License-compliant TLS stack for Apertis targets

Contents

The Apertis distribution provides both a development environment for electronic devices as well as a software stack to be used on them. In line with this goal, the Apertis project strives to provide software components that, where there is intent that they form part of the software stack on the devices themselves, are free from licensing constraints that may make it unsuitable in certain use cases. An example is software licensed under the terms of the GNU GPL-3[1] (General Public License) or LGPL-3[2] (Lesser General Public License) which are known to present a problem as they sometimes conflict with regulatory requirements[3] and thus Apertis will take measures to avoid such packages being provided as part of the "target" package repositories[4].

Goals and requirements

The goal here is to provide TLS functionality not just for the packages contained within its own repositories, but to support applications added by those utilizing Apertis as well.

- **Requirement:** TLS implementation does not require code covered by licenses that are incompatible with the target repositories rules
- **Requirement:** TLS implementation is licensed under terms that does not preclude its use from existing target applications
- **Requirement:** TLS implementation is licensed under terms that does not preclude its use from users proprietary applications

[1] https://www.gnu.org/licenses/gpl-3.0.en.html
[2] https://www.gnu.org/licenses/lgpl-3.0.en.html
[3] https://jwd.pages.apertis.org/apertis-website/policies/license-expectations/#licensing-constraints
[4] https://jwd.pages.apertis.org/apertis-website/policies/license-expectations/#apertis-repository-component-specific-rules

Given the security sensitive nature of the TLS stack, utilizing unmaintained software here would be best avoided. Putting maintenance aside, these versions of their respective TLS implementations may not be gaining support for any new ciphers and TLS protocol versions, which will severely limit their usefulness as time progresses. As well as not gaining newer protocol versions, the libraries may not be updated to reflect the frequently changing recommendations regarding minimal protocol versions[5] that should be supported, which may result in issues when attempting to access sites following the "Modern" recommendation. Additionally, it is likely that newer versions of the packages utilizing these TLS implementations will begin to require functionality added to newer versions of the TLS libraries thus reducing the ability of Apertis to upgrade to these too.

TLS stack pre v2021

The "target" section of Apertis ships a variety of packages which use TLS from a provided library. There are a number of software libraries that provide competing TLS implementations and which are provided under various licensing terms. However, these projects do not always provide the same programming interfaces, thus do not provide a drop in replacement for each other. Whilst some users of TLS libraries may provide some level of abstraction to support more than one TLS library, others may support only one and thus Apertis currently provides GnuTLS[6], OpenSSL[7] and NSS[8].

- **GnuTLS**: Apertis currently provides GnuTLS version 3.4.10. This is an approximately four-year-old version of GnuTLS as shipped in Ubuntu Xenial and thus is currently supported by Ubuntu and is expected to be until 2022. GnuTLS is used directly or indirectly via libcurl in just more than a dozen packages in target. Debian Buster, the current main upstream of Apertis, includes a newer version of GnuTLS (currently 3.6.7) though upgrading to this has already been avoided due to licensing issues that will be discussed below.

- **OpenSSL**: Apertis currently provides OpenSSL version 1.1.1. This is a relatively recent release in the 1.1.1 series and is packaged as part of Debian Buster. The 1.1.1 series is currently supported[9] as an LTS release by the OpenSSL project until September 2023. Support for Debian Buster is expected[10] until June 2024.

- **NSS**: Apertis currently provides NSS version 3.42.1. This version is approximately a year and a half old, and is packaged as part of Debian

[5] https://wiki.mozilla.org/Security/Server_Side_TLS
[6] https://www.gnutls.org/
[7] https://www.openssl.org/
[8] https://developer.mozilla.org/en-US/docs/Mozilla/Projects/NSS
[9] https://www.openssl.org/policies/releasestrat.html
[10] https://wiki.debian.org/LTS

Buster. As with OpenSSL, support for Debian Buster is expected until June 2024.

Some of the packages requiring TLS support only support one of the currently provided TLS implementations, often due to licensing compatibility. Other packages, most notably libraries, support multiple TLS backends, frequently including both GnuTLS and OpenSSL as options.

