EXPORT CONTROL OF SOFTWARE UNDER THE WASSENAAR ARRANGEMENT

FILIPPO BIGARELLA[*]

July 23, 2017

CONTENTS

ABSTRACT

The diffusion of surveillance software in the past 10 years has lead to an increased need for regulation. In 2013, the Wassenaar Arrangement included a set of items with the goal of controlling the export of surveillance software. In this paper, we explain the definitions provided by Wassenaar and offer a technical analysis on why such definitions might have unintended consequences which could impact security research.

[*] *Department of Information Engineering and Computer Science, Università degli Studi di Trento, Trento, Italy*

1 INTRODUCTION

The ever growing of computers and network devices all around the globe has allowed our society to evolve towards being largely information-based. As the activities performed change, so do the types of crimes that can be committed by malicious individuals. Therefore, law enforcement often need to obtain improved technologies in order to efficiently prevent these crimes or identify the criminals.

One of the categories of software emerged in the recent past is surveillance software. These are technologies developed in order to track and analyze the behavior and the information exchanged by individuals. Law enforcement agencies need to use these tools to obtain informations that would often not be accessible otherwise. When employed in a lawful, ethical way, this software represents an important instrument in fighting crime.

However, like with other powerful tools, surveillance software could also be used in nefarious ways if the intentions of the operators are not oriented towards protecting people. This is the case of oppressive regimes using surveillance software against their own people, in order to track activists and impede their freedom of speech.

Between 2012 and 2014, multiple episodes of surveillance abuse were exposed by CitizenLab, a group based at the University of Toronto that performs research on communication technologies and human rights [3]. From 2010 to 2012, Bahrain Government used Gamma Group's [6] FinFisher to monitor lawyers, activists, journalists and political opposition leaders [18] [15]. In 2014, the Remote Control System software produced by the Italian company Hacking Team [8], was found to be used against Ethiopian journalists [11].

Finally, in 2016, CitizenLab and Lookout exposed an attack performed on Ahmed Mansoor, a prominent human rights defender based in the United Arab Emirates. The uniqueness of its attack is represented by the extremely sophisticated nature of the malware used, called "Pegasus", and developed by Israeli cyber-arms dealer NSO Group, which exploited multiple, previously unknown vulnerabilities to take control of the target's iPhone [13].

The unlawful and unethical uses of surveillance software exposed the need for this category of technologies to be regulated. In order to prevent oppressive regimes from legally obtaining these technologies, the Wassenaar Arrangement included "Intrusion software" in its control lists in 2013, classifying it as a dual-use technology. Due to the restrictions imposed on the export of such tools, companies need to apply for an export license in order to legally trade them. This restriction has the goal of limiting such trade only between entities that can prove the software will be used for lawful and ethical purposes.

However, the definitions provided by Wassenaar have been strongly criticized by security researchers and professionals all over the world. Indeed, the language used in the control lists has unintended consequences that pose a threat to security research and development. By analyzing the definitions, we expose these consequences and we explain how they relate to security research. It will be important for security practitioners to closely follow and possibly influence the development of export control in order to help regulators outlining controls that have the main goal of protecting people.

OUTLINE The remainder of the paper is organized as follows. In Section 2 we introduce the Wassenaar Arrangement and explain how dual-use export controls are structured. In Section 3 we examine the relevant definitions provided by the control lists, which describe what items are controlled by Wassenaar. In Section 4 we analyze the definition of “Intrusion software" from a technological point of view, pointing out the potential flaws in the definitions. In Section 5 we describe how the restrictions imposed on these items may affect security research. We conclude in Section 6.

2 THE WASSENAAR ARRANGEMENT

In this section we introduce the Wassenaar Arrangement and explain how and why it relates to software.

2.1 Overview

The Wassenaar Arrangement on Export Controls for Conventional Arms and Dual-Use Goods and Technologies [10] is an international body, which aims to regulate the international trade of such items in order to build a more transparent and stable environment among the member states. The Arrangement was established on 12 July 1996 in Wassenaar, the Netherlands, by 33 founding members and it acts as a successor to COCOM (Coordinating Committee for Multilateral Export Controls), a group composed by NATO states that discussed the export of arms to non-NATO states. As of July 2017, the Wassenaar Arrangement includes 41 states.

