main vertical box */
         main_vb = gtk_vbox_new(FALSE, 7);
@@ -169,6 +178,80 @@ capture_prefs_show(void)
```

```
        gtk_table_attach_defaults(GTK_TABLE(main_tb), ifopts_bt, 1, 2, row, row+1);
        row++;

+       /* Bluetooth master MAC */
+       blue_lb = gtk_label_new("Bluetooth master address:");
+       gtk_table_attach_defaults(GTK_TABLE(main_tb), blue_lb, 0, 1, row, row+1);
+       gtk_misc_set_alignment(GTK_MISC(blue_lb), 1.0f, 0.5f);
+       gtk_widget_show(blue_lb);
+
+       if (prefs.capture_master_mac != NULL) {
+               mac_tokens = g_strsplit(prefs.capture_master_mac, ":", 6);
+               mac_token_len = g_strv_length(mac_tokens);
+       }
+
+       tooltips_text = "Enter the Master MAC Address. If you don't enter something here
    , RAW capture will not be possible.";
+       blue_hb = gtk_hbox_new(TRUE, 3);
+
+       blue_te1 = gtk_entry_new();
+       gtk_entry_set_max_length(blue_te1, 2);
+       gtk_entry_set_width_chars(blue_te1, 2);
+       gtk_tooltips_set_tip(tooltips, blue_te1, tooltips_text, NULL);
+       if (mac_token_len >= 1)
+               gtk_entry_set_text(GTK_ENTRY(blue_te1), mac_tokens[0]);
+       gtk_box_pack_start(GTK_BOX(blue_hb), blue_te1, TRUE, TRUE, 3);
+       g_object_set_data(G_OBJECT(main_vb), BLUE_TE1_KEY, blue_te1);
+
+       blue_te2 = gtk_entry_new();
+       gtk_entry_set_max_length(blue_te2, 2);
+       gtk_entry_set_width_chars(blue_te2, 2);
+       gtk_tooltips_set_tip(tooltips, blue_te2, tooltips_text, NULL);
+       if (mac_token_len >= 2)
+               gtk_entry_set_text(GTK_ENTRY(blue_te2), mac_tokens[1]);
+       gtk_box_pack_start(GTK_BOX(blue_hb), blue_te2, TRUE, TRUE, 3);
+       g_object_set_data(G_OBJECT(main_vb), BLUE_TE2_KEY, blue_te2);
+
+       blue_te3 = gtk_entry_new();
+       gtk_entry_set_max_length(blue_te3, 2);
+       gtk_entry_set_width_chars(blue_te3, 2);
+       gtk_tooltips_set_tip(tooltips, blue_te3, tooltips_text, NULL);
+       if (mac_token_len >= 3)
+               gtk_entry_set_text(GTK_ENTRY(blue_te3), mac_tokens[2]);
+       gtk_box_pack_start(GTK_BOX(blue_hb), blue_te3, TRUE, TRUE, 3);
+       g_object_set_data(G_OBJECT(main_vb), BLUE_TE3_KEY, blue_te3);
+
+       blue_te4 = gtk_entry_new();
+       gtk_entry_set_max_length(blue_te4, 2);
+       gtk_entry_set_width_chars(blue_te4, 2);
+       gtk_tooltips_set_tip(tooltips, blue_te4, tooltips_text, NULL);
+       if (mac_token_len >= 4)
+               gtk_entry_set_text(GTK_ENTRY(blue_te4), mac_tokens[3]);
+       gtk_box_pack_start(GTK_BOX(blue_hb), blue_te4, TRUE, TRUE, 3);
+       g_object_set_data(G_OBJECT(main_vb), BLUE_TE4_KEY, blue_te4);
+
+       blue_te5 = gtk_entry_new();
+       gtk_entry_set_max_length(blue_te5, 2);
+       gtk_entry_set_width_chars(blue_te5, 2);
+       gtk_tooltips_set_tip(tooltips, blue_te5, tooltips_text, NULL);
+       if (mac_token_len >= 5)
+               gtk_entry_set_text(GTK_ENTRY(blue_te5), mac_tokens[4]);
+       gtk_box_pack_start(GTK_BOX(blue_hb), blue_te5, TRUE, TRUE, 3);
+       g_object_set_data(G_OBJECT(main_vb), BLUE_TE5_KEY, blue_te5);
+
+       blue_te6 = gtk_entry_new();
+       gtk_entry_set_max_length(blue_te6, 2);
+       gtk_entry_set_width_chars(blue_te6, 2);
+       gtk_tooltips_set_tip(tooltips, blue_te6, tooltips_text, NULL);
+       if (mac_token_len >= 6)
+               gtk_entry_set_text(GTK_ENTRY(blue_te6), mac_tokens[5]);
+       gtk_box_pack_start(GTK_BOX(blue_hb), blue_te6, TRUE, TRUE, 3);
```

```
+       g_object_set_data(G_OBJECT(main_vb), BLUE_TE6_KEY, blue_te6);
+
+       if (mac_token_len >= 1)
+               g_strfreev(mac_tokens);
+       gtk_table_attach_defaults(GTK_TABLE(main_tb), blue_hb, 1, 2, row, row+1);
+       gtk_widget_show(blue_hb);
+       row++;
+
        /* Promiscuous mode */
        promisc_cb = create_preference_check_button(main_tb, row++,
            "Capture packets in promiscuous mode:",
@@ -218,14 +301,21 @@ void
 capture_prefs_fetch(GtkWidget *w)
 {
        GtkWidget *if_cbxe, *promisc_cb, *pcap_ng_cb, *sync_cb, *auto_scroll_cb, *
           show_info_cb;
-       gchar    *if_text;
+       GtkWidget *blue_te1, *blue_te2, *blue_te3, *blue_te4, *blue_te5, *blue_te6;;
+       gchar    *if_text, *blue_text_1, *blue_text_2, *blue_text_3, *blue_text_4, *
    blue_text_5, *blue_text_6, *blue_text;

        if_cbxe     = (GtkWidget *)g_object_get_data(G_OBJECT(w), DEVICE_KEY);
        promisc_cb = (GtkWidget *)g_object_get_data(G_OBJECT(w), PROM_MODE_KEY);
        pcap_ng_cb = (GtkWidget *)g_object_get_data(G_OBJECT(w), PCAP_NG_KEY);
        sync_cb     = (GtkWidget *)g_object_get_data(G_OBJECT(w), CAPTURE_REAL_TIME_KEY);
        auto_scroll_cb = (GtkWidget *)g_object_get_data(G_OBJECT(w), AUTO_SCROLL_KEY);
-        show_info_cb = (GtkWidget *)g_object_get_data(G_OBJECT(w), SHOW_INFO_KEY);
+       show_info_cb = (GtkWidget *)g_object_get_data(G_OBJECT(w), SHOW_INFO_KEY);
+       blue_te1    = (GtkWidget *)g_object_get_data(G_OBJECT(w), BLUE_TE1_KEY);
+       blue_te2    = (GtkWidget *)g_object_get_data(G_OBJECT(w), BLUE_TE2_KEY);
+       blue_te3    = (GtkWidget *)g_object_get_data(G_OBJECT(w), BLUE_TE3_KEY);
+       blue_te4    = (GtkWidget *)g_object_get_data(G_OBJECT(w), BLUE_TE4_KEY);
+       blue_te5    = (GtkWidget *)g_object_get_data(G_OBJECT(w), BLUE_TE5_KEY);
+       blue_te6    = (GtkWidget *)g_object_get_data(G_OBJECT(w), BLUE_TE6_KEY);

        if (prefs.capture_device != NULL) {
                g_free(prefs.capture_device);
@@ -252,6 +342,42 @@ capture_prefs_fetch(GtkWidget *w)
        prefs.capture_auto_scroll = GTK_TOGGLE_BUTTON (auto_scroll_cb)->active;