TLS stack post v2022

In order to have up to date libraries, specially TLS ones which very important for security reasons Apertis based them on Debian as covered in the Apertis Release Flow[11] which present the following issues for Apertis

GnuTLS

Whilst GnuTLS is licensed under the LGPL-2.1[12], it uses Nettle[13] and GMP[14]. Newer versions of both of these dependencies are now licensed as dual GPL-2 and LGPL-3, rather than **L**GPL-2.1.

To avoid including GnuTLS under LGPL-3 terms, should Apertis integrate a newer version it would need to be utilized under the GPL-2 terms. This would result in the binary GnuTLS library effectively being used under the terms of the GPL-2 rather than LGPL-2.1. This would restrict Apertis users from using this Apertis provided TLS implementation either directly or indirectly from any non-GPL-2 compatible applications they wish to integrate into their systems, for example in proprietary applications, where it would have the effect of requiring the app to also be GPL-2 licensed.

In such a scenario, a newer GnuTLS library could be allowed by accepting its dependencies under the GPL-2 license and restricting its use to places where this license wouldn't be problematic, such as existing GPL-2 software. As the existing applications written exclusively to use GnuTLS are GPL-2 or tolerant of GPL-2, this is viable.

OpenSSL

The currently used version of OpenSSL is licensed under a custom GPL-incompatible license. OpenSSL 3.0 (the next major version of OpenSSL) will be licensed under the Apache 2.0[15] license, which is compatible with the GPL-3, but not GPL-2. This means that GPL-2 tools like `tumbler`, `connman`, `apt` or `systemd-journal-remote` cannot use the newer versions of OpenSSL without

[11] https://jwd.pages.apertis.org/apertis-website/policies/release-flow/#apertis-release-flow
[12] https://www.gnu.org/licenses/old-licenses/lgpl-2.1.en.html
[13] https://www.lysator.liu.se/~nisse/nettle/nettle.html
[14] https://gmplib.org/
[15] https://www.apache.org/licenses/LICENSE-2.0

effectively becoming GPL-3 licensed or through these upstream projects applying a license exceptions (for example as OpenVPN[16] has). The OpenSSL project do not seem to hold a strong opinion on the compatibility, though suggest[17] either not using the GPL or applying an exception should you wish to gain some legal certainty.

The compatibility between the current OpenSSL licensing and GPL-2 is based on the premise that:

1. The OpenSSL license[18] contains licensing terms not in the GPL (such as the need to mention use of the software in all advertising material and derivatives not being able to be called OpenSSL).
2. Linking OpenSSL with a GPL-2 application creates a derivative work formed from the two pieces of code.
3. The GPL expressly states[19] that one can't "impose any further restrictions on the recipients' exercise of the rights granted herein" to the GPL licensed work.

Likewise, the Apache 2.0 license, to which version 3 of OpenSSL will be release under, contains clauses such as its patent litigation license termination clause[20].

While the argument made in step (2) is widely held by many, others disagree with this interpretation, especially when the library is dynamically linked to the application. For instance, it might be claimed[21] that a dynamically linked library is only truly combined with the application when run, not when distributed, so it would only become a derivative at that point, or it might be claimed[22] as this is the intended interface for interacting with a library this is excluded either due to fair use laws in some jurisdictions or explicitly allowed by the GPL when it states[23] "the act of running the Program is not restricted".

A further argument is that the GPL states[24] "as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable". If the library is distributed as part of the OS and can be considered a major component of it, then this clause doesn't require the library to be considered as part of the software and therefore falls outside of the scope of the license. A counter argument to this is that because the application may also be considered to be distributed as part of the operating system this exception doesn't apply especially in embedded devices where the

[16] https://spdx.org/licenses/openvpn-openssl-exception.html
[17] https://www.openssl.org/docs/faq.html#LEGAL2
[18] https://www.openssl.org/source/license-openssl-ssleay.txt
[19] https://www.gnu.org/licenses/old-licenses/gpl-2.0.html#section6
[20] http://www.apache.org/licenses/LICENSE-2.0#patent
[21] https://lwn.net/Articles/548216/
[22] https://www.linuxjournal.com/article/6366
[23] https://www.gnu.org/licenses/old-licenses/gpl-2.0.html#section0
[24] https://www.gnu.org/licenses/old-licenses/gpl-2.0.html#section3

software is distributed preinstalled as a complete entity.

