REPORT
OF THE
DEFENSE SCIENCE BOARD
TASK FORCE
ON
INTERNATIONAL ARMAMENTS COOPERATION

-- *SOURCE CODE WORKING GROUP* —

ASSESSMENT OF DoD SOURCE CODE EXPORT PRACTICES

AUGUST 1996

This report is a product of the Defense Science Board (DSB). The DSB is a Federal Advisory Committee established to provide independent advice to the Secretary of Defense. Statements, opinions, conclusions and recommendations in this report do not necessarily represent the official position of the Department of Defense.

Executive Summary

The Under Secretary of Defense for Acquisition and Technology has requested the Defense Science Board (DSB) Task Force on International Arms Cooperation to prepare an assessment of current DOD policy regarding the export control of weapon systems source code. The DSB Task Force established a Source Code Working Group to address this request.

The goal of the Working Group was to assess whether or not the present OSD practice (and those of other DOD Service Components, Activities and Agencies) of prohibiting export of weapon systems software source code:

1. should continue, and
2. if not, what the new policy should be,

From its assessment, the Working Group presents the following findings:

1. Source code is not sensitive in and of itself, but can be important because of the information concerning system capabilities and/or vulnerabilities it may contain. Guidelines and policies should focus on the means to protect the sensitive information source code contains, and not on protecting source code for the sake of protecting source code.

2. Current DOD export controls on weapon system software source code reflect a case-by-case review by multiple DOD organizations, with no integrated, centrally approved or directed policy. In most cases, these reviews assume a denial of release of software source code, even when machine code has been approved for release.

3. The lack of a single DOD definition for source code is a problem in our current export control process. It generates confusion, inconsistencies and arbitrariness in source code export release decisions. An accepted definition of software source code is an essential prerequisite for improving our current export control policies. (A proposed definition is contained herein.)

4. The current strategy of protecting information by prohibiting source code export is being rendered ineffective by the availability of modem software tools and information promulgated in the commercial sector. Also, machine code, released under approved foreign military sales, is vulnerable to assembly level reverse engineering.

5. The release of source code could permit alteration of machine code that has undergone validated testing and is under warranty by US manufacturers. Foreign access to source code may result in tampering that invalidates the manufacturer's warranty. (A solution is presented herein.)

6. Export flexibility for software source code can be achieved without compromising security. There are effective means to control the dissemination of information without exercising an automatic presumption of denial for the export of the complete vehicle containing the information.

7. The software source code release process currently in place presents an ongoing and vexing problem to the DOD acquisition community, US defense industry and to the economic and security relationships between the US and friendly nations.

Recommendations:

The need for an effective source code security process has not changed. But the need to introduce some degree of flexibility in certain areas has become acute. Further, the combined effects of declining domestic defense budgets and increased requirements for friendly country software programming facilities provide additional impetus for achieving greater export release flexibility.

Even with greater export release flexibility, the Working Group recognizes there may be a special need to protect some select weapon systems source code (perhaps deemed especially "sensitive" by government program managers). The Working Group believes that the number of cases warranting such special treatment is very small.

To increase flexibility in the current software source code export process, and to address the key issues described earlier, the Working Group makes the following recommendations:

1. Establish a Revised Security Review Process and Release Criteria.

The Working Group recommends that the Secretary of Defense promulgate a DOD-wide source code export policy that:

- **Establishes revised security standards for the export of weapon systems with their functionally essential machine code that recognizes the vulnerability of machine code to reverse engineering.** These standards should be taken fully into account in the export license review for weapon systems sales.

- **Establishes a revised security standard for the export of weapon systems software source code.** The revised security standard should recognize the current vulnerability of machine code to reverse engineering by presuming, *a priori,* a compromise of associated source code whenever binary (machine) code is released, regardless of any additional security precautions employed (such as the use of "tamper-proof' boxes or software "traps").

- **Makes the source code export decision consistent with the result of the revised security evaluation.** If the revised security evaluation permits the export of the weapon system, then foreign requests for associated source code should be reviewed with the presumption of approval, unless the release can be specifically demonstrated to be harmful to United States defense interests.

- **Protects US industry** by requiring formal notification that acquisition of source code nullifies any performance warranty provided for the weapon system in question.

- **Requires the purchasing country to provide the same level of protection** (regarding security and technology transfer) for the information contained in the source code as appropriate for that information in clear text.

2. Create an Official, Comprehensive Definition of Source Code

The DOD-wide policy described should include a single definition of software source code for use in export control decisions involving weapon systems. This definition should be used by all DOD Agencies, Service Components and Activities responsible for reviewing source code export release requests. The Working Group proposes the following definition:

> "Source code is any set of software instructions and data composed in a language higher than machine language (binary code). Source code is a form of software that is readily transformed into machine code (compiled or assembled). Source code is easier to read, understand and manipulate than machine code. It may be stored on paper, on magnetic disks, in circuitry or in other storage media. Source code can be written in assembly language or higher order languages."

3. Establish a Periodic Review of Software Source Code Export Policy

The policy should establish a recurring review of software source code export controls to ensure that they reflect modem technology developments. The Working Group recommends a review period not less than once every three years.

I. Tasking

The Under Secretary of Defense for Acquisition and Technology has requested the Defense Science Board (DSB) Task Force on International Arms Cooperation to prepare an assessment of current DOD policy regarding the export control of weapon systems software source code. Mr. Everett D. Greinke, a member of the Task Force, agreed to lead a Source Code Working Group to address this request.