        prefs.capture_show_info = !(GTK_TOGGLE_BUTTON (show_info_cb)->active);
+
+       if (prefs.capture_master_mac != NULL) {
+               g_free(prefs.capture_master_mac);
+               prefs.capture_master_mac = NULL;
+       }
+       blue_text_1 = g_strdup(gtk_entry_get_text(GTK_ENTRY(blue_te1)));
+       g_strstrip(blue_text_1);
+       blue_text_2 = g_strdup(gtk_entry_get_text(GTK_ENTRY(blue_te2)));
+       g_strstrip(blue_text_2);
+       blue_text_3 = g_strdup(gtk_entry_get_text(GTK_ENTRY(blue_te3)));
+       g_strstrip(blue_text_3);
+       blue_text_4 = g_strdup(gtk_entry_get_text(GTK_ENTRY(blue_te4)));
+       g_strstrip(blue_text_4);
+       blue_text_5 = g_strdup(gtk_entry_get_text(GTK_ENTRY(blue_te5)));
+       g_strstrip(blue_text_5);
+       blue_text_6 = g_strdup(gtk_entry_get_text(GTK_ENTRY(blue_te6)));
+       g_strstrip(blue_text_6);
+       /* If there was nothing but white space, treat that as an
+          indication that the user doesn't want to wire in a default
+          master MAC */
+       if (*blue_text_1 == '\0' || *blue_text_2 == '\0' || *blue_text_3 == '\0' ||
+               *blue_text_4 == '\0' || *blue_text_5 == '\0' || *blue_text_6 == '\0') {
+               blue_text = NULL;
+       }
+       else {
+           blue_text = g_strdup_printf("%s:%s:%s:%s:%s:%s",
+                   blue_text_1, blue_text_2, blue_text_3,
+                   blue_text_4, blue_text_5, blue_text_6);
+       }
```

```
+       g_free(blue_text_1);
+       g_free(blue_text_2);
+       g_free(blue_text_3);
+       g_free(blue_text_4);
+       g_free(blue_text_5);
+       g_free(blue_text_6);
+       prefs.capture_master_mac = blue_text;
 }

 void
diff -rupN -x .gitignore -x configure -x svn_version.xml -x 'grammar.[ch]' -x '
    dtd_grammar.[ch]' -x '*wslua.[ch]' -x faq.txt -x Info.plist -x 'mate_grammar.[ch]' -
    x ps.c -x svnversion.h -x idl2wrs.sh wireshark-1.4.4/gtk/stock_icons.c stegemanitons
    /wireshark/gtk/stock_icons.c
--- wireshark-1.4.4/gtk/stock_icons.c   2011-03-01 12:03:48.000000000 -0500
+++ stegemanitons/wireshark/gtk/stock_icons.c   2011-05-06 13:58:08.077174186 -0400
@@ -40,6 +40,7 @@
 #include "../image/toolbar/capture_restart_24.xpm"
 #include "../image/toolbar/capture_filter_24.xpm"
 #include "../image/toolbar/capture_details_24.xpm"
+#include "../image/toolbar/bluetooth_mac_24.xpm"
 #endif /* HAVE_LIBPCAP */
 #include "../image/toolbar/display_filter_24.xpm"
 #include "../image/wsicon16.xpm"
@@ -106,6 +107,7 @@ void stock_icons_init(void) {
 #ifdef HAVE_GEOIP
         { WIRESHARK_STOCK_MAP,                  "Map",                  0, 0, NULL },
 #endif
+        { WIRESHARK_STOCK_BLUETOOTH_MAC,        "_BluetoothMAC",        0, 0, NULL },
         { WIRESHARK_STOCK_FOLLOW_STREAM,        "Follow Stream",        0, 0, NULL },
         { WIRESHARK_STOCK_DISPLAY_FILTER,       "_Filter",      0, 0, NULL },
         { WIRESHARK_STOCK_DISPLAY_FILTER_ENTRY, "F_ilter:",  0, 0, NULL },
@@ -174,6 +176,7 @@ void stock_icons_init(void) {
 #ifdef HAVE_GEOIP
         { WIRESHARK_STOCK_MAP,                  internet_24_xpm},
 #endif
+        { WIRESHARK_STOCK_BLUETOOTH_MAC,        bluetooth_mac_24_xpm },
         { WIRESHARK_STOCK_DISPLAY_FILTER,       display_filter_24_xpm },
         { WIRESHARK_STOCK_DISPLAY_FILTER_ENTRY, display_filter_24_xpm },
         { WIRESHARK_STOCK_ABOUT,                wsicon16_xpm },
diff -rupN -x .gitignore -x configure -x svn_version.xml -x 'grammar.[ch]' -x '
    dtd_grammar.[ch]' -x '*wslua.[ch]' -x faq.txt -x Info.plist -x 'mate_grammar.[ch]' -
    x ps.c -x svnversion.h -x idl2wrs.sh wireshark-1.4.4/gtk/stock_icons.h stegemanitons
    /wireshark/gtk/stock_icons.h
--- wireshark-1.4.4/gtk/stock_icons.h   2011-03-01 12:03:49.000000000 -0500
+++ stegemanitons/wireshark/gtk/stock_icons.h   2011-05-06 13:59:33.549038047 -0400
@@ -40,6 +40,7 @@
 #ifdef HAVE_GEOIP
 #define WIRESHARK_STOCK_MAP                     "Wireshark_Stock_Map"
 #endif
+#define WIRESHARK_STOCK_BLUETOOTH_MAC           "Wireshark_Stock_BluetoothMAC"
 #define WIRESHARK_STOCK_FOLLOW_STREAM           "Wireshark_Stock_FollowStream"
 #define WIRESHARK_STOCK_DISPLAY_FILTER          "Wireshark_Stock_DisplayFilter"
 #define WIRESHARK_STOCK_DISPLAY_FILTER_ENTRY    "Wireshark_Stock_DisplayFilter_Entry"
diff -rupN -x .gitignore -x configure -x svn_version.xml -x 'grammar.[ch]' -x '
    dtd_grammar.[ch]' -x '*wslua.[ch]' -x faq.txt -x Info.plist -x 'mate_grammar.[ch]' -
    x ps.c -x svnversion.h -x idl2wrs.sh wireshark-1.4.4/image/toolbar/bluetooth_mac_24.
    xpm stegemanitons/wireshark/image/toolbar/bluetooth_mac_24.xpm
--- wireshark-1.4.4/image/toolbar/bluetooth_mac_24.xpm  1969-12-31 19:00:00.000000000
    -0500
+++ stegemanitons/wireshark/image/toolbar/bluetooth_mac_24.xpm  2011-04-28
    08:26:55.074452442 -0400
@@ -0,0 +1,209 @@
+/* XPM */
+static char * bluetooth_mac_24_xpm[] = {
+"24 24 182 2",
+"       c None",
+".      c #3E4042",
+"+      c #25292C",
+"@      c #1E2124",
```

```
+"#       c #1F2225",
+"$       c #2B2E32",
+"%       c #212120",
+"&       c #86929D",
+"*       c #6F8297",
+"=       c #395571",
+"-       c #3B5673",
+";       c #3B5773",
+">       c #49627C",
+",       c #495B6E",
+"'       c #1F2328",
+")       c #000000",
+"!       c #262D33",
+"~       c #6D8196",
+"{       c #375370",
+"]       c #2D4A69",
+"^       c #425C78",
+"/       c #5C738B",
+"(       c #304D6B",
+"_       c #2E4B6A",
+":       c #516A85",
+"<       c #38495C",
+"[       c #000203",
+"}       c #262D34",
+"|       c #7A8B9E",
+"1       c #47617B",
+"2       c #2F4C6B",
+"3       c #415B77",
+"4       c #FAFBFB",
+"5       c #556C85",
+"6       c #2F4C6A",
+"7       c #314E6C",
+"8       c #2E4C6A",
+"9       c #566C85",
+"0       c #2E3D4E",
+"a       c #010202",
+"b       c #667380",
+"c       c #597089",
+"d       c #3C5774",
+"e       c #F2F4F6",
+"f       c #F4F6F7",
+"g       c #61778E",
+"h       c #324F6D",
+"i       c #465E78",
+"j       c #050606",
+"k       c #74879C",
+"l       c #2C4A68",
+"m       c #3C5874",
+"n       c #FEFEFE",
+"o       c #4C657F",
+"p       c #2F4D6B",
+"q       c #415C78",
+"r       c #172534",
+"s       c #50565B",
+"t       c #4B6580",
+"u       c #48617C",
+"v       c #3F5A76",
+"w       c #A6B2BF",
+"x       c #D4DAE0",
+"y       c #F6F7F8",
+"z       c #60768E",
+"A       c #2C4A69",
+"B       c #2E3E4E",
+"C       c #04070B",
+"D       c #6F8092",
+"E       c #3A5572",
+"F       c #F1F3F5",
+"G       c #CAD1D9",
+"H       c #35516F",
+"I       c #F5F6F8",
```

