INSTITUTE FOR DEFENSE ANALYSES

Case Study: OpenSSL 2012 Validation

David A. Wheeler

August 2013

IDA Document D-4991
Log: H 13-001174
Copy

INSTITUTE FOR DEFENSE ANALYSES
4850 Mark Center Drive
Alexandria, Virginia 22311-1882

The Institute for Defense Analyses is a non-profit corporation that operates three federally funded research and development centers to provide objective analyses of national security issues, particularly those requiring scientific and technical expertise, and conduct related research on other national challenges.

About This Publication
This work was conducted by the Institute for Defense Analyses (IDA) under contract N66001-11-C-0001, subcontract D6384-S5, "Homeland Open Security Technology (HOST)," for Georgia Tech Research Institute. The views, opinions, and findings should not be construed as representing the official position of either the Department of Defense or the sponsoring organization.

INSTITUTE FOR DEFENSE ANALYSES

IDA Document D-4991

Case Study: OpenSSL 2012 Validation

David A. Wheeler

Executive Summary

This is a case study of the Federal Information Processing Standards (FIPS) 140-2 validation of the OpenSSL FIPS Object Module that led to certificate #1747 (initially awarded on June 27, 2012). This case study documents what happened during the validation, including identifying lessons learned for future projects.

OpenSSL is a cryptographic library available through an open source software (OSS) license. Under U.S. Federal Government policy, U.S. Federal agencies and their contractors are generally required to use software validated by the Cryptographic Module Validation Program (CMVP), using criteria defined in the FIPS 140 series, for the cryptographic protection of sensitive or valuable data within Federal systems. Unfortunately, this policy provides no mechanism to fund validation. Since OpenSSL is freely available, there is no simple funding mechanism to have this widely used software validated.

An "OpenSSL FIPS Object Module" (a.k.a. "FIPS module") had been previously created. The FIPS module is a specially devised software component that was designed for compatibility with OpenSSL and created so that users can use a version of OpenSSL as a FIPS 140-validated cryptographic module. The FIPS module is about one-sixth the size of full OpenSSL and contains only the mature and especially important capabilities of full OpenSSL. Although the FIPS module had been previously validated, it was based on a much older version of OpenSSL and thus lacked important capabilities.

The Defense Advanced Research Projects Agency (DARPA) provided funding for the evaluation of the OpenSSL FIPS module for two platforms in 2011 through 2012. Once DARPA committed to this initial funding, many other organizations (both government and private) joined the evaluation project by providing additional funding. The resulting evaluation covered many more platforms for an updated version of the FIPS module.

A number of lessons learned were identified:

1. *Keep everyone informed on the project's status*. The OSF ensured that all potential sponsors knew that validation time, or even success, could not be guaranteed before the validation process began, and sent emails to sponsors on a regular basis.
2. *A formal, incorporated entity is often needed*.

3. *Don't underestimate the amount of paperwork and the time it takes to get contracts*. It can be a full-time job, even for a small business.
4. *Funding can be erratic; it is best to ease into a business*.
5. *The Government has advantages as a customer*. The U.S. Federal Government imposes a lot of paperwork, but the Government is generally predictable in what it requires and it pays on time.
6. *OSS projects and companies often need help with marketing*. Many Government organizations who might want a service do not know who to contact for help with open source software, and do not know how to find them.
7. *Pool funding*. This validation applied to far more platforms and algorithms than any one sponsoring organization would have been willing to fund.
8. *Don't add code at the last minute before a validation*. A few cryptographic algorithms were added at the last minute; OSF wished (in hindsight) that it hadn't done that, because those additional algorithms extended the validation process by about 3 months.
9. *Do incremental validations*. OSF believes it would be less expensive and more efficient to do validations more often (e.g., once per year) as each validation would have to address far fewer changes.
10. *Where practical, make requirements (including their interpretations) public*. Commercial organizations can strive to meet government requirements, but they are less likely to succeed if the requirements (including their interpretations) are not public.
11. *Lack of response by developers is often misunderstood*. If key developers for popular OSS software become deluged by requests, they may ignore requests for help. This can be misunderstood by potential users as a lack of interest in the project. There are many ways to counter this problem; the key is to create a mechanism to counter the problem if it is needed.