The Wassenaar Arrangement provides two different control lists: the “Munition List" and the “List of Dual Use Goods and Technologies". The former includes items that are specifically designed for military use: therefore, it is not directly relevant to the topics discussed in this paper. The latter, as the name may suggest, covers a set of items that can have both a civilian and military use: this is the case for computers, telecommunications technologies and information security equipment.

One of the fundamental goals of the Wassenaar Arrangement is to prevent the proliferation and accumulation of goods that could be used to produce weapons of mass destruction (WMD), such as stopping chemical weapons precursors and uranium enrichment. Additionally, the Arrangement aims at controlling conventional arms and preventing the trade of such items with oppressive regimes, that might use them in a breach of civil rights.

2.2 Legal Structure

The control lists provided by the Wassenaar Arrangement serve as a reference for the participating states: indeed, such lists do not have binding legal power throughout the member states. The states need to implement the export controls through national laws, in order for them to have effect. Therefore, the decision to deny or allow for the export of any item is ultimately taken by each participating state. This is also referred to as the voluntary approach chosen by Wassenaar.

A plenary meeting is held every year among the participating states, during which possible updates to the control lists are discussed by their representatives. In order to perform modifications to the control lists, all the participating states must agree. As a result, the national implementations should be updated to be consistent with the newly provided lists.

2.2.1 *Dual-Use Regulation in the European Union*

Through the Council Regulation (EC) No 428/2009 of 5 May 2009 [14], the European Union provides a reference which can be used directly by member states without the need for them to implement export controls in their national laws. Such a legislative act is legally binding for member states, since it has an immediate and direct application in each one of them.

However, while each state can refer directly to the Dual-Use Regulation, the lists provided may not directly reflect the updates performed by the Wassenaar Arrangement, due to the different pace followed by the European Union in updating the Regulation.

2.3 Inclusion of Surveillance Technologies

As the technological panorama evolves throughout the years, new categories of items have surfaced that may have both a civilian and military use. In order to coherently adhere with its original goals, the Wassenaar Arrangement has updated the control lists to include such items.

Indeed, during the plenary meeting of December 2013, the Arrangement has included in the newly updated control lists a set of items to represent those technologies or equipments that could be used by different entities to produce and deploy surveillance and intelligence gathering tools.

As explained by the Department for Business Innovation & Skills in the United Kingdom, such controls have been introduced "because of real concerns about the use of such tools to breach human rights and the risks that they pose to national security" [12]. That is to say, the main goal behind the introduction of the aforementioned item is to regulate the trade of those technologies in activities that lead to the breach of human rights of the people involved [11] [15].

Specifically, two different categories have been introduced to regulate the export of surveillance technologies: "intrusion software" and "IP surveillance systems". However, the definitions provided in the control lists (discussed in Section 3) may not be adequate in correctly defining the scope necessary to reach the aforementioned goal.

3 CONTROLLED ITEMS UNDER THE WASSENAAR ARRANGEMENT

In this section we outline and discuss the definitions of the items concerning surveillance systems provided by the Wassenaar Arrangement. Additionally, we explain how the control lists relate to Open Source Software.

3.1 Definitions

In December 2013, a set of items has been added to the Wassenaar Arrangement control lists with the main goal of regulating the export of surveillance software. However, both the industry and the independent security research community expressed their disagreement with the broad scope of the definition provided in such document (as better discussed in Section 5).

Therefore, after a certain number of consultation rounds, the language used for some definitions has been slightly updated, to try and improve the scope of the definitions and better define class of software that should be regulated.

We now explore the item categories added in December 2013 to include surveillance software: however, the definitions taken into account reflect the most recent revision of the control lists provided by the Wassenaar Arrangement (February 2017) [21].

3.1.1 *Intrusion Software*

During the past decade, many different companies have surfaced whose business model is based on developing, selling and supporting software aimed at exploiting software vulnerabilities in order to surveil different targets [8] [6]. These technologies are then sold by the companies to different entities, including but not limited to foreign nation states, government agencies or companies.

The Wassenaar Arrangement tries to cover that kind of software through the definition of “Intrusion Software". The relevant sections in the List of Dual-Use Goods and Technologies covering “Intrusion Software" are 4.A.5., 4.D.4. and 4.E.1.c., which are quoted as follows:

> **4.A.5.** Systems, equipment, and components therefor, specially designed or modified for the generation, command and control, or delivery of "intrusion software".
>
> **4.D.4.** "Software" specially designed or modified for the generation, command and control, or delivery of "intrusion software".
>
> **4.E.1.c.** "Technology" for the "development" of "intrusion software".