```
+"J      c #94A3B3",
+"K      c #EBEEF1",
+"L      c #FDFDFD",
+"M      c #627990",
+"N      c #2A445E",
+"O      c #0C141C",
+"P      c #758A9F",
+"Q      c #34516E",
+"R      c #324F6C",
+"S      c #788B9F",
+"T      c #FFFFFF",
+"U      c #CCD3DA",
+"V      c #8C9CAC",
+"W      c #516A83",
+"X      c #EDF0F2",
+"Y      c #E6E9ED",
+"Z      c #4B647F",
+"`      c #2F4B68",
+" .     c #020406",
+"..     c #70859B",
+"+.     c #33506D",
+"@.     c #2B4968",
+"#.     c #7E90A3",
+"$.     c #B1BCC7",
+"%.     c #E9EBEE",
+"&.     c #B5C0CA",
+"*.     c #E9ECF0",
+"=.     c #EAEDF0",
+"-.     c #435D78",
+";.     c #010203",
+">.     c #667C94",
+",.     c #6F8398",
+"'.     c #FFFEFE",
+").     c #FEFFFE",
+"!.     c #E4E8EB",
+"~.     c #2C4968",
+"{.     c #9DAAB9",
+"].     c #49637D",
+"^.     c #586F87",
+"/.     c #C6CED6",
+"(.     c #62788E",
+"_.     c #677C92",
+":.     c #FCFCFD",
+"<.     c #D6DCE2",
+"[.     c #ECEFF1",
+"}.     c #F6F7F9",
+"|.     c #C8D0D7",
+"1.     c #020405",
+"2.     c #4D637A",
+"3.     c #5B728A",
+"4.     c #36526F",
+"5.     c #EFF1F3",
+"6.     c #8D9EAE",
+"7.     c #637990",
+"8.     c #CAD2D9",
+"9.     c #35516E",
+"0.     c #2E4A67",
+"a.     c #101A24",
+"b.     c #415468",
+"c.     c #3D5875",
+"d.     c #FAFAFB",
+"e.     c #E3E7EB",
+"f.     c #435D79",
+"g.     c #3D5874",
+"h.     c #F5F7F8",
+"i.     c #98A6B6",
+"j.     c #DEE2E7",
+"k.     c #73879B",
+"l.     c #2D4662",
+"m.     c #131E2A",
```

```
+"n.     c #395674",
+"o.     c #60778E",
+"p.     c #4F6881",
+"q.     c #F4F6F8",
+"r.     c #91A1B0",
+"s.     c #B4BFCA",
+"t.     c #314F6D",
+"u.     c #1D2C3E",
+"v.     c #0F151B",
+"w.     c #314B67",
+"x.     c #72869A",
+"y.     c #192735",
+"z.     c #334354",
+"A.     c #8193A5",
+"B.     c #283F58",
+"C.     c #0C1219",
+"D.     c #010102",
+"E.     c #4A6684",
+"F.     c #415C77",
+"G.     c #2C4661",
+"H.     c #0F1923",
+"I.     c #14212D",
+"J.     c #425A72",
+"K.     c #34516F",
+"L.     c #8E9EAD",
+"M.     c #314E6D",
+"N.     c #203347",
+"O.     c #090F15",
+"P.     c #415264",
+"Q.     c #445F7B",
+"R.     c #304D6A",
+"S.     c #253B53",
+"T.     c #131B24",
+"U.     c #05080C",
+"V.     c #000102",
+"W.     c #070C11",
+"X.     c #080E13",
+"Y.     c #030508",
+"                                                  ",
+"                    . + @ # $                     ",
+"                  % & * = - ; > , ' )             ",
+"                ! ~ { ] ^ / ( ( _ : < [           ",
+"              } | 1 2 ( 3 4 5 6 7 8 9 0           ",
+"            a b c 6 7 ( d e f g 2 7 h i )         ",
+"            j k l ( ( ( m e n f o p ( q r         ",
+"            s t 8 u v 7 m f w x y z A 6 B C       ",
+"            D E ; F G H m I J ; K L M 6 N O       ",
+"            P Q R S T U 6 F V W X Y Z ( `   .     ",
+"            ..+.7 @.#.n $.%.&.*.=.-.( 7 7 ;.      ",
+"          ) >.8 7 7 6  ,.'.n ).!.m p 7 7 7 ;.     ",
+"          ) >.8 7 7 7 ~.{.T T ].( 7 7 7 7 ;.      ",
+"          ) >.8 7 7 _ ^.I T T /.p 7 7 7 7 ;.      ",
+"            (.6 7 ] _.:.<.[.G }.|.{ ( 7 7 1.      ",
+"            2.p ( 3.:.[.4.5.6.7.f 8.9.( 0.a.     ",
+"            b.p c.d.e.f.g.h.i.~.j.T k.6 l.m.      ",
+"            a n.8 o.p.7 m q.r.s.L #.~.t.u.        ",
+"            v.w.7 6 p ( m e y n x.8 7 ` y.        ",
+"              z.+.( 7 ( ; e n A.8 7 7 B.C.        ",
+"              D.E.7 7 ( F.T k 6 ( 7 G.H.          ",
+"                I.J.K.( u L.( 7 M.6 N.O.          ",
+"                  ) P.Q.M.t.t.R.S.T.U.            ",
+"                    ) V.W.X.W.Y.) )               "};
diff -rupN -x .gitignore -x configure -x svn_version.xml -x 'grammar.[ch]' -x '
    dtd_grammar.[ch]' -x '*wslua.[ch]' -x faq.txt -x Info.plist -x 'mate_grammar.[ch]' -
    x ps.c -x svnversion.h -x idl2wrs.sh wireshark-1.4.4/tshark.c stegemanitons/
    wireshark/tshark.c
--- wireshark-1.4.4/tshark.c    2011-03-01 12:06:09.000000000 -0500
+++ stegemanitons/wireshark/tshark.c    2011-05-11 09:19:46.872361824 -0400
@@ -244,6 +244,7 @@ print_usage(gboolean print_ver)
```

```
    fprintf(output, "  -y <link type>           link layer type (def: first appropriate)\
        n");
    fprintf(output, "  -D                       print list of interfaces and exit\n");
    fprintf(output, "  -L                       print list of link-layer types of iface
        and exit\n");
+   fprintf(output, "  -U <MAC address>         use MAC address as Bluetooth piconet
     master\n");
    fprintf(output, "\n");
    fprintf(output, "Capture stop conditions:\n");
    fprintf(output, "  -c <packet count>        stop after n packets (def: infinite)\n");
@@ -820,11 +821,13 @@ main(int argc, char *argv[])

 #ifdef HAVE_PCAP_CREATE
 #define OPTSTRING_I "I"
+#define OPTSTRING_U "U:"
 #else
 #define OPTSTRING_I ""
+#define OPTSTRING_U ""
 #endif

-#define OPTSTRING "a:b:" OPTSTRING_B "c:C:d:De:E:f:F:G:hi:" OPTSTRING_I "K:lLnN:o:pPqr:
     R:s:St:T:u:vVw:xX:y:z:"
+#define OPTSTRING "a:b:" OPTSTRING_B "c:C:d:De:E:f:F:G:hi:" OPTSTRING_I "K:lLnN:o:pPqr:
     R:s:St:T:u:" OPTSTRING_U "vVw:xX:y:z:"

    static const char    optstring[] = OPTSTRING;

@@ -1055,6 +1058,7 @@ main(int argc, char *argv[])
         case 'p':        /* Don't capture in promiscuous mode */
 #ifdef HAVE_PCAP_CREATE
         case 'I':        /* Capture in monitor mode, if available */
+        case 'U':        /* Set master MAC address */
 #endif
         case 's':        /* Set the snapshot (capture) length */
         case 'w':        /* Write to capture file x */
```

Appendix B: libpcap Source Code Modifications

The following is a patch to libpcap, version 1.1.1.

