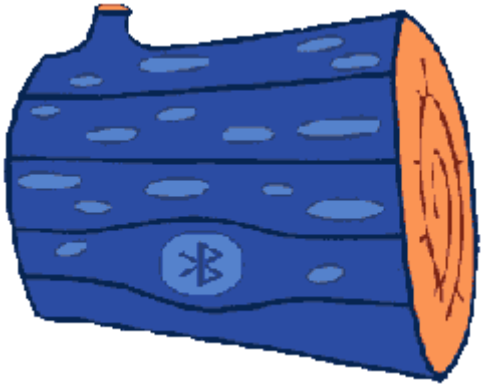

Bluelog

Bluelog is a Linux Bluetooth scanner with optional daemon mode and web front-end, designed for site surveys and traffic monitoring. It's intended to be run for long periods of time in a static location to determine how many discoverable Bluetooth devices there are in the area.

Since Bluelog is meant to be run unattended, it doesn't have a user interface or require any interaction once started. It features a fully configurable log file format, as well as the ability to log to syslog for centralized logging over the network.

Bluelog was meant to be lean and portable (its only requirement is BlueZ), and runs well on x86, MIPS, and ARM architectures. Bluelog is included in Kali Linux (www.kali.org), and on the Pwn Pad and Pwn Plug penetration testing devices from Pwnie Express (www.pwnieexpress.com). It's also available in the official OpenWRT repository and Arch Linux AUR community repository.

Bluelog Live

Bluelog Live is a constantly updated feed of discovered devices and their applicable information which can be viewed in any web browser. Live mode is designed primarily for public display, and is inspired by the infamous "Wall of Sheep" (www.wallofsheep.com).

Technically Live mode is a separate piece of software from the main Bluelog tool. It's comprised of simple static HTML pages and a CGI module that loads in the log files from Bluelog.

Bluelog is NOT a web server, it simply generates the HTML files which you will then serve with your HTTP daemon of choice (Apache, lighttpd, etc). Therefore, before you try and start Bluelog Live, make sure your web server is already configured properly. Bluelog is in fact completely unaware of the status of the Live pages while running, it just pumps out data and hopes for the best.

Basic Options

-i

This option tells Bluelog which Bluetooth device you want to scan with. You can use either the HCI device name (like hci2) or the MAC of the local adapter. As a bonus, if you give a device which doesn't exist, Bluelog will fall back on autodetection to find a working device.

-o

This is the (optional) filename of the log file to write. The default filename has the format of "bluelog-YYYY-MM-DD-HHMM.log", located in the current directory.

-v

Use this option to toggle displaying found devices on the console. Verbose output will also contain device class information and timestamps. Default is disabled.

-q

Turn off nonessential terminal output. In normal mode this means you will only see the start time of the scan and the message indicating proper shutdown. When used with daemon mode (-d), there will be no terminal output at all. The only exception to this option are critical errors, for obvious reasons.

-d

This option will daemonize Bluelog so that it runs in the background. You will still see the boilerplate and startup messages, but after that you will no longer see any info from Bluelog in the terminal.

-k

When running an instance of Bluelog in daemon mode, the -k option can be used to kill it.

Logging Options

-n

Use this option to toggle displaying device names for discovered devices. Finding the device name takes extra time during scanning, and occasionally fails. Therefore by not resolving device names, Bluelog can scan faster and more accurately. Default is disabled.

-m

This option, if enabled in the build, performs hardware manufacturer lookups of discovered devices via the MAC OUI. The hardware manufacturer will be logged in the standard log file, as well as Bluelog Live. The manufacturer database needs to be installed for this function to work, which makes it prohibitively large for some platforms (such as OpenWRT).

-c

This option toggles writing the raw device class to the log file. Enabling this option disables the -f option. Default is disabled.

-f

This option takes the device class and interprets it into a more human friendly format. It will tell you what class the device is and also what it's core capabilities are. For example, the class "0x7a020c" would appear as: "Smart Phone,(Net Capture Obex Audio Phone)". Enabling this option disables the -c option. Default is disabled.