In order to better understand the previously quoted item definitions, we need to focus on the concept of “Intrusion software" itself, which is defined as follows:

> "Software" specially designed or modified to avoid detection by ’monitoring tools’, or to defeat ’protective countermeasures’, of a computer or network- capable device, and performing any of the following:
>
> a. The extraction of data or information, from a computer or network- capable device, or the modification of system or user data; or
>
> b. The modification of the standard execution path of a program or process in order to allow the execution of externally provided instructions.

Additionally, the definition is supplemented by a note that defines certain exceptions of items that are not to be considered intrusion software:

> 1. "Intrusion software" does not include any of the following:
>
>   a. Hypervisors, debuggers or Software Reverse Engineering (SRE) tools;

> b. Digital Rights Management (DRM) "software"; or
>
> c. "Software" designed to be installed by manufacturers, administrators or users, for the purposes of asset tracking or recovery.

As analyzed by Bratus et al. [20], in order to regulate intrusion software, the Wassenaar Arrangement defines a conceptual framework composed of two different parts. The first part is what is included in the definition of “intrusion software": these items are not directly regulated. Instead, the control lists directly include all the technologies aimed at the “development", “generation, command and control, or delivery" of the intrusion software. These technologies constitute the items that are directly controlled by the Wassenaar Arrangement.

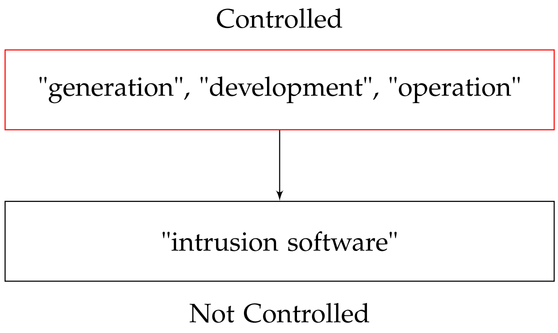

Figure 1: Conceptual framework used by Wassenaar.

This approach might seem to favor security research in spite of the development of commercial malware. However, as we show in Section 4, the extremely broad definition of intrusion software causes the inclusion of a large number of commonly used technologies that are essential in the development of general purpose software and to perform security research.

3.1.2 *IP Surveillance Systems*

Another category introduced by the amendments of December 2013 to the control list of the Wassenaar Arrangement is one under the name of “IP Surveillance System". This category of items is defined in section 5.A.1.j. as follows:

> **5.A.1.j.** IP network communications surveillance systems or equipment, and specially designed components therefor, having all of the following:
>
> 1. Performing all of the following on a carrier class IP network (e.g., national grade IP backbone):
>
>   a. Analysis at the application layer (e.g., Layer 7 of Open Systems Interconnection (OSI) model (ISO/IEC 7498-1));
>
>   b. Extraction of selected metadata and application content (e.g., voice, video, messages, attachments); and
>
>   c. Indexing of extracted data; and

> 2. Being specially designed to carry out all of the following:
>   a. Execution of searches on the basis of 'hard selectors'; and
>   b. Mapping of the relational network of an individual or of a group of people.

Similarly to Intrusion software, the definition is supplemented by a note defining certain exemptions of items that can not be defined as IP Surveillance systems:

> 5.A.1.j. does not apply to systems or equipment, specially designed for any of the following:
>
> a. Marketing purpose;
>
> b. Network Quality of Service (QoS);or
>
> c. Quality of Experience (QoE).

The scope of this definition is narrowed in particular by Section 5.A.1.j.2: indeed, the requirement of "Mapping of the relational network of an individual or of a group of people" is extremely specialized and denotes a very detailed design purpose.

While this paper primarily focuses on the definition of Intrusion software, some considerations can be performed on this definition.

On one hand, as opposed to Intrusion software, the approach chosen for this definition limits the amount of controversy that might originate from the introduction of this category, as the listed requirement are extremely specialized.

However, on the other hand, the provided definition does not try to cover all the possible technologies within this field that could be used to breach human rights. Moreover, it deliberately allows the use of Deep Packet Inspection for marketing purposes, which might constitute a privacy breach for the user of certain networks.