Due to the pooling of funding from different organizations, by 2013-08-30 the certificate covered 63 operating environments, a significant growth from the validation of two operating environments (platforms) originally funded by DARPA. Overall, this case study demonstrates that when organizations pool their resources, they can achieve far more than any one of them would have been willing to do on its own.

Contents

1. Case Study Overview

This case study follows the Department of Homeland Security (DHS) case study template, and is intended for inclusion in the set of DHS Homeland Open Security Technology (HOST) project case studies on open source software (OSS) projects involving the U.S. Government.

- **Case study subject name:** OpenSSL 2012 Validation, Certificate #1747
- **Case study subject description:** The FIPS 140-2 validation, in 2011 through 2012, of the "OpenSSL FIPS Object Module" (a cryptographic module), receiving certificate #1747. Defense Advanced Research Projects Agency (DARPA) initially funded the evaluation of OpenSSL for two platforms; other organizations (both government and private) then joined the evaluation project by providing additional funding, resulting in coverage of a vast number of platforms for an updated version of OpenSSL.
- **Key Organizations and Projects:**
  - OpenSSL Software Foundation (OSF), http://www.openssl.com, a for-profit company,
  - OpenSSL project, http://openssl.org/, a non-profit open source software (OSS) project.
- **Estimated number of people involved in implementing the case study:** At least 20 people were directly involved (as sponsors, coordinators, developers, and validators).
- **Estimated number of people who used the case study result:** Unknown, however, there are probably millions of users who are impacted directly, and hundreds of millions who are indirectly affected. Cryptographic libraries are essential for providing confidentiality and integrity over networks (including the World Wide Web); in many cases the U.S. Federal government is required to use Federal Information Processing Standards (FIPS)-validated cryptographic libraries; and it is reported that most FIPS validations today are based on OpenSSL. Even companies that are not required to use FIPS-validated libraries use validation to reduce their risks and enhance product marketing. Since the FIPS module is based on full OpenSSL, validation also provides some evidence that the full OpenSSL software works correctly.

- **What time period does the case study cover**? 2010−2012. Validation began January 4, 2011, and the validation was initially awarded on June 27, 2012. The time period covered here is slightly longer, to include the work before validation officially began and follow-on work via change letters (which can expand the covered platforms).
- **Study Category:** HOST Partner

2. Background

Cryptography "is a branch of mathematics based on the transformation of data. It provides an important tool for protecting information and is used in many aspects of computer security" [NIST SP 800-12]. For example, a web browser viewing an "https://" URL uses the cryptographic protocol defined in the Transport Layer Security (TLS) specification or its predecessor the Secure Sockets Layer (SSL) specification to create a secure interaction. Cryptographic functions are provided by cryptographic modules.

Because of the importance of cryptographic modules, on July 17, 1995, the National Institute of Standards and Technology (NIST) established the Cryptographic Module Validation Program (CMVP). The CMVP's purpose is to validate cryptographic modules as a joint effort between NIST and the Communications Security Establishment Canada (CSEC). CMVP now validates modules against the Federal Information Processing Standards (FIPS) 140-2, Security Requirements for Cryptographic Modules, first released on May 25, 2001. The CMVP depends on various commercial testing labs which perform the actual testing. U.S. Federal agencies and their contractors are generally required to use FIPS 140 validated cryptographic modules when using cryptography. Any particular validation applies to only specific platforms, where a "platform" is defined by key properties such as the central processing unit (CPU) used (e.g., x86 or ARM[1]), the instruction bit size (e.g., 32-bit or 64-bit), and the operating system. Validating additional platforms requires additional resources.

The development of the cryptographic library "SSLeay" by Eric A. Young and Tim J. Hudson also began in 1995. This work effectively (though unofficially) ended around December 1998 when both started to work for the company RSA Security. Others were interested in continuing to maintain an open source software cryptographic library. This successor library, based on SSLeay, was termed OpenSSL.