Most distributions seem to either ignore this potential issue or do not consider a policy to be needed. The Fedora project have deemed OpenSSL to be a system library[25] as defined by the GPL and thus there is no incompatibility. Debian historically decided that a linked library creates a derivative work and all the packages it ships should be considered a combined work, though the decision has recently been taken[26] to follow Fedora's lead here.

NSS

Network Security Services[27] (NSS) is a set of security libraries developed by Mozilla. NSS provides its own API, which is currently only supported by a few of the applications which use TLS in Apertis. It is licensed as MPL-2.0[28].

Approach

In order to fullfil the requirements the approach taken has been to upgrade GnuTLS to a new version for those applications that can use it licensed as GPL-2. With OpenSSL upgraded and retained as a system library, utilizing it, inline with the approach taken by other distributions that have documented a specific policy covering this.

The one outlier is the printing support in GTK which uses GnuTLS and which potentially ends up causing GPL-2 dependencies in GTK. Whilst Debian have also declared CUPS as a system library, we feel that the differing use cases for Debian and Apertis make this less of a realistic position to take. We have therefore dropped printing support from GTK in order to remove this dependency as we don't feel that this functionality is critical to Apertis' aim.

Summary

The tables below summarize the use of TLS libraries in various releases of Apertis target images. We would expect proprietary applications to either utilize the OpenSSL or NSS libraries as deemed appropriate by the individual projects.

TLS stack pre v2022

Component	License	OpenSSL	GnuTLS	Notes
apt	GPL-2+		X	

[25] https://fedoraproject.org/wiki/Licensing:FAQ?rd=Licensing/FAQ#What.27s_the_deal_with_the_OpenSSL_license.3F

[26] https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=924937#105

[27] https://developer.mozilla.org/en-US/docs/Mozilla/Projects/NSS

[28] https://www.mozilla.org/en-US/MPL/2.0/

Component	License	OpenSSL	GnuTLS	Notes
connman	GPL-2		X	
curl	curl and BSD-3-Clause and BSD-4-Clause-UC and ISC	X	X	also produ
glib-networking	LGPL-2.1+ and LGPL-2.1+ with OpenSSL exception		X	
liboauth	Expat/MIT		curl	
libmicrohttpd	LGPL-2.1+		X	uses curl f
neon27	LGPL-2.1+	X	X	
openjpeg	BSD-2		curl	
openldap	OLDAP-2.8		X	
rtmpdump	GPL-2+ (tools), LGPL-2.1+ (library)		X	
systemd	LGPL-2.1+ and GPL-2[+] and PD X		curl	
tumbler	LGPL-2.1+ and GPL-2+		curl	

163 # TLS stack post v2022

Component	License	OpenSSL	GnuTLS	Notes
apt	GPL-2+		X	
connman	GPL-2		X	
curl	curl and BSD-3-Clause and BSD-4-Clause-UC and ISC	X	X	also produ
glib-networking	LGPL-2.1+ and LGPL-2.1+ with OpenSSL exception	X		after rebas
liboauth	Expat/MIT	curl		
~~libmicrohttpd~~	LGPL-2.1+		X	removed s
neon27	LGPL-2.1+	X	X	
openjpeg	BSD-2	curl		package li
openldap	OLDAP-2.8	X		
~~rtmpdump~~	GPL-2+ (tools), LGPL-2.1+ (library)		X	removed s
systemd	LGPL-2.1+ and GPL-2[+] and PD X	curl		package sy
tumbler	LGPL-2.1+ and GPL-2+	curl		