This report presents the findings, conclusions and recommendations of the Source Code Working Group. The Working Group focused primarily on the broad policy issues influencing software source code export control and not upon creating or formulating new export control language. While a number of authorities were consulted for this analysis, the findings, conclusions and recommendations are those only of the Working Group, and do not necessarily represent the views of any Defense Agency, Activity or Service Component.

II. Objective and Goal

The objective of this report is to assess current export control release processes involving software source code associated with weapon systems from the aspect of DOD's overall acquisition, armaments cooperation and Foreign Military Sales (FMS) activities.

The goal is to assess whether or not the present OSD practice (and those of other DOD Service Components, Activities and Agencies) of prohibiting export of weapon systems software source code:

1. should continue, and
2. if not, what the new policy should be.

Given rapidly changing technology, does the present practice provide the requisite security for weapon systems information conveyed by software source code?

III. Working Group Definition of Software Source Code

Software is an integral part of all programmable digital computers. The software field has matured and now employs well known engineering principles for its design, development, test and evaluation and quality control processes, as in other scientific and engineering disciplines.

Still, software is a relatively new technology area and is a rapidly changing discipline. Software engineers and general managers alike struggle to stay *current* and *relevant* with changing technology. Keeping pace with state-of-the-art software developments has, in part, resulted in multiple definitions for software source code. The Working Group has identified several definitions of source code in existing DOD documents that address the export control of weapon systems software source code. We suspect that there are more definitions in other documents found within both the public and private sector.

Instead of selecting one definition over another, the Working Group crafted a consolidated definition of source code for use in this report. This definition combines the essential and common elements from the definitions that were identified during the assessment. We believe the proposed definition helps remove an unnecessary mystique about source code protection which complicates the export control review and decision making process.

Working Group Definition of Software Source Code

First we define *software,* then we define software *source code.*

As defined in this study, *software* is any set of instructions (and associated data) that describe actions to be performed by a general purpose, programmable digital computer. Today’s computers are based upon high performance microprocessors which use software to carry out assigned tasks. Software may be contained in many different physical forms, including storage on magnetic disks, within large scale integrated circuits (such as application specific integrated circuits), in highly dense magnetic cards, or even as text on paper.

Software increasingly is being stored within integrated circuits. Software stored in this fashion is sometimes called “firmware”.

All software is designed to provide the processor the necessary instructions to perform specific tasks. Without software, the computer is “just an empty shell”. With the proper instructions and data (“programs”), the computer becomes a machine that performs a variety of operations or tasks.

Thus, we can categorize software by the task it is designed to have the computer perform. Examples of software categories used by some export control specialists are Operational

software, Application software, Maintenance/Diagnostic software and Support software/tools. Sub-categories may include fire control software, radar processing software, data processing software and display software, for example.
Software can also provide great insight into the functions and operations being performed: it can be classified as intellectual property, and considered as proprietary or “sensitive” by its owners. In fact, the International Traffic in Arms Regulation (ITAR) defines software as “Technical Data”.

Weapon systems software share these characteristics. For example, weapon systems software offers detailed (potentially critical) insight into the operational missions, data, functions and related platform, hardware and weapons characteristics.

Software Source Code

To define source code, we must first understand machine code.

Regardless of the intended task, all software instructions and data eventually must be transformed into code that the computer (i.e. microprocessor) can “understand” (operate upon). This code is termed “machine code”.

Machine code always consists of a string of binary numbers (l’s and 0’s). Machine code is the most fundamental form of software representation. The protocol for organizing the binary strings is called “machine language” and is determined by the specific microprocessor being used. Different microprocessor families have different protocols (different machine languages). Commercial microprocessor families have their protocols published in the public domain. (Some military versions of commercial microprocessor families also exist. There are only slight differences between the protocols published for commercial versions and those used with military versions.) Uniquely Government developed and/or controlled computers may or may not have their protocols published in the public domain. Access to these protocols is an important factor in the viability, or lack thereof, of current export control practices involving source code.

Machine code is provided to friendly countries when they buy modem weapon systems from the United States. The code is typically provided on tape that can be loaded into the weapon system for its operation but it may also be provided in other forms. If a foreign country prints out (downloads) the machine code, the visual representation it receives is that of binary patterns (strings of l’s and 0’s) arranged according to the machine language protocols appropriate to the host hardware.

Because code written in machine language is notoriously hard to read, interpret and modify, progressively more convenient software languages have been created over time. Code written in these so called “higher level” languages becomes easier to manipulate and understand. These languages constitute a ease-of-use hierarchy for programmers and system analysts. In essence, with each new generation of higher level language, the ease of creating software programs increases, often dramatically.

Nevertheless, all code written in any higher level language must be translated into machine code for the computer to act upon it. Translation from a higher level language to machine code is accomplished in a series of steps by another software program -- a *compiler.*

The Working Group defines source code to be any set of software instructions and data composed in a language higher than machine language (binary code). Source code is a form of software that is readily transformed into machine code (compiled or assembled). Source code is easier to read, understand and manipulate than machine code. It may be stored on paper, on magnetic disks, in circuitry or in other storage media. Source code can be written in assembly language or higher order languages, as shown in Figure 1, (Note, however, that the highest potential language shown in Figure 1, spoken language, does not yet have reliable compilers).