3.2 Open Source Software

In order to understand how the aforementioned definitions relate to Open Source Software, in particular the ones of "Intrusion software" and the affine technologies, we can analyze different notes provided in the control list [21]. In this context, by Open Source Software, we refer to software whose source code is publicly available, without any distinction on the licenses chosen.

Based on the Statement of Understanding at page 233, it is possible to determine that "software" and "technology" controls automatically extend to "source code", unless stated otherwise:

> Taking into account national practices and legislation, Participating States agree that "source code" items are controlled either by "software" or by "software" and "technology" controls, except when such "source code" items are explicitly decontrolled.

However, the final part of the General Technology Note at page 3, states that such controls do not apply if the technology is "in the public domain":

> Controls do not apply to "technology" "in the public domain", to "basic scientific research" or to the minimum necessary information for patent applications.

This fact is reinforced by the General Software Note at page 3, which extends that conditional inclusion to software, at point 2:

> The Lists do not control "software" which is any of the following:
>
> 1. Generally available to the public by being:
>   a. Sold from stock at retail selling points without restriction, by means of:
>   1. Over-the-counter transactions;
>   2. Mail order transactions;
>   3. Electronic transactions; or
>   4. Telephone call transactions; and
>
>   b. Designed for installation by the user without further substantial support by the supplier;
> 2. "In the public domain"; or
> 3. The minimum necessary "object code" for the installation, operation, maintenance (checking) or repair of those items whose export has been authorised.

As a last step, it is fundamental to understand how the "public domain" is defined by the control list. We can find a definition at page 215:

> This means "technology" or "software" which has been made available without restrictions upon its further dissemination.
> Note Copyright restrictions do not remove "technology" or "sof

Therefore, we can conclude that Open Source Software, as considered in this context, is not subject to the controls defined by the Wassenaar Arrangement.

4 INTRUSION SOFTWARE: A TECHNICAL ANALYSIS

In this section we discuss the definition of “Intrusion software", as provided by Wassenaar, from a technical point of view.

4.1 Standard Execution Path

As pointed out by Bratus et al. [20], one of the fundamental concepts in the definition of “Intrusion software" provided in the control list is the “standard execution path". Indeed, the “modification of the standard execution path of a program" is how the idea of intrusion is defined by the Arrangement. However, this definition does not offer a correct specification of the software constituting a surveillance implant.

First of all, it might not be possible to define a “standard" execution path for every kind of software. In fact, any software which implements different functionalities by the use of plugins does not have a common execution path across different installations or deployments. This fact is reinforced if such plugins can be developed by third party developers, not involved in the development of the original software. This is the case for many common software projects, such as the Apache [1] and nginx [9] Web servers, the Chrome [2] and Firefox [5] Web browsers, etc. .

Additionally, there exist many use cases of techniques based on modifying the execution paths of computer software, of which nefarious intrusion with the purpose of surveillance represent only one. This technique is more commonly referred to as *hooking* and it is used in many different software environments. Hooking can be used to inject code into running programs, for example to patch a security vulnerability, to add code used for instrumentation purposes or to upgrade such software, if it lacks an update mechanism or it cannot be stopped to perform that procedure. The *Detours* [17] library developed by Microsoft is an example of a framework to apply the hooking technique on the Windows operating system.

The mistake committed by the definition provided by Wassenaar is using a technique which may or may not be used by surveillance software, in order to characterize such software. This approach leads to the inclusion of non-malicious software and it is not possible to provide exceptions in order to exclude all the tools, some of which are essential to security research, as we explore in Section 5.

4.2 Defeat 'protective countermeasures'

Another fundamental concept for the definition of “Intrusion software" is the fact that the program should “defeat ’protective countermeasures’" in order to be classified as such. However, this wording does not characterize software in a meaningful way.

“Protective countermeasures" are defined in a technical note attached to the definition of intrusion software:

> ’Protective countermeasures’: techniques designed to ensure the safe execution of code, such as Data Execution Prevention (DEP), Address Space Layout Randomisation (ASLR) or sandboxing.

Specifically, this definition lists a set of mitigations commonly implemented in modern operating systems and sets the bypass of any of these techniques as a requirement for the classification of intrusion software.