The OpenSSL project is a "collaborative effort to develop a robust, commercial-grade, full-featured, and Open Source toolkit implementing the [SSL and TLS] protocols as well as a full-strength general purpose cryptography library. The project is managed by a worldwide community of volunteers…" [OpenSSL 2012a]. The OpenSSL project maintains OpenSSL, a widely used cryptographic module (a toolkit and library). The first release of OpenSSL, version 0.9.1c, was released on December 23, 1998, and can be

[1] ARM™ stands for Advanced RISC Machine (though at one time it stood for Acorn RISC Machine). RISC, in turn, stands for Reduced Instruction Set Computing. ARM is the name of the computer architecture family licensed by ARM Holdings plc that are widely used in mobile devices.

considered the beginning of the OpenSSL project. The OpenSSL project has continued since.

In 2001, the Defense Medical Logistics Standard Support (DMLSS) system developers selected OpenSSL because it was a technically efficient program and they did not wish to re-allocate limited funds to acquire an expensive proprietary solution to replace OpenSSL. They decided to sponsor OpenSSL for FIPS 140-2 validation. DMLSS contracted the Open Source Software Institute (OSSI) to manage coordinating the OpenSSL development community, various corporate sponsors, and the testing lab to do this validation; the OSSI/DMLSS technical lead was Steve Marquess. As part of this process, OSSI created the "OpenSSL FIPS Object Module" (aka the "FIPS module"). The FIPS module is a specially devised software component that was designed for compatibility with OpenSSL and created so that users can use a version of OpenSSL with a FIPS 140 validated cryptographic module. The FIPS module is about one-sixth the size of full OpenSSL and contains only the mature and especially important capabilities of full OpenSSL.

The FIPS module validation was successful, although it took far longer than most validations (which typically take less than a year). This validation was "down to the source level"; this was a challenge for the FIPS validation process since these processes had historically dealt with hardware and binary executables. Additionally, the validation was challenged by some competing vendors. On January 19, 2006, CMVP awarded the first validation number to the project; this was recanted in January 23. Validation was later awarded on March 21, 2006, as certification #642, but this validation was listed as "not available," making the certification basically useless. On February 6, 2007, a third successful validation (certificate #733) was awarded, enabling U.S. federal government organizations to use the OpenSSL FIPS module at a substantial per-module savings to the Government. By this point, it was clear that there was a need for a more permanent legal entity to perform follow-on validations and other services related to OpenSSL.

In 2009, the OpenSSL Software Foundation (OSF) was incorporated as a for-profit U.S. company. OSF serves as a "corporate entity representing the OpenSSL project for the purpose of providing financial support in the form of support contracts, consulting services, and donations" [OSF 2012]. The OSF hires and funds many of the OpenSSL project's developers. Steve Marquess is the co-founder, president, and business manager of OSF.

The focus of this case study is the FIPS 140-2 validation of the OpenSSL cryptographic library FIPS module that received certificate #1747. The validation being considered here began January 4, 2011; validation was awarded on June 27, 2012. However, the time period covered here is January 2010 through November 2012, because this case study considers work that occurred before validation officially began, as well as the work that continued afterward through change letters (which have expanded the

covered platforms). OSF worked as the primary submitter of the FIPS module and was the vendor for purposes of the validation. Steve Marquess (of OSF) was the key coordinator of this validation effort. As discussed below, many organizations eventually agreed to co-sponsor this validation.

3. Problem Statement

The problem was the need to validate, under FIPS 140, the updated version (2.X series) of the open source software OpenSSL FIPS object module, with many additional algorithms and many different supported platforms (e.g., Android).