```
diff -rupN -x configure -x .gitignore libpcap-1.1.1/INSTALL.txt stegemanitons/libpcap/
    INSTALL.txt
--- libpcap-1.1.1/INSTALL.txt   2010-03-11 20:56:53.000000000 -0500
+++ stegemanitons/libpcap/INSTALL.txt   2011-04-19 09:32:59.142199668 -0400
@@ -361,6 +361,8 @@ pcap/sll.h - public definition of DLT_LI
 pcap/usb.h     - public definition of DLT_USB header
 pcap-bpf.c     - BSD Packet Filter support
 pcap-bpf.h     - header for backwards compatibility
+pcap-bt-csr-linux.c    - Bluetooth capture support for Linux using CSR chipsets
+pcap-bt-csr-linux.h    - Bluetooth capture support for Linux usring CSR chipsets
 pcap-bt-linux.c        - Bluetooth capture support for Linux
 pcap-bt-linux.h        - Bluetooth capture support for Linux
 pcap-dag.c     - Endace DAG device capture support
diff -rupN -x configure -x .gitignore libpcap-1.1.1/Makefile.in stegemanitons/libpcap/
    Makefile.in
--- libpcap-1.1.1/Makefile.in   2010-04-05 13:54:05.000000000 -0400
+++ stegemanitons/libpcap/Makefile.in   2011-04-19 09:33:17.032070310 -0400
@@ -267,6 +267,8 @@ EXTRA_DIST = \
        org.tcpdump.chmod_bpf.plist \
        packaging/pcap.spec.in \
        pcap-bpf.c \
+       pcap-bt-csr-linux.c \
+       pcap-bt-csr-linux.h \
        pcap-bt-linux.c \
        pcap-bt-linux.h \
        pcap-can-linux.c \
diff -rupN -x configure -x .gitignore libpcap-1.1.1/configure.in stegemanitons/libpcap/
    configure.in
--- libpcap-1.1.1/configure.in  2010-03-11 20:56:53.000000000 -0500
+++ stegemanitons/libpcap/configure.in  2011-04-19 09:38:54.509630087 -0400
@@ -1378,7 +1378,8 @@ if test "x$enable_bluetooth" != "xno" ;
                AC_CHECK_HEADER(bluetooth/bluetooth.h,
                 [
                   AC_DEFINE(PCAP_SUPPORT_BT, 1, [target host supports Bluetooth sniffing
                       ])
-                  BT_SRC=pcap-bt-linux.c
+                  BT_SRC="pcap-bt-linux.c pcap-bt-csr-linux.c"
+                          LIBS="$LIBS -lbluetooth"
                   AC_MSG_NOTICE(Bluetooth sniffing is supported)
                 ],
                 AC_MSG_NOTICE(Bluetooth sniffing is not supported; install bluez-lib
                     devel to enable it)
diff -rupN -x configure -x .gitignore libpcap-1.1.1/inet.c stegemanitons/libpcap/inet.c
--- libpcap-1.1.1/inet.c        2010-03-11 20:56:54.000000000 -0500
+++ stegemanitons/libpcap/inet.c        2011-04-18 09:07:28.358191928 -0400
@@ -704,7 +704,7 @@ pcap_lookupnet(device, netp, maskp, errb
            || strstr(device, "septel") != NULL
 #endif
 #ifdef PCAP_SUPPORT_BT
-           || strstr(device, "bluetooth") != NULL
+           || strstr(device, "hci") != NULL
 #endif
 #ifdef PCAP_SUPPORT_USB
            || strstr(device, "usbmon") != NULL
diff -rupN -x configure -x .gitignore libpcap-1.1.1/pcap/pcap.h stegemanitons/libpcap/
    pcap/pcap.h
--- libpcap-1.1.1/pcap/pcap.h   2010-03-11 20:56:54.000000000 -0500
+++ stegemanitons/libpcap/pcap/pcap.h   2011-05-11 12:17:08.091325013 -0400
@@ -277,6 +277,7 @@ int pcap_set_rfmon(pcap_t *, int);
 int    pcap_set_timeout(pcap_t *, int);
 int    pcap_set_buffer_size(pcap_t *, int);
 int    pcap_activate(pcap_t *);
+int pcap_set_master_bd_addr(pcap_t *p, unsigned char *mymaster);

 pcap_t *pcap_open_live(const char *, int, int, int, char *);
 pcap_t *pcap_open_dead(int, int);
```

```
diff -rupN -x configure -x .gitignore libpcap-1.1.1/pcap-bt-csr-linux.c stegemanitons/
    libpcap/pcap-bt-csr-linux.c
--- libpcap-1.1.1/pcap-bt-csr-linux.c    1969-12-31 19:00:00.000000000 -0500
+++ stegemanitons/libpcap/pcap-bt-csr-linux.c    2011-05-03 09:39:33.560861969 -0400
@@ -0,0 +1,115 @@
+#ifndef lint
+static const char rcsid[] _U_ =
+    "@(#) $Header: /frontline/frontline/frontline.c,v 1.1.1.1 2007/08/05 19:11:27
    abittau Exp $ (LBL)";
+#endif
+
+#ifdef HAVE_CONFIG_H
+#include "config.h"
+#endif
+
+#include "pcap-bt-csr-linux.h"
+
+#include <bluetooth/bluetooth.h>
+#include <bluetooth/hci.h>
+#include <bluetooth/hci_lib.h>
+#include <stdlib.h>
+
+int
+send_debug(struct csr_state *s, struct dbg_packet *dp, void *rp, int rplen)
+{
+    unsigned char cp[254];
+    struct hci_request rq;
+    unsigned char *p = cp;
+
+    memset(&rq, 0, sizeof(rq));
+    memset(cp, 0, sizeof(cp));
+
+    /* payload descriptor */
+    *p++ = FRAG_FIRST | FRAG_LAST | CHAN_DEBUG;
+    memcpy(p, dp, sizeof(*dp));
+    p += sizeof(*dp);
+
+    rq.ogf = OGF_VENDOR_CMD;
+    rq.ocf = 0x00;
+    rq.event = EVT_VENDOR;
+    rq.cparam = cp;
+    rq.clen = p - cp;
+    rq.rparam = rp;
+    rq.rlen = rplen;
+
+    if (hci_send_req(s->s_fd, &rq, 2000) < 0)
+        return 1;
+
+    return 0;
+}
+
+int
+send_debug_no_rp(struct csr_state *s, struct dbg_packet *dp)
+{
+    unsigned char rp[254];
+
+    return send_debug(s, dp, rp, sizeof(rp));
+}
+
+unsigned int
+get_timer(struct csr_state *s)
+{
+    unsigned char rp[254];
+    struct dbg_packet pkt;
+
+    memset(rp, 0, sizeof(rp));
+    memset(&pkt, 0, sizeof(pkt));
+
+    pkt.dp_type = CMD_TIMER;
+
```

```
+    if (send_debug(s, &pkt, rp, sizeof(rp)))
+        return 1;
+
+    return *((unsigned int *) &rp[2]);
+}
+
+int
+set_csr_filter(struct csr_state *s, unsigned char val)
+{
+    struct dbg_packet pkt;
+
+    memset(&pkt, 0, sizeof(pkt));
+
+    pkt.dp_type = CMD_FILTER;
+    pkt.dp_data[0] = val;
+
+    return send_debug_no_rp(s, &pkt);
+}
+
+int
+sniff_stop(struct csr_state *s)
+{
+    struct dbg_packet pkt;
+
+    memset(&pkt, 0, sizeof(pkt));
+
+    pkt.dp_type = CMD_STOP;
+
+    return send_debug_no_rp(s, &pkt);
+}
+
+int
+sniff_start(struct csr_state *s)
+{
+    struct dbg_packet pkt;
+    struct start_packet *sp = (struct start_packet *) &pkt.dp_data;
+    unsigned char *master = s->s_master_mac;
+    unsigned char *slave = s->s_slave_mac;
+    int i;
+
+    memset(&pkt, 0, sizeof(pkt));
+    pkt.dp_type = CMD_START;
+
+    for (i = 5; i >= 0; i--)
+        sp->sp_master_rev[i] = *master++;
+
+    for (i = 5; i >= 0; i--)
+        sp->sp_slave_rev[i] = *slave++;
+
+    return send_debug_no_rp(s, &pkt);
+}
diff -rupN -x configure -x .gitignore libpcap-1.1.1/pcap-bt-csr-linux.h stegemanitons/
    libpcap/pcap-bt-csr-linux.h
--- libpcap-1.1.1/pcap-bt-csr-linux.h   1969-12-31 19:00:00.000000000 -0500
+++ stegemanitons/libpcap/pcap-bt-csr-linux.h   2011-05-11 12:17:38.349522001 -0400
@@ -0,0 +1,44 @@
+#ifndef _PCAP_BT_CSR_LINUX_H_
+#define _PCAP_BT_CSR_LINUX_H_
+
+#include <stdint.h>
+#include <sys/types.h>
+
+#define __packed __attribute__((__packed__))
+
+#define FRAG_FIRST (1 << 6)
+#define FRAG_LAST  (1 << 7)
+#define CHAN_DEBUG 20
+
+#define CMD_START  0x30
+#define CMD_STOP   0x32
```