However, this legal requirement may not hold as a technical requirement to have a functioning surveillance software: it is perfectly possible to develop remote monitoring software that does not bypass such mitigations. For example, the software might be installed by using a phishing attack [16] [13]. Additionally new dissemination techniques are constantly being deployed. As reported by Nagaraja and Anderson [19], social malware has already been used to target activists in Tibet.

Moreover, legitimate software may need to “defeat" such mitigations in order to implement its functionalities: as explained by Bratus et al. in their public comment [20], an example could be the case of *hot-patching* mechanisms. These techniques rely on modifying memory locations in order to update or fix running software. In order to apply such modifications, they need to scan and analyze the memory to identify the areas that need to be modified. It is possible to achieve that goal only by determining the exact location in memory of specific patterns, effectively overcoming the protection imposed by Address Space Layout Randomization (ASLR).

4.3 Controlled Items

As we explained in Section 3, the control lists of the Wassenaar Arrangement define a two-tiers conceptual framework regarding intrusion software. One tier is constituted by intrusion software itself, which is not directly regulated. The other tier includes all the technology needed for the “development" and “generation, command and control, or delivery" of intrusion software.

The over broad definition of intrusion software leads to an extremely generic definition for the technologies included in the second tier, whose export is therefore directly controlled. Arguably, compilers such as Clang [4] or GCC [7] could be included in the second tier, because they are used in the development and the generation of software that might be classified as intrusion software by the definition provided by Wassenaar.

5 IMPACT ON SECURITY RESEARCH

In this section we explore the impact on security research of the previously discussed controls.

5.1 Offense and Defense

Security research is usually categorized as either defensive or offensive. Defensive security research focuses on improving the security of software by fixing issues or developing mitigations against particular classes of attacks. Offensive security research has the goal of finding flaws in the software that can be potentially exploited to take control of a program or a system. Additionally, offensive security researchers and engineer develop the software to exploit those vulnerabilities.

Besides having a very different objective, these two fields of security research also differ in a more abstract way, and that is how their effectiveness can be verified. In offensive security, it is extremely easy to measure the effectiveness of a given approach or technology: by writing an exploit, the researcher is actually providing a proof of existence of a given vulnerability. On the other side, effectiveness in defensive security can only be measured by verifying what kind of attacks can be stopped.

Therefore, offensive research is essential for defensive security, as developing and creating new attacks is the most effective way to improve the security of software. Limiting offensive research would hinder the development of defensive security.

The definitions provided by Wassenaar would consider as directly controlled items certain technologies that are used both by surveillance software developers and defensive security researchers. It is very hard, if not impossible, to draw a line separating the tools based on the goal of the software that is being developed through their use.

5.2 Case Study: Fuzzers

A set of tools that might be problematic under the definitions provided by Wassenaar are *fuzzers*. Fuzz testing is a form of software testing that consists in randomly generating a large number of inputs and test whether the program’s behaviour is consistent when handling them. Many software vendors use fuzz testing in their development process. Some of them, such as Google and Microsoft, offer fuzzing infrastructures as a service to allow developers to find bugs in their software.

In security research, these tools are used both by offense and defense in order to find inputs that might cause a specific program to crash. Fuzzers play a fundamental role in the development of “intrusion software", as they allow security researcher to find the inputs that might allow for the “modification of the standard execution path of a program".

Therefore, this kind of tools would be a directly controlled item. However, limiting the development and the improvement of fuzzers may have catastrophic results in defensive security.

5.3 Impact on the Security Industry

Other activities in the security industry are also directly impacted by Wassenaar. When considering incident response and vulnerability triaging, for example, it could seem that the aforementioned two-tiers framework proposed by Wassenaar tries to facilitate communication among professionals. Indeed, it is architected in order to allow security professional to share malware samples, since “Intrusion software", as we explained, is not directly regulated.

However, sharing malware samples is often not enough to understand the behaviour of such malicious software: most of the times engineers need to develop tools and technologies that are used to analyze the malware. These tools would be directly controlled under Wassenaar, since they communicate with intrusion software. A company with teams working around the globe would need to apply for an export license for every country their workers are located in.

Another activity concerned by these restrictions is the use and the development of *penetration testing* tools. Penetration testing consists in simulating attacks against a certain architecture in order to evaluate its security. The people carrying out the attack are employed by the company controlling the architecture. All the identified security issues are then reported to the owner.