FIPS 140 validation is a requirement across nearly all U.S. Federal government activities, and many projects (government and not) use OpenSSL. Thus, many U.S. Federal Government agencies and their contractors needed a validated FIPS module and would want a FIPS-validated version of OpenSSL. Additionally, at least some commercial organizations wanted OpenSSL validated to justify to potential customers that this key component was trustworthy. The OpenSSL FIPS module had been validated before, but for many this old validation was not sufficient. Many wanted or needed to have an updated OpenSSL validation for the version 2.X series instead of the obsolete 1.X series. Many wanted a FIPS module that supported additional platforms (e.g., Android), satisfied new CMVP testing guidelines, and provided various algorithms not in some previous validations. Examples of these additional capabilities include adding new pseudo-random number generator (PRNG) implementations, adding algorithm test programs for certain algorithms, adding the RSA encryption algorithm, updating the Digital Signature Algorithm-2 (DSA2) algorithm for key sizes greater than 1,024 bits, and supporting TLS version 1.2 (the current version of the protocol used by "https:" URLs). These additional capabilities are absolutely vital for many uses of the module. The OpenSSL developers have also developed several improvements such a better implementation of power-on self-test (POST), required by FIPS validation, that takes 3 seconds instead of 3 minutes. There was also a desire to re-design the FIPS module to more completely separate it from the "normal" OpenSSL code, to greatly simplify future maintenance of the full OpenSSL suite.

While there was no serious technical challenge, there were other hurdles. The fundamental challenge is that various U.S. Federal Government regulations require that cryptographic modules be validated for FIPS 140 compliance, but do not directly provide for any funding to perform this validation. Thus, from the point of view of U.S. Federal Government services and agencies, FIPS validation is an unfunded mandate. It had been presumed that cryptographic module suppliers would pay for this validation, and then charge the Government (in the form of higher prices). But the OpenSSL module is available for use at no charge, so this presumption is simply false for OpenSSL. Since anyone could use the results of an OpenSSL evaluation, different organizations each hoped that "someone else" would pay for the evaluation. This led to a standoff in which

each organization was waiting for someone else to pay for the work. Steve Marquess tried to broker a cooperative agreement for an updated source-level validation, without success. A few organizations were willing to provide a small amount of money, but not enough to commit the resources required for the validation, and these agreements were often for limited times (making it difficult to accrete a large-enough total at any one time).

At this point, DARPA had developed some prototype software on Android using widely used commercial off-the-shelf (COTS) components, including a recent version of OpenSSL. DARPA needed to get the software through standard acquisition requirements for long-term use, and a key roadblock was FIPS validation of the version of OpenSSL that they used. DARPA determined that the best approach would be to validate a current version of OpenSSL for use on Android.

4. Approach

DARPA determined in 2010 that it needed to validate an updated OpenSSL on Android (initially on two platforms), and decided to contract OSF to help do this. This made sense; previous OpenSSL validations had made clear that a legal entity to spearhead OpenSSL validation was needed, and validation support was one of the reasons OSF had been formed. DARPA agreed to pay the funds expected to be necessary to do a full validation, originally for just two platforms: (1) Android 2.2 on an ARM processor with the NEON[2] instruction set, and (2) Android 2.2 on an ARM processor without NEON. (DARPA later added Android 3.0 and Android 4.0, in both cases for ARM with NEON and ARM without NEON.)

Once DARPA committed to sponsoring the validation, OSF asked other potential sponsors if they wanted join the validation, e.g., to add platforms or algorithms, by providing additional funds to do so. One sponsor quickly funded evaluation for another platform, Windows 7 32-bit x86. OpenGear then funded the platform ucLinux on ARM version 4 (without NEON), and Quintessence funded the platform Fedora 14 on x86 64-bit with the AES-NI instructions (Fedora is a distribution of a Linux-based operating system). Additional platforms were soon funded, such as HP-UX (both 32 & 64 bit) and Ubuntu 32-bit x86. One sponsor wanted Windows 7 for both 32-bit and 64-bit, and learned that another sponsor was already paying for the 32-bit x86.