```
+#define CMD_FILTER 0x33
+#define CMD_TIMER  0x34
+
+struct dbg_packet {
+    uint8_t  dp_type;
+    uint16_t dp_unknown1;
+    uint16_t dp_unknown2;
+    uint8_t  dp_data[19];
+} __packed;
+
+struct start_packet {
+    uint8_t  sp_master_rev[6];
+    uint32_t sp_unknown;
+    uint8_t  sp_slave_rev[6];
+} __packed;
+
+struct csr_state {
+    int      s_fd;
+    u_char   s_master_mac[6];
+    u_char   s_slave_mac[6];
+} _state;
+
+int send_debug(struct csr_state *s, struct dbg_packet *dp, void *rp, int rplen);
+int send_debug_no_rp(struct csr_state *s, struct dbg_packet *dp);
+unsigned int get_timer(struct csr_state *s);
+int set_csr_filter(struct csr_state *s, unsigned char val);
+int sniff_stop(struct csr_state *s);
+int sniff_start(struct csr_state *s);
+
+#endif
diff -rupN -x configure -x .gitignore libpcap-1.1.1/pcap-bt-linux.c stegemanitons/
    libpcap/pcap-bt-linux.c
--- libpcap-1.1.1/pcap-bt-linux.c       2010-03-11 20:56:54.000000000 -0500
+++ stegemanitons/libpcap/pcap-bt-linux.c       2011-05-13 08:13:10.620584275 -0400
@@ -11,8 +11,8 @@
  * 2. Redistributions in binary form must reproduce the above copyright
  * notice, this list of conditions and the following disclaimer in the
  * documentation and/or other materials provided with the distribution.
- * 3. The name of the author may not be used to endorse or promote
- * products derived from this software without specific prior written
+ * 3. The name of the author may not be used to endorse or promote
+ * products derived from this software without specific prior written
  * permission.
  *
  * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
@@ -35,13 +35,14 @@
 static const char rcsid[] _U_ =
     "@(#) $Header: /tcpdump/master/libpcap/pcap-bt-linux.c,v 1.15 2008-07-01 07:05:54
         guy Exp $ (LBL)";
 #endif
-
+
 #ifdef HAVE_CONFIG_H
 #include "config.h"
 #endif

 #include "pcap-int.h"
 #include "pcap-bt-linux.h"
+#include "pcap-bt-csr-linux.h"
 #include "pcap/bluetooth.h"

 #ifdef NEED_STRERROR_H
@@ -59,8 +60,9 @@ static const char rcsid[] _U_ =

 #include <bluetooth/bluetooth.h>
 #include <bluetooth/hci.h>
+#include <bluetooth/hci_lib.h>

-#define BT_IFACE "bluetooth"
+#define BT_IFACE "hci"
```

```
 #define BT_CTRL_SIZE 128

 /* forward declaration */
@@ -70,67 +72,79 @@ static int bt_inject_linux(pcap_t *, con
 static int bt_setfilter_linux(pcap_t *, struct bpf_program *);
 static int bt_setdirection_linux(pcap_t *, pcap_direction_t);
 static int bt_stats_linux(pcap_t *, struct pcap_stat *);
+static int bt_is_raw(int dev_id);

-int
-bt_platform_finddevs(pcap_if_t **alldevsp, char *err_str)
+int
+bt_platform_finddevs(pcap_if_t **alldevsp, char *errbuf)
 {
-	pcap_if_t *found_dev = *alldevsp;
-	struct hci_dev_list_req *dev_list;
-	struct hci_dev_req *dev_req;
-	int i, sock;
+	struct hci_dev_list_req *dl;
+	struct hci_dev_req *dr;
+	int i, sk;
+	struct hci_dev_info di;
+	int dd;
+	char dev_name[20], dev_descr[40];
 	int ret = 0;
-
-	sock  = socket(AF_BLUETOOTH, SOCK_RAW, BTPROTO_HCI);
-	if (sock < 0)
+
+	sk = socket(AF_BLUETOOTH, SOCK_RAW, BTPROTO_HCI);
+	if (sk < 0)
 	{
-		/* if bluetooth is not supported this this is not fatal*/
+		/* if bluetooth is not supported this is not fatal*/
 		if (errno == EAFNOSUPPORT)
 			return 0;
-		snprintf(err_str, PCAP_ERRBUF_SIZE, "Can't open raw Bluetooth socket %d
    :%s",
+		snprintf(errbuf, PCAP_ERRBUF_SIZE, "Can't open raw Bluetooth socket %d:%
    s",
 			errno, strerror(errno));
 		return -1;
 	}

-	dev_list = malloc(HCI_MAX_DEV * sizeof(*dev_req) + sizeof(*dev_list));
-	if (!dev_list)
+	dl = malloc(HCI_MAX_DEV * sizeof(*dr) + sizeof(*dl));
+	if (!dl)
 	{
-		snprintf(err_str, PCAP_ERRBUF_SIZE, "Can't allocate %zu bytes for
    Bluetooth device list",
-			HCI_MAX_DEV * sizeof(*dev_req) + sizeof(*dev_list));
+		snprintf(errbuf, PCAP_ERRBUF_SIZE, "Can't allocate %zu bytes for
    Bluetooth device list",
+			HCI_MAX_DEV * sizeof(*dr) + sizeof(*dl));
 		ret = -1;
 		goto done;
 	}

-	dev_list->dev_num = HCI_MAX_DEV;
+	memset(dl, 0, HCI_MAX_DEV * sizeof(*dr) + sizeof(*dl));
+	dl->dev_num = HCI_MAX_DEV;

-	if (ioctl(sock, HCIGETDEVLIST, (void *) dev_list) < 0)
+	if (ioctl(sk, HCIGETDEVLIST, dl) < 0)
 	{
-		snprintf(err_str, PCAP_ERRBUF_SIZE, "Can't get Bluetooth device list via
    ioctl %d:%s",
+		snprintf(errbuf, PCAP_ERRBUF_SIZE, "Can't get Bluetooth device list via
    ioctl %d:%s",
```

```
                        errno, strerror(errno));
                ret = -1;
                goto free;
        }

-       dev_req = dev_list->dev_req;
-       for (i = 0; i < dev_list->dev_num; i++, dev_req++) {
-               char dev_name[20], dev_descr[30];
-
-               snprintf(dev_name, 20, BT_IFACE"%d", dev_req->dev_id);
-               snprintf(dev_descr, 30, "Bluetooth adapter number %d", i);
-
-               if (pcap_add_if(&found_dev, dev_name, 0,
-                       dev_descr, err_str) < 0)
-               {
+       dr = dl->dev_req;
+       for (i = 0; i < dl->dev_num; i++, dr++) {
+               if (hci_devinfo(dr->dev_id, &di) < 0)
+                       continue;
+
+               if (!hci_test_bit(HCI_UP, &di.flags))
+                       continue;
+
+               dd = hci_open_dev(di.dev_id);
+               if (dd < 0)
+                       continue;
+               hci_close_dev(dd);
+
+               snprintf(dev_name, 20, BT_IFACE"%d", dr->dev_id);
+               if (bt_is_raw(dr->dev_id))
+                       snprintf(dev_descr, 40, "RAW Bluetooth adapter number %d", i);
+               else
+                       snprintf(dev_descr, 40, "Bluetooth adapter number %d", i);
+
+               if (pcap_add_if(alldevsp, dev_name, 0, dev_descr, errbuf) < 0) {
                        ret = -1;
                        break;
                }
-
        }

 free:
-       free(dev_list);
-
+       free(dl);
 done:
-       close(sock);
+       close(sk);
        return ret;
 }

@@ -148,27 +162,29 @@ bt_create(const char *device, char *ebuf
 }

 static int
-bt_activate(pcap_t* handle)
+bt_activate(pcap_t *handle)
 {
        struct sockaddr_hci addr;
        int opt;
-       int             dev_id;
-       struct hci_filter       flt;
+       int dev_id;
+       struct hci_filter flt;
        int err = PCAP_ERROR;
+       int i;