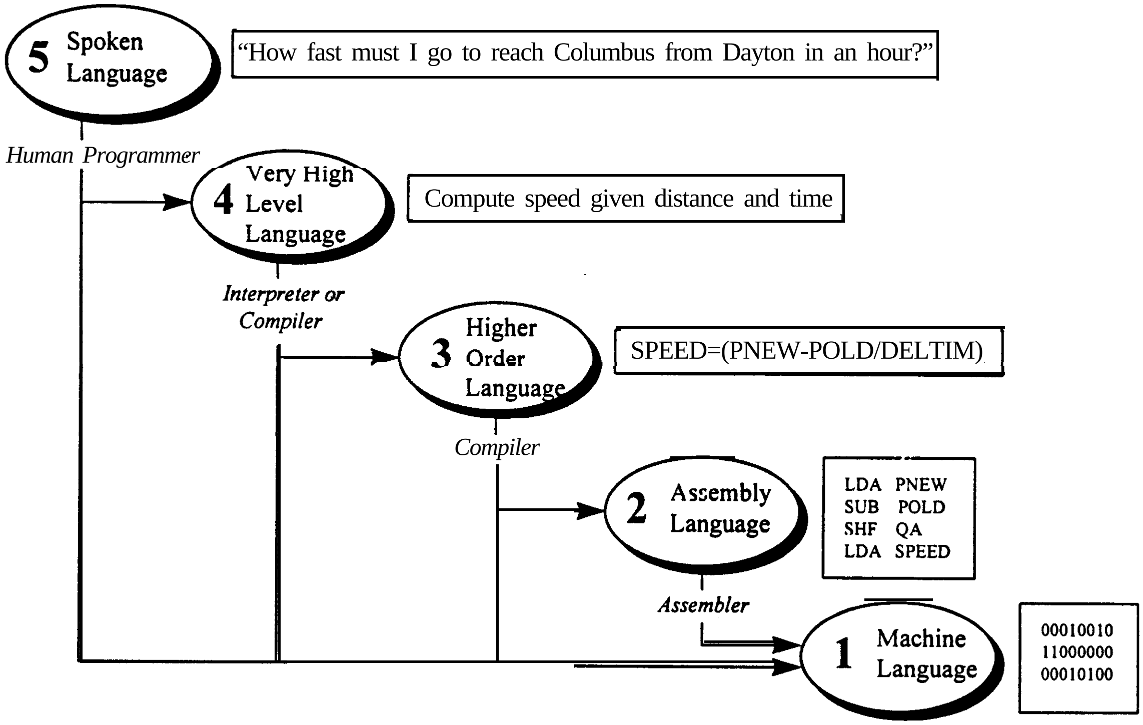


Figure 1. Programming Languages Hierarchy
Adapted from Figure 2-4 of Reference 10

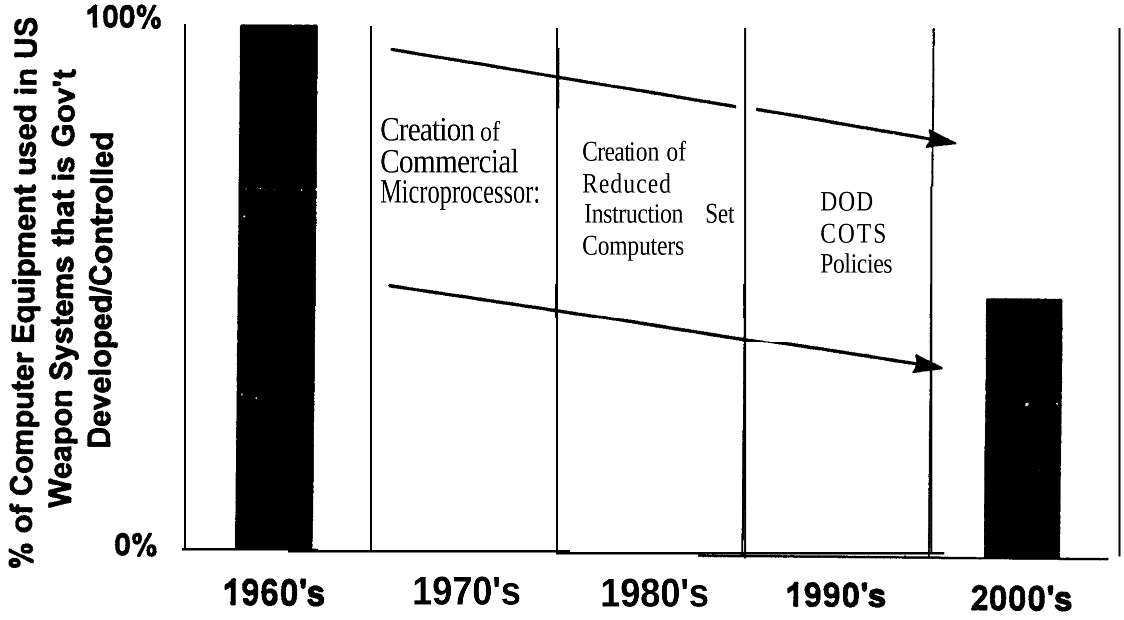
IV. Overview of Current Situation

1. General Overview

Current export control guidelines for the release of weapon systems software source code are aimed at preventing the release of sensitive information concerning: operational capabilities, functions, system technical characteristics, technology advances as well as other related items (such as algorithms for intelligence data fusion and signal processing and information about vulnerabilities and limitations of the system under operational conditions).