Eventually there were over 12 distinct sponsors (including 4 U.S. Government agencies) for this validation, and later follow-on work received even more sponsors. The main rule for this validation was that additional sponsors could pay for additional capabilities (typically additional platforms), but at the expense of the overall validation schedule. This rule was later slightly relaxed; see the discussion below for more about this. Sponsors varied in their reasons for joining; OpenGear (a commercial company) noted that, "we use OpenSSL in our products, so things that benefit OpenSSL benefit us too." Full disclosure: the DHS became one of those later sponsors.

Steve Marquess of the OSF stated, "without DARPA funding, this validation would never have gone anywhere; it would never have even started. DARPA made it possible; the other later secondary sponsors made it better."

---

[2] NEON™ is a combined 64- and 128-bit single instruction multiple data (SIMD) instruction set for accelerating media and signal processing for the ARM processor family. We have not found a phrase it stands for, so we are considering it as a name and not as an acronym.

DARPA initially did not want to be identified as a key sponsor of the validation. This was not a serious problem, however; they could simply work with OSF using a non-disclosure agreement, and OSF would be the supplier of record. OSF reported that DARPA later decided to permit public acknowledgement of their support.

The validation being considered here began January 4, 2011; validation was awarded on June 27, 2012, as certificate #1747. The time period covered in this case study is actually January 2010 through November 2012, so that it can include the preparation work before validation officially began and the follow-on work that continued afterward through change letters (these expanded the number of covered platforms). In particular, on July 9, 2012, the first "change letter" update was approved, adding six additional platforms and a new revision number of 2.0.1. This validation built on the validation of the OpenSSL FIPS object module version 1.2 as described in FIPS 140-2 certificate #1051; that previous validation is not the subject of this case study.

5. Issues

The validation itself took somewhat longer and more money than was originally expected. The goal was to complete the validation in 12 months; it actually took 18 months. Of those extra 6 months, at least 3 months of delay were attributable to the late addition of new cryptographic algorithms; it only took a month to develop that code, but those changes caused at least 3 more months (total) of delays.

Sponsors varied in their sensitivity to this delay. For at least one sponsor, the delay was critically important, as it affected government deployment. For at least one other sponsor (OpenGear), delay was not as important; for them, validation was primarily a way to augment confidence in the software, both for themselves and for potential customers (e.g., as an item in marketing literature).

A general problem was that although the open source edition of OpenSSL had been validated before, it was a significantly older version (there had been a large delay between validations). Thus, there was a lot of work to do for the new validation because so many changes had accumulated in OpenSSL.

One challenge for validation in general is the relative lack of public written interpretations of FIPS 140-2. The FIPS 140-2 requirements were not designed for today's modern software-driven systems, and thus must be interpreted. By itself, that is inevitable; technology simply changes too fast for things to be otherwise. Unfortunately, there seems to be no CMVP equivalent of the National Information Assurance Partnership/Common Criteria Evaluation and Validation Scheme (NIAP/CCEVS) public interpretations database, where interpretations are made and publicly posted for future reference. Instead, CMVP interpretations appear to be treated as confidential information and are not made available to the public. This lack of public written interpretations results in uncertainty on what the requirements mean, and this affects the time, cost, and predictability of the validation process.

6. Results

In the end, the OpenSSL validation effort of 2011-2012 was successful. Its key goal was to get an updated OpenSSL FIPS module validated, and it was validated. The pooling of funding also meant that the validation covered far more algorithms and platforms than any individual sponsor would have been willing to fund. For example, at the beginning, DARPA only had enough funding to validate two platforms. Due to pooling of funding from different organizations, by 2013-08-30 the certificate covered 63 operating environments (platforms) [CMVP 2012].

All evidence suggests that overall the sponsors were satisfied. Some sponsors did not like the delays, but OSF made clear in its contracts that OSF couldn't promise that validation would be complete by a certain date (or that it would complete), so the sponsors knew the limitations of the project. OSF contacted sponsors first for their concurrence in cases where a decision might delay evaluation, so the time-sensitive sponsors knew that OSF was trying to complete the validation quickly. Perhaps the most telling evidence of sponsor satisfaction is that many sponsors (easily half of the original sponsors) are funding OSF for follow-on work (e.g., adding more platforms).