        /* get bt interface id */
        if (sscanf(handle->opt.source, BT_IFACE"%d", &dev_id) != 1)
        {
```

```
                snprintf(handle->errbuf, PCAP_ERRBUF_SIZE,
-                       "Can't get Bluetooth device index from %s",
+                       "Can't get Bluetooth device index from %s",
                         handle->opt.source);
                return PCAP_ERROR;
        }

        /* Initialize some components of the pcap structure. */
-       handle->bufsize = handle->snapshot+BT_CTRL_SIZE+sizeof(pcap_bluetooth_h4_header)
    ;
+       handle->bufsize = handle->snapshot + BT_CTRL_SIZE + sizeof(
    pcap_bluetooth_h4_header);
        handle->offset = BT_CTRL_SIZE;
        handle->linktype = DLT_BLUETOOTH_HCI_H4_WITH_PHDR;
+       handle->is_raw = 0;

        handle->read_op = bt_read_linux;
        handle->inject_op = bt_inject_linux;
@@ -179,9 +195,9 @@ bt_activate(pcap_t* handle)
        handle->setnonblock_op = pcap_setnonblock_fd;
        handle->stats_op = bt_stats_linux;
        handle->md.ifindex = dev_id;
-
+
        /* Create HCI socket */
-       handle->fd = socket(AF_BLUETOOTH, SOCK_RAW, BTPROTO_HCI);
+       handle->fd = hci_open_dev(dev_id);
        if (handle->fd < 0) {
                snprintf(handle->errbuf, PCAP_ERRBUF_SIZE, "Can't create raw socket %d:%
                    s",
                         errno, strerror(errno));
@@ -209,27 +225,17 @@ bt_activate(pcap_t* handle)
                goto close_fail;
        }

-       /* Setup filter, do not call hci function to avoid dependence on
-        * external libs        */
+       /* Setup filter */
        memset(&flt, 0, sizeof(flt));
-       memset((void *) &flt.type_mask, 0xff, sizeof(flt.type_mask));
-       memset((void *) &flt.event_mask, 0xff, sizeof(flt.event_mask));
+       hci_filter_clear(&flt);
+       hci_filter_all_ptypes(&flt);
+       hci_filter_all_events(&flt);
        if (setsockopt(handle->fd, SOL_HCI, HCI_FILTER, &flt, sizeof(flt)) < 0) {
                snprintf(handle->errbuf, PCAP_ERRBUF_SIZE, "Can't set filter %d:%s",
                         errno, strerror(errno));
                goto close_fail;
        }

-
-       /* Bind socket to the HCI device */
-       addr.hci_family = AF_BLUETOOTH;
-       addr.hci_dev = handle->md.ifindex;
-       if (bind(handle->fd, (struct sockaddr *) &addr, sizeof(addr)) < 0) {
-               snprintf(handle->errbuf, PCAP_ERRBUF_SIZE, "Can't attach to device %d %d
    :%s",
-                        handle->md.ifindex, errno, strerror(errno));
-               goto close_fail;
-       }
-
        if (handle->opt.rfmon) {
                /*
                 * Monitor mode doesn't apply to Bluetooth devices.
@@ -252,6 +258,25 @@ bt_activate(pcap_t* handle)
        }

        handle->selectable_fd = handle->fd;
+
+       if (bt_is_raw(handle->md.ifindex)) {
```

```
+               /* set slave MAC address to 00:00:00:00:00:00 */
+               memset(&handle->csr_state.s_slave_mac, 0, 6);
+
+               handle->csr_state.s_fd = handle->fd;
+
+               /* call some frontline stuff */
+               if (set_csr_filter(&handle->csr_state, 7)) {
+                       snprintf(handle->errbuf, PCAP_ERRBUF_SIZE, "Failed to set CSR
    filter");
+                       goto close_fail;
+               }
+               if (sniff_start(&handle->csr_state)) {
+                       snprintf(handle->errbuf, PCAP_ERRBUF_SIZE, "Failed to start CSR
    sniffing");
+                       goto close_fail;
+               }
+               handle->is_raw = 1;
+       }
+
        return 0;

 close_fail:
@@ -264,14 +289,14 @@ bt_read_linux(pcap_t *handle, int max_pa
 {
        struct cmsghdr *cmsg;
        struct msghdr msg;
-       struct iovec  iv;
+       struct iovec iv;
        struct pcap_pkthdr pkth;
-       pcap_bluetooth_h4_header* bthdr;
+       pcap_bluetooth_h4_header *bthdr;
+
+       bthdr = (pcap_bluetooth_h4_header *) &handle->buffer[handle->offset];
+       iv.iov_base = &handle->buffer[handle->offset + sizeof(pcap_bluetooth_h4_header)
    ];
+       iv.iov_len = handle->snapshot;

-       bthdr = (pcap_bluetooth_h4_header*) &handle->buffer[handle->offset];
-       iv.iov_base = &handle->buffer[handle->offset+sizeof(pcap_bluetooth_h4_header)];
-       iv.iov_len  = handle->snapshot;
-
        memset(&msg, 0, sizeof(msg));
        msg.msg_iov = &iv;
        msg.msg_iovlen = 1;
@@ -288,14 +313,14 @@ bt_read_linux(pcap_t *handle, int max_pa
                }
        } while ((pkth.caplen == -1) && (errno == EINTR));

-
+
        if (pkth.caplen < 0) {
                snprintf(handle->errbuf, PCAP_ERRBUF_SIZE, "Can't receive packet %d:%s",
                        errno, strerror(errno));
                return -1;
        }

-       /* get direction and timestamp*/
+       /* get direction and timestamp*/
        cmsg = CMSG_FIRSTHDR(&msg);
        int in=0;
        while (cmsg) {
@@ -303,18 +328,18 @@ bt_read_linux(pcap_t *handle, int max_pa
                        case HCI_CMSG_DIR:
                                in = *((int *) CMSG_DATA(cmsg));
                                break;
-                       case HCI_CMSG_TSTAMP:
+                       case HCI_CMSG_TSTAMP:
                                pkth.ts = *((struct timeval *) CMSG_DATA(cmsg));
                                break;
                }
```

```
                cmsg = CMSG_NXTHDR(&msg, cmsg);
        }
-       if ((in && (handle->direction == PCAP_D_OUT)) ||
+       if ((in && (handle->direction == PCAP_D_OUT)) ||
                                ((!in) && (handle->direction == PCAP_D_IN)))
                return 0;

        bthdr->direction = htonl(in != 0);
-       pkth.caplen+=sizeof(pcap_bluetooth_h4_header);
+       pkth.caplen += sizeof(pcap_bluetooth_h4_header);
        pkth.len = pkth.caplen;
        callback(user, &pkth, &handle->buffer[handle->offset]);
        return 1;
@@ -326,47 +351,54 @@ bt_inject_linux(pcap_t *handle, const vo
        snprintf(handle->errbuf, PCAP_ERRBUF_SIZE, "inject not supported on "
                "bluetooth devices");
        return (-1);
-}
-
+}

-static int
+static int
 bt_stats_linux(pcap_t *handle, struct pcap_stat *stats)
 {
        int ret;
        struct hci_dev_info dev_info;
-       struct hci_dev_stats * s = &dev_info.stat;
+       struct hci_dev_stats *s = &dev_info.stat;
        dev_info.dev_id = handle->md.ifindex;
-
+
        /* ingnore eintr */
        do {
                ret = ioctl(handle->fd, HCIGETDEVINFO, (void *)&dev_info);
        } while ((ret == -1) && (errno == EINTR));
-
+
        if (ret < 0) {
                snprintf(handle->errbuf, PCAP_ERRBUF_SIZE, "can get stats"
                        " via ioctl %d:%s", errno, strerror(errno));
                return (-1);
-
        }

-       /* we receive both rx and tx frames, so comulate all stats */
-       stats->ps_recv = s->evt_rx + s->acl_rx + s->sco_rx + s->cmd_tx +
+       /* we receive both rx and tx frames, so comulate all stats */
+       stats->ps_recv = s->evt_rx + s->acl_rx + s->sco_rx + s->cmd_tx +
                s->acl_tx +s->sco_tx;
        stats->ps_drop = s->err_rx + s->err_tx;
        stats->ps_ifdrop = 0;
        return 0;
 }

-static int
+static int
 bt_setfilter_linux(pcap_t *p, struct bpf_program *fp)
 {
        return 0;
 }