Current export control practices evolved over time in an environment when the technology of computers used in weapon systems were dominated by Government owned and/or controlled equipment. Using a strategy of source code denial to protect our software advantage was practical and effective in this environment. The dominance of unique Government developed/controlled equipment (which includes hardware, software and technical data) in weapon systems continued unchallenged until the early 1970's when the introduction of commercially available microprocessors, which were more capable and less expensive than Government developed/controlled computers, began a trend toward the use of commercially available (commercial off-the-shelf or "COTS") technology in military hardware, as shown in Figure 2.

Figure 2. Commercially Developed Computers Replacing Gov't Developed Computers in US Weapon Systems (Illustration Only)



From the early 1960's through the late 1980's, most, if not all, software source code, tools and documents were provided as Government developed/controlled equipment or supplied by a DOD or Service Component Software Support Activity (SSA). Use of uniquely Government developed/controlled computers in development and subsequent operational and support phases of the weapon system was the norm for almost three decades, regardless of the solid state electronics being used.

In the 1980's, commercial processor chip sets were introduced into new mission computer applications as well as in up-grades to various platform types and models. These COTS (and non-developmental items or "NDI") processors were supplied with industry developed software tools, environments, documentation and source code. They either replaced the original code or resided in the same mission computer chassis "along" with new software programs developed using the latest higher order languages (such as Ada, C, C++, etc.).

Upgrade options: a mix of technologies

Use of Reduced Instruction Set Computer (RISC) processors and the DOD-wide commitment to the use of COTS and NDI products accelerated the trend toward a mix of technologies, commercially-based and Government developed/controlled equipment, in US weapon systems. As weapon systems evolve, a variety of upgrade options become available.

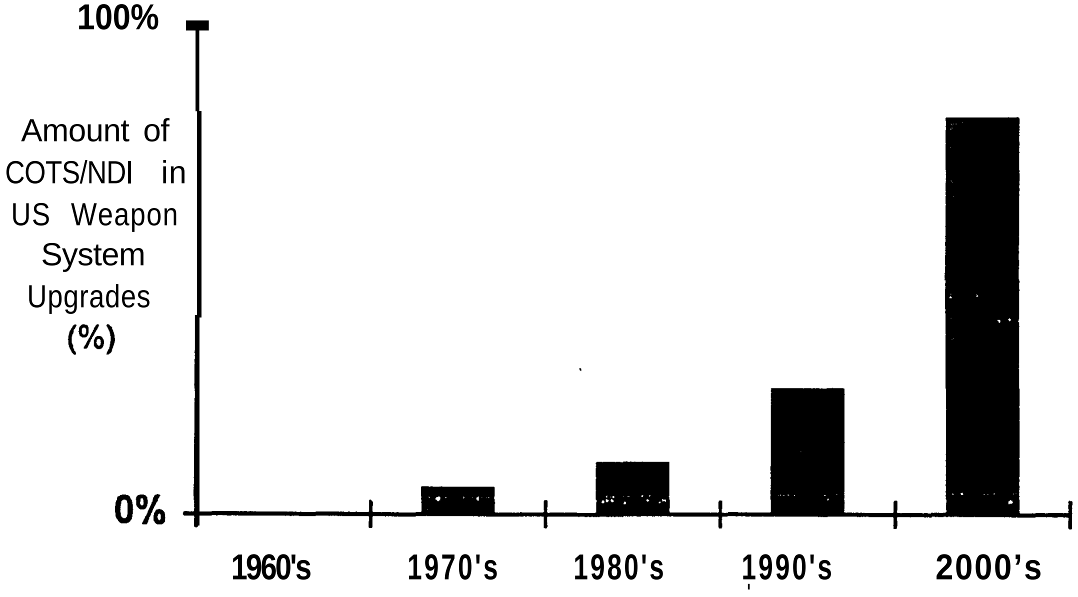
One upgrade option is to carry forward the original software of the legacy system in its original language. This is a high cost option given the time and effort required to replicate older technology and in general it does not take full advantage of the hardware improvements available to the system.

A second upgrade option is to re-do the legacy software using modem higher order languages. This approach also incurs a large cost up front. But the cost is not recurring, the option also eases software maintenance over the lifetime of the weapon system and it reduces life cycle operational and support costs. This approach also permits leveraging modem software advances (such as the creation of highly modular code) and is more suitable for exploiting hardware advances as well.

A third option is to keep some or all of the legacy Government controlled/developed software and include it concurrently with new COTS/NDI software. This costly option involves maintaining two separate sets of source code and associated software support packages.

Over time, cost and performance requirements will increasingly require the use of COTS and NDI technology in US weapon systems, regardless of the option employed, as shown in Figure 3.

Figure 3. Trend: Increasing use of COTS/NDI in US Weapon System Upgrades (Illustration Only)



Reverse Engineering: Growing Ease

Unlike their Government developed/controlled predecessors, commercially available microprocessor-based computers have extensive documentation available in the public domain. This documentation is more than sufficient to allow “reverse engineering” of any provided machine code into higher levels of source code. This is also true for military versions of commercial microprocessor families. Thus, the increasing use of commercially-based microprocessors in modem weapon systems is undermining US efforts to control access to weapon system information by controlling access to source code. Legacy systems and upgrades also are affected.

Weapon systems sales from the United States to allied or friendly nations typically involve the transfer of software in the form of machine code, but do not normally permit the release of the associated source code. Machine code is released because the buyer must be able to load (and re-load) the operational program into the weapon system. Thus, machine code is considered to be an integral part of the weapon system. DOD export control practices tend to treat source code as an entity separate from the underlying weapon system information.