7. Lessons Learned

1. *Keep everyone informed on the project's status.* The OSF ensured that all potential sponsors knew that validation time, or even success, could not be guaranteed before the validation process began. As validation proceeded, the OSF sent emails to sponsors on a regular basis, explaining the validation status (including problems and good news). This was helpful, especially for those who had particular expectations or needs.
2. *A formal, incorporated entity is often needed.* OSF was specifically created as a legal entity to perform actions that are difficult to do without one. For example, OSF's existence greatly simplifies contract work, e.g., it can act as the supplier for validation and it can provide annual support contracts. There is often a need for an organization that can act as a broker – a bridge – between the "contracting world" and the "open source software development world." The original developers of an OSS project are often the most qualified people to update the software, but they are sometimes unprepared (or unwilling) to deal with the contracting process. Thus, these developers often need an intermediary to connect them to those potential customers who need their services.
3. *Don't underestimate the amount of paperwork and the time it takes to get contracts.* Steve Marquess reported that he spent three-fourths of his time just doing corporate paperwork (proposals, contracts, invoices, non-disclosure agreements, etc.). Indeed, he reported that, "I thought I could do it part-time, but the time I spend talking on the phone, etc., is easily a full-time job, [even though it is] for just a handful of people."
4. *Funding can be erratic; it is best to ease into a business.* Many small businesses, including OSF, depend on contract revenue that comes in "spasmodically." Additionally, many contracts are paid primarily after the work is done (and possibly long after the invoice is sent), not before. While some small companies can commit to something in a day, most other organizations will not, so there is often a long lag between discussion and commitment. These cash flow problems make it difficult to do long-term planning. It is best to "ease into" such a business, to avoid spending all the money before finishing work necessary to receive funding.

5. *The Government has advantages as a customer.* The U.S. Federal Government imposes a lot of paperwork, but the Government is generally predictable in what it requires and it pays on time.

6. *OSS projects and companies often need help with marketing.* Many Government organizations who might want a service do not know who to contact for help with open source software, and do not know how to find them. For example, many organizations have technical staff who use OpenSSL, yet many of those same staff do not even know that the OSF exists.

7. *Pool funding.* This validation applied to far more platforms and algorithms than any one sponsoring organization would have been willing to fund.

8. *Don't add code at the last minute before a validation*. A few cryptographic algorithms were added at the last minute; OSF wished (in hindsight) that it hadn't done that, because those additional algorithms extended the validation process by about 3 months. OSF did get DARPA's approval to do this, and that was important, but OSF underestimated how long the delay was going to be. If the code isn't ready before a validation, OSF recommends that they "wait for the next one."

9. *Do incremental validations.* Because it had been a long time since the last full open source validation of the OpenSSL FIPS module, much had changed, and much needed to be done. OSF believes it would be less expensive and more efficient to do validations more often (e.g., once per year) as each validation would have to address far fewer changes.

10. *Where practical, make requirements (including their interpretations) public.* Commercial organizations can strive to meet government requirements, but they are less likely to succeed if the requirements (including their interpretations) are not public. Publicly releasing written interpretations also provides an opportunity for broader feedback, potentially resulting in better and clearer interpretations.

11. *Lack of response by developers is often misunderstood.* Many potential users of OSS understandably want free private help. However, the developers for popular OSS software can quickly become deluged by these requests. A common coping mechanism is for the key developers to ignore requests for help; after all, they don't have time and they want to write code instead of doing customer relations. This can be misunderstood by potential users as a lack of interest in the project. This can be countered by creating a mechanism to fund those developing and supporting the software. There are many ways to do this (this case study discusses just one), but the key is to create such a mechanism when it is needed.

References

[OpenSSL 2012a] OpenSSL project. OpenSSL front page. http://www.openssl.org. Retrieved 2012-10-29.