-t

Use this option to toggle displaying timestamps for both the start and end of the scan and each new device found in the log file. Default is disabled.

-x

Use this option to toggle MAC address obfuscation. With this option enabled, Bluelog will display the manufacturer portion of each discovered MAC, but block out the device specific identifier. Default is disabled.

-e

Use this option to toggle CRC32 MAC address encoding. With this option enabled, the discovered MAC addresses will never be logged to disk, rather, each device will have a unique ID generated for it. This prevents privacy concerns during activities such as Bluetooth traffic monitoring. Default is disabled.

-a

This option enables "amnesia mode", which causes Bluelog to forget it has seen a particular device after a set amount of time, given here as minutes. When Bluelog encounters a device it has forgotten through this option, it will print it to the logs again as if it was the first time it has been seen, and the time found will be updated.

Output Options

-l

This option switches Bluelog over to Live mode, which uses an automatically updated web page to show results rather than the console and regular log files.

-b

This option will set the log format so that the resulting data is suitable for upload to ronin's Bluetooth Profiling Project (BlueProPro). This overrides most other logging options, and disables Bluelog Live. For more information on this project, and the additional steps required to submit your data for inclusion, visit: www.hackfromacave.com

-s

Use this option to toggle syslog only mode. In this mode Bluelog will not write its normal log file, and instead write only to the system log file (/var/log/syslog). This mode is especially useful when combined with a network aware syslog daemon, which can be used to add rudimentary central logging to multiple Bluelog nodes.

Acknowledgements

The initial code for Bluelog was based on sample code included in the book "Bluetooth Essentials for Programmers", by Albert Huang. This is a very informative book, and helps a lot if you are looking to get into BlueZ programming. It almost makes up for the terrible documentation from the BlueZ project.

The website for this book is located at: http://www.btessentials.com/

Bluelog also implements device class parsing code from "Inquisition", written by Michael John Wensley and released under the GPLv2.

You can read more about "Inquisition" from his site: http://www.wensley.org.uk/

Bluelog implements a modified version of the CRC hashing functions from "CRC Tester" by Sven Reifegerste.

Sven's page about CRC encoding: http://www.zorc.breitbandkatze.de/crc.html

The device cache rewrite took inspiration, if not literal code, from "SpoofTooph" by .ronin. You can check out "SpoofTooph" and .ronin's other projects on his site: http://www.hackfromacave.com/

Bluelog's UDP functions are based on code submitted by Ian Macdonald (ianmac51@googlemail.com).

Bluelog also uses some code inspired by functions from pidfile.c by Martin Schulze.

The font used for Bluelog Live's logo is "Electric Boots" by Jakob Fischer. You can see a collection of his fonts at: http://www.pizzadude.dk/

The OpenWRT version of Bluelog would not have been possible without the work of Gary Bonner and the logistical support of Joshua Hurst and Dean Nielson. Thanks also to Stephen Walker, who maintains the official OpenWRT packages for Bluelog, and has been an invaluable source of information on the platform.

Thanks to Jonas "onny" Heinrich for maintaining the Arch Linux build script for Bluelog. Read about his projects at: http://www.project-insanity.org/

Thanks to Dave Porcello, Jonathan Cran, and the entire Pwnie Express team for their support and assistance on the Pwn Plug optimized build of Bluelog.

Thanks to Paolo Valleri and the Integreen project for patches to Bluelog developed during the construction of their traffic monitoring network for Bolzano, Italy. Project site: http://www.integreen-life.bz.it/

Thanks to Teresa Brooks for adding the DEFCON Privacy Village theme for Bluelog Live.

Thanks to all users who have taken the time to contact me with comments and suggestions about Bluelog, which keeps pushing me in the right direction.

Finally, a special thanks to those who have donated Bluetooth devices to me for calibration purposes. Writing a Bluetooth scanner without any devices to scan is rather difficult, so the hardware has been very valuable to me.

License