With increasing frequency, friendly countries are requesting the release of source code for the weapon systems they have purchased. Their desire to have source code encompasses a wide spectrum of operational, maintenance and economic considerations. Access to the source code, along with the associated software development tools (software engineering environment), helps these countries to:

- Tailor or "fine tune" the weapon system's functions and tasks to their own unique theater of operations.
- Reduce hardware and software maintenance costs and minimize the dependency upon activities located within the CONUS to institute changes to the operational programs.
- Gain a better understanding of the overall military capability of the weapon system

Current DOD-wide Policy

There is no overall, *integrated, officially approved* DOD policy that addresses the release of software source code. Rather, the Office of the Secretary of Defense (OSD), the individual Service Components, and the affected DOD Agencies and Activities have established separate export control release practices, judgments and de facto "policies" based upon case-by-case analyses.

Therefore, current DOD practices on source code export is more an amalgamation of the perspectives of many organizations rather than the implementation of a single, centrally directed overall policy approved by the Secretary of Defense. This case-by-case approach has both advantages and disadvantages to it.

The export case reviewer's criteria place the burden of proof solely upon the party initiating the request to demonstrate that release criteria have been met. In practice, the release process is typically based on the *a priori* premise of denial of the request. This approach offers, to a large degree, protection from undesired release of the information. But the process tends to generate decisions that, when examined collectively, appear unnecessarily rigid and contradictory. The DOD review process currently uses either OSD draft policies and draft guidelines or Service policies generated between 1989 and 1992.

2. Acquisition and Technology Perspectives

Computer hardware and its associated software have become essential technology elements in the development and acquisition of modem weapon systems. Our future war-fighting capability increasingly will be determined by the quality of computers and software employed by weapon systems and their associated communications, logistics and management systems.

But the growing dominance of computers and software in defining military capability has come with rapidly escalating software costs. As stated in Air Force software development guidelines:

> “During the 1970’s, the rapid evolution of sophisticated electronic circuitry resulted in smaller processors, producing more computing power for a fraction of the cost. These advances, coupled with more demanding requirements, dramatically increased DOD’s use of software intensive systems. In the 1992 fiscal-year alone, DOD spent over $35.2 billion on computer dominated systems, $29.1 billion, or 83% of which, was for software." [10]

Proportionally similar costs are also being felt by defense establishments around the world.

Adding to this perspective, major software acquisition reform also has occurred in the United States Federal Government. On February 10, 1996, President Clinton signed the Information Technology Management Reform Act of 1995, which outlines new software acquisition requirements for DOD and the entire federal government.

The legislation calls for performance measurements in new Federal Acquisition Regulations (FAR’s) for software and specifies that any new regulations require:

- The use of commercial-off-the-shelf (COTS) software to the maximum extent practical.
- Incremental purchases of software capabilities vice large single projects.

The acquisition and technology perspective must balance many conflicting forces involving software source code. First, in part to reduce costs, our own defense software investment is moving toward the use of COTS software. This requirement will tend to internationalize the availability of weapon systems software source code (and modularize it) over time. Second, friendly nations increasingly are requesting access to source code for weapon systems software already purchased in order to reduce high maintenance costs and to refine their regional operational capabilities. Third, we must endeavor to protect our technology advantage, including our industrial base, which is increasingly defined by the sophistication of computer hardware, software, other electronic devices and the availability of domestic technology sources.

3. Present Policy Perspectives

The present policy perspective is based upon the historical practice of protecting national security through denial of requests to release weapon systems source code. Foreign availability of comparable technology should play a major role in the release or denial decision, but often does not because of a lack of accurate, timely information on the source code and weapons capabilities of foreign sources.

This perspective is based upon the premise that denying access to source code:

- Helps maintain US operational advantage.
- Helps protect the system engineering and system integration "know-how" of US industry.
- Helps maintain jobs in the US defense industrial base.
- Slows "reverse engineering" efforts and makes them more costly.
- May protect the means to identify collection resources and other covert capabilities.

It should be noted that release of source code for a specific weapon system often serves as a precedent for obtaining follow-on releases with greater ease.

4. Service Components, Agencies and Activities Perspectives

The Service Components, Agencies and Activities practice is one of denying source code release requests in most cases. These practices are not necessarily coordinated with one another, but represent each organization's independent approach to source code export controls. These organizations share many of the same concerns of the acquisition policy community, and have some additional concerns, including:

- Losing configuration control by releasing weapon systems source code, especially for systems that require coalition interoperability.
- Contractual claims of operational failure for these systems may become the responsibility of the DOD organizational element that approved the release of the source code, or may involve highly complex legal issues should the source code be released through direct commercial sale.
- Maintaining strict adherence to the organization's export control policies, including any unique interpretations of that policy by the individual organizational elements.
- The desire by Service Components to control follow-on software support associated with the weapon system. Such support includes major platform upgrades, corrections of deficiencies or new functional or operational requirements.

5. Industry Perspectives

The Working Group did not attempt to undertake a comprehensive survey of industry perspectives. However, the Working Group reviewed industry comments concerning a 1992 draft OSD policy document entitled "International Transfer and Export Control of Software and Software Source Code, Documentation and Software Development Tools."[3] The Working Group also reviewed the 1992 White Paper of the Electronic Industries Association which provided industry comments on current export control practices involving software source code and concluded:

> "The current policy on source code release is an unnecessary burden upon industry. As our examples have noted, this policy has had a direct negative impact upon industry's exports. Encouragement of competitors to provide source code does not achieve the goal of protecting the release of all source code. It also results in the loss of jobs at a time of critical importance for our industry and the country as a whole. A policy which removes the current presumption of denial while allowing for its release within established parameters and based upon foreign availability would assist industry and government. Like DOD, industry does not want to release data deemed absolutely critical for national security. In addition, industry does not want to release information which could be proprietary or could be used to develop competitive products. However, industry does have a need to release various types or levels of source code." [4]

The Working Group believes that these comments still represent industry opinion concerning the release of source code.