-
-static int
+static int
 bt_setdirection_linux(pcap_t *p, pcap_direction_t d)
 {
        p->direction = d;
        return 0;
 }
```

```
+
+static int
+bt_is_raw(int dev_id)
+{
+       struct hci_dev_info di;
+       if (hci_devinfo(dev_id, &di) < 0)
+               return 0;
+
+       return hci_test_bit(HCI_RAW, &di.flags);
+}
diff -rupN -x configure -x .gitignore libpcap-1.1.1/pcap-int.h stegemanitons/libpcap/
    pcap-int.h
--- libpcap-1.1.1/pcap-int.h    2010-03-11 20:56:54.000000000 -0500
+++ stegemanitons/libpcap/pcap-int.h    2011-05-13 08:13:10.620584275 -0400
@@ -60,6 +60,10 @@ extern CRITICAL_SECTION g_PcapCompileCri
 #include <snf.h>
 #endif

+#ifdef PCAP_SUPPORT_BT
+#include "pcap-bt-csr-linux.h"
+#endif
+
 #if (defined(_MSC_VER) && (_MSC_VER <= 1200)) /* we are compiling with Visual Studio 6,
      that doesn't support the LL suffix*/

 /*
@@ -269,6 +273,11 @@ struct pcap {
        int fddipad;
 #endif

+#ifdef PCAP_SUPPORT_BT
+       struct csr_state csr_state;
+       u_char is_raw;
+#endif
+
 #ifdef MSDOS
         void (*wait_proc)(void); /*           call proc while waiting */
 #endif
diff -rupN -x configure -x .gitignore libpcap-1.1.1/pcap-linux.c stegemanitons/libpcap/
    pcap-linux.c
--- libpcap-1.1.1/pcap-linux.c  2010-03-11 20:56:54.000000000 -0500
+++ stegemanitons/libpcap/pcap-linux.c  2011-05-11 12:20:24.413117101 -0400
@@ -180,6 +180,7 @@ static const char rcsid[] _U_ =

 #ifdef PCAP_SUPPORT_BT
 #include "pcap-bt-linux.h"
+#include "pcap-bt-csr-linux.h"
 #endif

 #ifdef PCAP_SUPPORT_CAN
@@ -388,7 +389,7 @@ pcap_create(const char *device, char *eb
 #endif /* HAVE_SNF_API */

 #ifdef PCAP_SUPPORT_BT
-       if (strstr(device, "bluetooth")) {
+       if (strstr(device, "hci")) {
                return bt_create(device, ebuf);
        }
 #endif
@@ -1099,6 +1100,12 @@ static void      pcap_cleanup_linux( pcap_t *
                free(handle->md.device);
                handle->md.device = NULL;
        }
+
+#ifdef PCAP_SUPPORT_BT
+       if (handle->linktype == DLT_BLUETOOTH_HCI_H4_WITH_PHDR && handle->is_raw)
+               sniff_stop(&handle->csr_state);
+#endif
+
        pcap_cleanup_live_common(handle);
```

```
 }

diff -rupN -x configure -x .gitignore libpcap-1.1.1/pcap.c stegemanitons/libpcap/pcap.c
--- libpcap-1.1.1/pcap.c        2010-03-11 20:56:54.000000000 -0500
+++ stegemanitons/libpcap/pcap.c        2011-05-13 08:13:10.620584275 -0400
@@ -258,6 +258,7 @@ pcap_create_common(const char *source, c
        pcap_set_snaplen(p, 65535);     /* max packet size */
        p->opt.promisc = 0;
        p->opt.buffer_size = 0;
+       memset(&p->csr_state, 0, sizeof(p->csr_state));
        return (p);
 }

@@ -290,6 +291,20 @@ pcap_set_promisc(pcap_t *p, int promisc)
        return 0;
 }

+int
+pcap_set_master_bd_addr(pcap_t *p, unsigned char *mymaster)
+{
+       int i;
+
+       if (pcap_check_activated(p))
+               return PCAP_ERROR_ACTIVATED;
+
+       for (i = 0; i < 6; ++i)
+               p->csr_state.s_master_mac[i] = mymaster[i];
+
+       return 0;
+}
+
 int
 pcap_set_rfmon(pcap_t *p, int rfmon)
 {
```

Appendix C: DAVE_CONFIG

The following is the DAVE_CONFIG script that compiles the custom libpcap and Wireshark builds above.

```
#!/bin/bash

# First, make sure we're on Linux, as that is all that's supported right now
unamestr=`uname`
if [ "$unamestr" != "Linux" ]; then
    echo "BloosCloos is only supported on Linux at this time."
    exit
fi

# Define some variables we'll need
numcpu=`grep -c 'model name' /proc/cpuinfo`
numcpu=$(($numcpu + 1))
distro=0
pkg_list=()
clean=0
hcidev=""
silent=0
rundfu=0
uninstall=0

# Print the given message and exit with status 1
function die {
    echo "$1"
    exit 1
}

# Print usage and exit with provided status
function usage {
    cat >&2 <<END
DAVE_CONFIG builds and installs BloosCloos to your system

Usage: ./DAVE_CONFIG [options]

Options:
    -c          Clean libpcap and Wireshark builds, then rebuild
    -C          Clean libpcap and Wireshark builds, then exit
    -h          Print this message and exit
    -i <dev>    Attempt to run firmware upgrade on specified Bluetooth device
    -s          Silent operation, do not print 'configure' and 'make' output
    -u          Uninstall libpcap and Wireshark from system

END

    exit $1
}

# Clean both libpcap and Wireshark, then exit
function clean_and_quit {
    echo -e "Cleaning libpcap and Wireshark...\n"
    cd libpcap-1.1.1

    if [ $silent -ne 0 ]; then
        make -s distclean
    else
        make distclean
    fi

    cd ../wireshark-1.4.4

    if [ $silent -ne 0 ]; then
        # Wireshark doesn't obey -s option for make
        make distclean > /dev/null
    else
        make distclean
```

```
    fi

    cd ..
    exit 0
}

# Uninstall both libpcap and Wireshark from system, exit if successful
function uninstall_and_quit {
    # if we have both makefiles, uninstall. otherwise, we'll do it later.
    if [[ -r libpcap-1.1.1/Makefile && -r wireshark-1.4.4/Makefile ]]; then
        cd libpcap-1.1.1
        if [ $silent -ne 0 ]; then
            sudo -k -p "Please enter your password to uninstall libpcap: " \
                make -s uninstall
        else
            sudo -k -p "Please enter your password to uninstall libpcap: " \
                make uninstall
        fi
        cd ../wireshark-1.4.4
        echo
        if [ $silent -ne 0 ]; then
            # Wireshark doesn't obey -s option for make
            sudo -k -p "Please enter your password to uninstall Wireshark: " \
                make uninstall > /dev/null
        else
            sudo -k -p "Please enter your password to uninstall Wireshark: " \
                make uninstall
        fi

        cd ..
        exit 0
    fi
}

# Try to install firmware to given Bluetooth device
function flash_firmware {
    if [ -n "$hcidev" ]; then
        sudo -k -p "Please enter your password to flash new Bluetooth firmware: " \
            bccmd psget -s 0x0000 0x02be
        sudo bccmd psset -s 0x0000 0x02be 0x0a12
        sudo bccmd psget -s 0x0000 0x02be
        sudo bccmd psset -s 0x0000 0x02bf 0x0002
        sudo bccmd psget -s 0x0000 0x02bf
        sudo hciconfig "$hcidev" down
        sudo dfutool archive firmware/old_firmware.dfu
        sudo dfutool upgrade firmware/airsnifferdev46bc4.dfu
        if [ $? -ne 0 ]; then
            sudo hciconfig "$hcidev" reset
            sudo dfutool upgrade firmware/airsnifferdev46bc4.dfu
            if [ $? -ne 0 ]; then
                die "Could not upgrade firmware"
            fi
        fi
        sudo hciconfig "$hcidev" up
        echo -e "Finished upgrading Bluetooth Dongle.  :)\n"
    else
        die "No interface provided"
    fi

    exit 0
}

# Check our command line options
while getopts ":cChi:su" opt; do
    case $opt in
        c)
            clean=1
            ;;
        C)
            clean=2
```