Under Wassenaar’s definition of intrusion software, it is not possible to distinguish penetration testing tools from nefarious surveillance software. The exemptions do not exclude such tools, as their scope can not be defined to be “asset tracking", and they are not part of any exempted category previously outlined. Therefore, once again, this leads to the inclusion of a much broader set of tools, that are essential in developing the technologies used to improve the security of many architectures in the software industry.

6 CONCLUSION

After a series of prominent unlawful and unethical uses of surveillance software, the Wassenaar Arrangement set out in 2013 to regulate the export of such technologies. The definitions provided in the control lists introduce a two-tiers conceptual framework for the restrictions of “Intrusion software". The items directly controlled under Wassenaar are all the technologies used to develop and generate intrusion software, which is not directly restricted. The motivation of this approach can be found in the desire of allowing security researchers to share intrusion software samples.

However, the over broad definition of intrusion software leads to the inclusion in the list of directly controlled items of a set of technologies whose usages can be orthogonal to aiding the development of intrusion software. This has been explained in this paper by technically analyzing the definitions and providing examples of technologies that are used to improve the security of software, and not only to develop sophisticated attacks.

Additionally, by explaining the mechanisms that regulate the relations between defense and offense in security research, we determined how problematic regulating offense would be. Indeed, as offensive technologies is the most effective tool to empirically verify the effectiveness of defensive research, restricting attacks would certainly mean stagnating the development and improvement of software security.

It is fundamental for security researchers to work with regulators in order to provide better definitions that can be concretely applied to restrict the export of surveillance software, without impacting security research.

BIBLIOGRAPHY

[1] Apache: HTTP server project. https://httpd.apache.org.

[2] Chrome. https://www.google.com/chrome/.

[3] CitizenLab. https://citizenlab.ca.

[4] clang: a C language family frontend for LLVM. https://clang.llvm.org.

[5] Firefox. http://www.wassenaar.org.

[6] Gamma Group. https://www.gammagroup.com.

[7] GCC, the GNU Compiler Collection. https://gcc.gnu.org.

[8] HackingTeam. http://www.hackingteam.it.

[9] nginx. https://www.nginx.com.

[10] The Wassenaar Arrangement. http://www.wassenaar.org.

[11] Bill Marczak, Claudio Guarnieri, M. M.-B. J. S.-R. Hacking team and the targeting of ethiopian journalists. https://citizenlab.ca/2014/02/hacking-team-targeting-ethiopian-journalists/.

[12] BIS. Intrusion software tools and export control, 2015. http://www.blogs.bis.gov.uk/exportcontrol/files/2015/08/Intrusion-Software-Tools-and-Export-Control1.pdf.

[13] Citizen Lab, L. Sophisticated, persistent mobile attack against high-value targets on iOS. https://blog.lookout.com/trident-pegasus.

[14] Council of European Union. Council regulation (EC) no 428/2009, 2009. http://eur-lex.europa.eu/eli/reg/2009/428/.

[15] Desmukh, F. Bahrain government hacked lawyers and activists with uk spyware. https://bahrainwatch.org/blog/2014/08/07/uk-spyware-used-to-hack-bahrain-lawyers-activists.

[16] Jagatic, T. N., Johnson, N. A., Jakobsson, M., and Menczer, F. Social phishing. *Commun. ACM 50*, 10 (Oct. 2007), 94–100.

[17] Microsoft. Detours. https://www.microsoft.com/en-us/research/project/detours/.

[18] Morgan Marquis-Boire, B. M. From bahrain with love. https://citizenlab.ca/2012/07/from-bahrain-with-love-finfishers-spy-kit-exposed/.

[19] Nagaraja, S., and Anderson, R. The snooping dragon: social-malware surveillance of the Tibetan movement. Tech. rep., University of Cambridge, Computer Laboratory, 2009.

[20] Sergey Bratus, D J Capelis, M. L. A. S. Why wassenaar arrangement's definitions of intrusion software and controlled items put security research and defense at risk—and how to fix it, 2014.

[21] The Wassenaar Arrangement. List of Dual-Use Goods and Technologies and Munitions List (WA-LIST (16) 1 Corr. 1), 2017. http://www.wassenaar.org/wp-content/uploads/2016/12/List-of-Dual-Use-Goods-and-Technologies-and-Munitions-List-Corr.pdf.