The Working Group also noted that software performance warranties are of legitimate concern to US industry. Foreign buyers demand performance warranties for understandable reasons: validated and reliable software is critical for military operation, especially for man-in-the-loop functions.

However, a request for source code access implies an intent to modify the delivered software -- why else ask for the code? Any modification can potentially damage the functionality of the delivered system and result in an unjustified warranty claim. Thus, releasing source code without providing protection from unjustified claims is also undesirable.

6. Foreign Country Perspectives

The Working Group did not attempt a survey of foreign nation perspectives. However, experience shows that, often, a nation has specific reasons for each request for release of weapon systems software source code.

Viewed from the international perspective:

- The United States is one of many competing defense product suppliers around the world. Foreign buyers will go to the most attraetive competitive source. Access to source code increasingly is an important discriminator between competitors.
- Budget pressures require optimal use of foreign weapon systems investments, and that means increasing access to source code as a cost-driven requirement. The worldwide decline in defense budgets places increasing pressure on foreign buyers to develop their own source code capability or to purchase it from abroad.
- In general, it is faster and cheaper to buy source code than to reverse engineer available machine code, even though purchasing source code tends to reduce the urgency to develop full indigenous capabilities. Still, any nation with the ability to exploit provided source code will most likely also have sufficient ability to reverse engineer machine code into source code, should they deem it vital to their national security. It should be noted that the costs associated with developing an indigenous reverse engineering capability are decreasing as software tools advance.

V. Assessment of Current Practices

1. Strengths of the current guidelines

Export controls on software source code evolved to protect sensitive information and data that is inherent to the operation of a weapon system. Without some kind of access to source code, buyers are often prevented from obtaining a specific understanding of the military capabilities and functions of their purchased weapon systems that the US government does not want them to have.

The underlying security philosophy in DOD's current source code control approach is clear: make it as hard as possible to gain insight into the system from a technical viewpoint, force heavy investments in resources and build in a time delay to "decipher" any information contained in the machine code.

This approach has been in place for almost two decades. It evolved during an - environment when weapon systems software and hardware protocols were uniquely defined and controlled by the US government. Protection of source code was a feasible and prudent security practice under these conditions.

Under current practice, requests for software source code export are almost always denied by DOD organizations. The primary advantage of this practice is that it minimizes the risk of inadvertent release of sensitive operational data and characteristics. It also provides a case-by-case review opportunity for each Service and Defense Activity or Agency concerned. Although not centralized, the process does appear to involve the entire FMS and export control security establishment.

2. Shortcomings of the current guidelines

The major shortcoming of maintaining the current export control practices for software source code stems from fundamental change in the technology environment that renders a denial strategy ineffective. By continuing practices designed for an earlier environment, we create a false sense of security while simultaneously harming the US industrial base. Furthermore, a culture of denying source code for the sake of denying source code perpetuates an approach which misses the true security issue of protecting information, rather than the vehicle containing the information.

As described earlier, those nations wishing to take full advantage of their weapon systems purchases now have increasing budgetary incentives to gain access to the source code, whether through reverse engineering or through access granted by the seller. At the same time, reverse engineering from machine code to source code has never been easier, especially for assembly level source code.

Software engineers interviewed by the Working Group estimate that 95% of machine code currently in use in operationally deployed systems can be reverse engineered into assembly level code without much difficulty. This is due to the large amount of technical knowledge available in the public domain about the mission computers (and associated microprocessors) currently in use.

Today, reverse engineering of machine code to source code is readily achievable for modem systems (namely, those employing microprocessors based on commercial products) with the success rate directly related to the amount of resources applied to perform the reverse engineering, as well as the specific higher level language being targeted.

Furthermore, major software developers, whether private sector firms studying products of a competitor or nations trying to gain military advantage, will have some form of reverse engineering capability. Investment and desire are the key determinants to the degree of sophistication of the capability. By not releasing source code, we give foreign buyers an increased security incentive to domestically reverse engineer the machine code they already possess. Encapsulating source code in tamper proof boxes or building security "traps" in the software can help slow the reverse engineering process, but can not be counted on to stop it.

Software stored within an integrated circuit ("firmware") is also vulnerable to compromise, although the reverse engineering techniques used against firmware-stored information are quite different from those used against traditionally stored software. Information stored as firmware may represent itself in the circuit layout, gate design or static information stored in the chip. In the case of firmware-stored information, reverse engineering may be attempted by de-encapsulating the chip that holds the firmware and by studying the chip with equipment, tools, and processes that are well-established in the semiconductor industry. Firmware-storage of information can provide more levels of physical security against tampering than does traditional software storage, and reverse engineering of firmware-stored information requires more collective expertise (from the semiconductor side). However, the tools and techniques needed to defeat most of these protection schemes are widely available in the worldwide semiconductor industry.