```
            ;;
        h)
            usage 0
            ;;
        i)
            hcidev="$OPTARG"
            rundfu=1
            ;;
        s)
            silent=1
            ;;
        u)
            uninstall=1
            ;;
        \?)
            echo -e "Invalid option: -$OPTARG\n" >&2
            usage 1
            ;;
        :)
            echo -e "Option -$OPTARG requires an argument.\n" >&2
            usage 1
            ;;
    esac
done

# Let's try to check for some packages we need
#
# Check for an Arch system first
which pacman > /dev/null 2>&1
if [ $? -eq 0 ]; then
    distro=1

    pkgs=('gcc' 'flex' 'bison' 'bluez' 'openssl' 'libcap' 'gnutls' 'gtk2' 'make'
          'pkg-config' 'python2' 'autoconf' 'automake' 'm4' 'libtool')

    for pkg in "${pkgs[@]}"
    do
        pacman -Q "$pkg" > /dev/null 2>&1
        if [ $? -ne 0 ]; then
            pkg_list=("${pkg_list[@]}" "$pkg")
        fi
    done

    if [ ${#pkg_list[@]} -ne 0 ]; then
        echo "We need to install the following:"
        for pkg in "${pkg_list[@]}"
        do
            echo "    $pkg"
        done
        echo
        sudo -k -p "Please enter your password to install dependencies: " \
            pacman -S "${pkg_list[@]}"
    fi
fi

# Check for Debian/Ubuntu system
which apt-get > /dev/null 2>&1
if [ $? -eq 0 -a $distro -eq 0 ]; then
    distro=1

    pkgs=('gcc' 'flex' 'bison' 'make' 'libbluetooth3' 'libbluetooth-dev'
          'libgtk2.0-dev' 'libssl-dev' 'libgnutls-dev' 'libcap-dev' 'python'
          'pkg-config' 'autoconf' 'automake' 'm4' 'libtool')

    for pkg in "${pkgs[@]}"
    do
        dpkg -s "$pkg" > /dev/null 2>&1
        if [ $? -ne 0 ]; then
            pkg_list=("${pkg_list[@]}" "$pkg")
        fi
```

```
    done

    if [ ${#pkg_list[@]} -ne 0 ]; then
        echo "We need to install the following:"
        for pkg in "${pkg_list[@]}"
        do
            echo "    $pkg"
        done
        echo
        sudo -k -p "Please enter your password to install dependencies: " \
            apt-get install "${pkg_list[@]}"
    fi
fi

# check for Red Hat system
which yum > /dev/null 2>&1
if [ $? -eq 0 -a $distro -eq 0 ]; then
    distro=1

    pkgs=('pkgconfig' 'gcc' 'flex' 'bison' 'make' 'python' 'libcap-devel'
          'bluez-libs-devel' 'bluez-libs' 'openssl-devel' 'gnutls-devel'
          'gtk2-devel' 'autoconf' 'automake' 'm4' 'libtool')

    for pkg in "${pkgs[@]}"
    do
        rpm -q "$pkg" > /dev/null 2>&1
        if [ $? -ne 0 ]; then
            pkg_list=("${pkg_list[@]}" "$pkg")
        fi
    done

    if [ ${#pkg_list[@]} -ne 0 ]; then
        echo "We need to install the following:"
        for pkg in "${pkg_list[@]}"
        do
            echo "    $pkg"
        done
        echo
        sudo -k -p "Please enter your password to install dependencies: " \
            yum install "${pkg_list[@]}"
    fi
fi

if [ $distro -eq 0 ]; then
    echo -e "Could not determine your Linux distro, trying to build anyway...\n"
fi

# Check for the right version of Python
python -V 2>&1 | grep -q 'Python 2'
if [ $? -ne 0 ]; then
    my_py=""

    py_opts=('python2' 'python2.7' 'python2.6' 'python2.5' 'python2.4' 'python2.3')
    for py in "${py_opts[@]}"
    do
        which "$py" > /dev/null 2>&1
        if [ $? -eq 0 ]; then
            my_py="$py"
            break
        fi
    done

    if [ -n "$my_py" ]; then
        for file in `find wireshark/ -name '*.py'`
        do
            sed -i -e "s_/usr/bin/env\spython_/usr/bin/env ${my_py}_g" "$file"
        done
        sed -i -e "s_/usr/bin/env\spython_/usr/bin/env ${my_py}_g" wireshark/tools/
            msnchat
    else
```

```
        echo -e "Could not determine your Python version, trying to build anyway...\n"
    fi
fi

if [ $uninstall -ne 0 ]; then
    uninstall_and_quit
fi

if [ $clean -eq 2 ]; then
    clean_and_quit
fi

if [ $rundfu -ne 0 ]; then
    flash_firmware
fi

# Build and install libpcap
cd libpcap-1.1.1
export CFLAGS="-march=native -O2 -pipe"
export CXXFLAGS="${CFLAGS}"
export LDFLAGS="-Wl,--hash-style=gnu -Wl,--as-needed"
export MAKEFLAGS="-j$numcpu"

if [ $clean -ne 0 ]; then
    echo -e "\nCleaning libpcap build...\n"
    if [ $silent -ne 0 ]; then
        make -s distclean
    else
        make distclean
    fi
fi

echo -e "Building libpcap...\n"
if [ $silent -ne 0 ]; then
    autoconf > /dev/null
    ./configure --prefix=/usr --enable-ipv6 > /dev/null
else
    autoconf
    ./configure --prefix=/usr --enable-ipv6
fi
if [ $? -ne 0 ]; then
    die "Failed to properly configure libpcap, exiting"
fi

# check if we still need to uninstall
if [ $uninstall -ne 0 ]; then
    if [ $silent -ne 0 ]; then
        sudo -k -p "Please enter your password to uninstall libpcap: " \
            make -s uninstall
    else
        sudo -k -p "Please enter your password to uninstall libpcap: " \
            make uninstall
    fi
else
    if [ $silent -ne 0 ]; then
        make -s
    else
        make
    fi
    if [ $? -ne 0 ]; then
        die "Failed to build libpcap, exiting"
    fi
    if [ $silent -ne 0 ]; then
        sudo -k -p "Please enter your password to install libpcap: " make -s install
    else
        sudo -k -p "Please enter your password to install libpcap: " make install
    fi
fi

# Build and install Wireshark
```

```
cd ../wireshark-1.4.4
if [ $silent -ne 0 ]; then
    # The Wireshark build has lots of warnings, let's suppress them
    # if user wanted silent mode
    export CFLAGS="-fno-unit-at-a-time -march=native -O2 -pipe -w"
else
    export CFLAGS="-fno-unit-at-a-time -march=native -O2 -pipe"
fi
export CXXFLAGS="${CFLAGS}"
export LDFLAGS="-Wl,--hash-style=gnu -Wl,--as-needed"
export MAKEFLAGS="-j$numcpu"

if [ $clean -ne 0 ]; then
    echo -e "\nCleaning Wireshark build...\n"
    if [ $silent -ne 0 ]; then
        # Wireshark doesn't obey -s option for make
        make distclean > /dev/null
    else
        make distclean
    fi
fi

echo -e "Building Wireshark...\n"
if [ $silent -ne 0 ]; then
    autoconf > /dev/null
    ./configure --prefix=/usr --with-ssl --with-zlib=no > /dev/null
else
    autoconf
    ./configure --prefix=/usr --with-ssl --with-zlib=no
fi
if [ $? -ne 0 ]; then
    die "Failed to properly configure Wireshark, exiting"
fi

# check if we still need to uninstall
if [ $uninstall -ne 0 ]; then
    if [ $silent -ne 0 ]; then
        # Wireshark doesn't obey -s option for make
        sudo -k -p "Please enter your password to uninstall Wireshark: " \
            make uninstall > /dev/null
    else
        sudo -k -p "Please enter your password to uninstall Wireshark: " \
            make uninstall
    fi
else
    if [ $silent -ne 0 ]; then
        # Wireshark doesn't obey -s option for make
        make all > /dev/null
    else
        make all
    fi
    if [ $? -ne 0 ]; then
        die "Failed to build Wireshark, exiting"
    fi
    if [ $silent -ne 0 ]; then
        # Wireshark doesn't obey -s option for make
        sudo -k -p "Please enter your password to install Wireshark: " make install > /
            dev/null
    else
        sudo -k -p "Please enter your password to install Wireshark: " make install
    fi
fi

# Go back to where we started and exit
cd ..
echo "Done!"
exit 0
```