[OpenSSL 2012b] OpenSSL project. "OpenSSL and FIPS 140-2 Validation Status." http://www.openssl.org/docs/fips/fipsvalidation.html. Retrieved 2012-10-29.

[OSF 2012] OpenSSL Software Foundation (OSF). OSF front page. http://www.openssl.com. Retrieved 2012-10-29.

[CMVP 2012] Cryptographic Module Validation Program (CMVP). "Validated FIPS 140-1 and FIPS 140-2 Cryptographic Modules." http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140val-all.htm#1747. Retrieved 2013-08-30.

[NIST SP 800-12] NIST. "Special Publication 800-12: An Introduction to Computer Security - The NIST Handbook" NIST SP 800-12. http://csrc.nist.gov/publications/nistpubs/800-12/800-12-html/

Much of the material included in this case study was derived from private interviews.

Appendix A
Acronyms

ARM™	Advanced RISC Machine (was: Acorn RISC Machine)
CCEVS	Common Criteria Evaluation and Validation Scheme
CMVP	Cryptographic Module Validation Program
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit
CSEC	Communications Security Establishment Canada
DARPA	Defense Advanced Research Projects Agency
DHS	Department of Homeland Security
DSA2	Digital Signature Algorithm-2 (per FIPS 186-3)
FIPS	Federal Information Processing Standards
HOST	Homeland Open Security Technology
NIAP	National Information Assurance Partnership
NIST	National Institute of Standards and Technology
OSF	OpenSSL Software Foundation
OSS	Open Source Software
RISC	Reduced Instruction Set Computing
RSA	(Ron) Rivest – (Adi) Shamir – (Leonard) Adleman
SIMD	Single Instruction Multiple Data
SSL	Secure Sockets Layer (predecessor of TLS)
TLS	Transport Layer Security (successor of SSL)
DMLSS	Defense Medical Logistics Standard Support
OSI	Open Source Initiative
OSSI	Open Source Software Institute
POST	Power-on self-test

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE (DD-MM-YY)	2. REPORT TYPE	3. DATES COVERED (From – To)
26-08-2013	Final	

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Case Study: OpenSSL 2012 Validation	N66001-11-C-0001, subcontract D6384-S5
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBERS
6. AUTHOR(S)	5d. PROJECT NUMBER
David A. Wheeler	
	5e. TASK NUMBER GT-5-3329
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESSES	8. PERFORMING ORGANIZATION REPORT NUMBER
Institute for Defense Analyses 4850 Mark Center Drive Alexandria, VA 22311-1882	D-4991 H13-001174
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR'S / MONITOR'S ACRONYM
Joshua L. Davis Georgia Tech Research Institute (GTRI) 250 14th Street NW, Room 256 Atlanta, GA 30318	GTRI
	11. SPONSOR'S / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

Project Leader: David A. Wheeler

14. ABSTRACT

This is a case study of the Federal Information Processing Standards (FIPS) 140-2 validation of the OpenSSL FIPS Object Module that led to certificate #1747 (initially awarded on June 27, 2012). This case study documents what happened during the validation, including identifying lessons learned for future projects. OpenSSL is a cryptographic library available through an open source software (OSS) license. The Defense Advanced Research Projects Agency (DARPA) provided funding for the evaluation of the OpenSSL FIPS module for two platforms in 2011 through 2012. Once DARPA committed to this initial funding, many other organizations (both government and private) joined the evaluation project by providing additional funding. Overall, this demonstrates that when organizations pool their resources, they can achieve far more than any one of them would have been willing to do on its own.

15. SUBJECT TERMS

OpenSSL, FIPS, FIPS 140, FIPS 140-2, CMVP, open source software, free software, validation, certification, certificate, cryptography, cryptographic library, cryptographic module, DARPA, HOST, DHS, GTRI, OpenSSL Software Foundation, OSF, OpenSSL FIPS object module, 1747, DMLSS, lessons learned

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Joshua L. Davis
Unclassified	Unclassified	Unclassified	Unlimited	30	19b. TELEPHONE NUMBER (Include Area Code) 678-831-0182