Of immediate concern is the relative ease of converting machine code into assembly level source code. Assembly language source code is the easiest language level reached by reverse engineering. Furthermore, assembly language is the "language-of-choice" for today's operational weapon system source code -- most currently operational weapon systems software is written in assembly language -- and will be for the foreseeable future. Higher order languages (like Ada , C, C++, etc.) are not yet widely used in weapon system applications.

In time, higher level languages will find their way into operational weapon systems software source code. Continuing advances in software tools and environments will make the process of reverse engineering to these higher languages even easier than it is at present.

Therefore, no matter what languages are used in future weapon systems, the belief that weapon systems software source code can be protected after the machine code is released is no longer a prudent security assumption.

3. Key Issues for Evaluating Current Practices and Guidelines Development

The following key issues are central to developing pragmatic and effective export control policies and guidelines for software source code which protect US national security and related military advantage.

Issue 1: Control information, not the vehicle that carries it.

A fundamental precept for developing effective policies and guidelines is to recognize that source code, by itself, is not the object of security concerns. Rather, it is the **information** contained within the source code that may warrant review. Source code is only one means of carrying this information. Machine code, which is always released in approved export sales, not only carries nearly the same information as source code but also provides direct weapon systems functionality. Information control for modem weapon systems must now take into account demonstrated vulnerabilities of machine code to reverse engineering and the growing use of modular software in weapon systems components.

Issue 2: Adequately reflect the current technology environment.

At present, DOD export control practices for software source code assume Government developed/controlled equipment dominance -- an environment that is rapidly disappearing. Guidelines must be structured to properly reflect existing and future technology environments and the mix of technologies that will comprise them, old and new.

Issue 3: Achieve a consistent approach throughout DOD.

The current practice contains the embedded potential for inconsistent implementation of export controls. A DOD-wide process that reflects the appropriate organizational responsibilities and perspectives needs to be crafted.

Issue 4: Stay relevant when technology advances.

Technology trends are fast moving. For example, fully modular approaches to software design are emerging. Such trends can undermine any export control strategy over time. Therefore, guidelines and policies need to have periodic re-evaluations of their effectiveness to ensure that they have not been rendered obsolete or counterproductive as technology advances.

VI. Findings of the Working Group

Based upon the analyses given in the preceding sections, the Working Group presents the following findings:

1. Source code is not sensitive in and of itself, but can be important because of the information concerning system capabilities and/or vulnerabilities it may contain. Guidelines and policies should focus on the means to protect the sensitive information source code contains, and not on protecting source code for the sake of protecting source code.

2. Current DOD export controls on weapon systems software source code reflect a case-by-case review by multiple DOD organizations, with no integrated, centrally approved or directed policy. In most cases, these reviews assume a denial of release of software source code, even when machine code has been approved for release.

3. The lack of a single DOD definition for source code is a problem in our current export control process. It generates confusion, inconsistencies and arbitrariness in source code export release decisions. An accepted definition of software source code is an essential prerequisite for improving our current export control policies.

4. The current strategy of protecting information by prohibiting source code export is being rendered ineffective by the availability of modem software tools and information promulgated in the commercial sector. Also, machine code, released under approved foreign military sales, is vulnerable to assembly level reverse engineering.

5. The release of source code could permit alteration of machine code that has undergone validated testing and is under warranty by US manufacturers. Foreign access to source code may result in tampering that invalidates the manufacturer’s warranty.

6. Export flexibility for software source code can be achieved without compromising security. There are effective means to control the dissemination of information without exercising an automatic presumption of denial for the export of the complete vehicle containing the information.

7. The software source code release process currently in place presents an ongoing and vexing problem to the DOD acquisition community, US defense industry and to the economic and security relationships between the US and friendly nations.

VII. Recommendations

The need for an effective source code security process has not changed. But the need to introduce some degree of flexibility in certain areas has become acute. Further, the combined effects of declining domestic defense budgets and increased requirements for friendly country software programming facilities provide additional impetus for achieving greater export release flexibility.

Even with greater export release flexibility, the Working Group recognizes there may be a special need to protect some select weapon systems source code (perhaps deemed especially "sensitive" by government program managers). The Working Group believes that the number of cases warranting such special treatment is very small.

To increase flexibility in the current software source code export process, and to address the key issues described earlier, the Working Group makes the following recommendations:

1. Establish a Revised Security Review Process and Release Criteria

The Working Group recommends that the Secretary of Defense promulgate a DOD-wide source code export policy that:

- **Establishes revised security standards for the export of weapon systems with their functionally essential machine code that recognizes the vulnerability of machine code to reverse engineering.** These standards should be taken fully into account in the export license review for weapon systems sales.

- **Establishes a revised security standard for the export of weapon systems software source code.** The revised security standard should recognize the current vulnerability of machine code to reverse engineering by presuming, *a priori,* a compromise of associated source code whenever binary (machine) code is released, regardless of any additional security precautions employed (such as the use of "tamper-proof' boxes or software "traps").

- **Makes the source code export decision consistent with the result of the revised security evaluation.** If the revised security evaluation permits the export of the weapon system, then foreign requests for associated source code should be reviewed with the presumption of approval, unless the release can be specifically demonstrated to be harmful to United States defense interests.

- **Protects US industry** by requiring formal notification that acquisition of source code nullifies any performance warranty provided for the weapon system in question.

- **Requires the purchasing country to provide the same level of protection** (regarding security and technology transfer) for the information contained in the source code as appropriate for that information in clear text.

2. Create an Official, Comprehensive Definition of Source Code

The DOD-wide policy described should include a single definition of software source code for use in export control decisions involving weapon systems. This definition should be used by all DOD Agencies, Service Components and Activities responsible for reviewing source code export release requests. The Working Group proposes the following definition:

> “Source code is any set of software instructions and data composed in a language higher than machine language (binary code). Source code is a form of software that is readily transformed into machine code (compiled or assembled). Source code is easier to read, understand and manipulate than machine code. It may be stored on paper, on magnetic disks, in circuitry or in other storage media. Source code can be written in assembly language or higher order languages.”

3. Establish a Periodic Review of Software Source Code Export Policy

The policy should establish a recurring review of software source code export controls to ensure that they reflect modem technology developments. The Working Group recommends a review period not less than once every three years.

Appendices

A. Definitions

Commercial Items. Items regularly used in the course of normal business operations for other than Government purpose which: (a) have been sold or licensed to the general public; (b) have not been sold or licensed, but have been offered for sale or license to the general public; (c) are not yet available in the commercial marketplace, but will be available for commercial delivery in a reasonable period of time; or (d) are described in (a) through (c) above, that would require only minor modification in order to meet the requirements of the procuring agency (DFARS, Part 2 I 1).

Commercial Off-the-Shelf (COTS). "Commercial items that require no unique government modifications or maintenance over the life-cycle of the product to meet the needs of the procuring agency." [Source: DOD Directive Number 5000.2, Subject: Mandatory procedures for Major Defense Acquisition Programs (MDAPs) and Major Automated Information System Acquisition Programs (MAISAPs), Appendix VII Glossary of Key Terms & Concepts].

Computer Software (or Software). "A combination of associated computer instructions and computer data definitions required to enable the computer hardware to perform computational or control functions." [Source: DOD Directive Number 5000.2, Subject: Mandatory procedures for Major Defense Acquisition Programs (MDAPs) and Major Automated Information System Acquisition Programs (MAISAPs), Appendix VII Glossary of Key Terms & Concepts].

Nondevelopmental Item: "Any item of supply that is available in the commercial marketplace; any previously developed item of supply that is in use by a department of the United States, a State or local government, or a foreign government with which the United States has a mutual defense cooperation agreement." [Source: DOD Directive Number 5000.2, Subject: Mandatory procedures for Major Defense Acquisition Programs (MDAPs) and Major Automated Information System Acquisition Programs (MAISAPs), Appendix VII Glossary of Key Terms and Concepts].

Sensitive: Unclassified information which warrants a degree of protection and administrative control and meets the exception criteria from mandatory public disclosure under the Freedom of Information Act. [Source: OSD Security Policy, 1996].

Software: "Instruction sets or programs necessary to operate an computational or logical device or system (e.g., computer programs, Programmable Logic Array (PLA) truth table)". [Source: Draft document "International Transfer and Export Control of Software and Software Source Code, Documentation, and Software Development Tools", Prepared by DUSD/TSPD, 01 January 1993].

Source Code: The Working Group defines source code to be any set of software instructions and data composed in a language higher than machine language (binary code). Source code is a form of software that is readily transformed into machine code (compiled or assembled). Source code is easier to read, understand and manipulate than machine code. It may be stored on paper, on magnetic disks, in circuitry or in other storage media. Source code can be written in assembly language or higher order languages.

Weapon System: "Items that can be used directly by the armed forces to carry out combat missions and that cost more than $100,000 or for which the eventual total procurement cost is more than $10,000,000. Such term does not include commercial items sold in substantial quantities to the general public (See Title 10, United States code, Section 2403, "Major weapon systems: contractor guarantees")" [Source: DOD

Directive Number 5000.2, Subject: Mandatory procedures for Major Defense Acquisition Programs (MDAPs) and Major Automated Information System Acquisition Programs (MAISAPs), Appendix VII Glossary of Key Terms & Concepts].

B. Documents and References

The following documents were used as sources of information in drafting this report. It should be noted that these documents were provided by the activities concerned with the release of software source code. No inference should be given to the validity or currency of these documents.

1. Draft document "International Transfer And Export Control Of Software And Software Source Code, Documentation And Software Development Tools", Prepared by DUSD/TSP (DTSA/TSPD), 01 January 1993.

2. Draft document "International Transfer And Export Control Of Software And Software Source Code, Documentation And Software Development Tools", Prepared by DUSD/TSP (DTSA/TSPD), 17 July 1992.

3. Draft document "International Transfer And Export Control Of Software And Software Source Code, Documentation And Software Development Tools", Prepared by DUSD/TSP (DTSA/TSPD), DPACT recommended modifications, November, 1992.

4. Electronics Industries Association "White Paper On Source Code Issues", August 14, 1992.

5. HQ USAF Release Policy For Foreign Disclosure Of Weapon/Defense system Software Technology, 13 Dec 1989.

6. Draft Policy Statement for Software Support of U.S Produced Avionic Systems (U) DSAA, 2 January 1990.

7. Technology Transfer And Security Assistance Review Board (TTSARB) Decision Memorandum On DON Policy On Release of Warfare System Software Technology to Foreign Countries, Case Number 9l-11, March 27, 1992.

8. DoN Policy on the Foreign Disclosure of Weapon/Defense System Software Technology.

9. International Traffic In Arms Regulations (ITAR), March 1996.

10. "Guidelines for Successful Acquisition and Management cf Software Intensive Systems: Weapons Systems; Command and Control Systems; Management Information Systems", September 1994, Volumes I and 2, Software Technology Support Center, Department of the Air Force.