Cyber Weapons and Export Control: Incorporating Dual Use with the PrEP Model

Trey Herr[*] and Paul Rosenzweig[**]

Introduction

How do existing export control laws treat malware and cyber weapons, and what complications arise with their use? This paper presents a technically grounded framework to examine under what conditions malicious software components might be covered by the Export Administration Regulations (EAR) and International Traffic in Arms Regulations (ITAR). It presents the law in question and examines several key challenges in classifying and restricting the flow of cyber weapons.

When is malicious software ("malware") a weapon of war? When is it just an annoyance? Could malware or even cyber weapons become the subject of an arms control treaty? Can the United States limit the export of software that is a component of malware?

These and other questions are, today, inherently indeterminate. Increasingly, both theorists and practitioners have come to grips with the prospect of conflict utilizing digital tools in a supporting if not central fashion. Recent developments include the formation, in the United States, of a unified sub-command, known as Cyber Command, for the conduct of offensive and defensive missions in cyberspace.[1] More globally, a group of international experts have devoted years of effort to the compilation and publication of international legal norms that should govern conflict in the cyber domain,[2] and numerous multilateral entities, like NATO,[3] and the European Union,[4] have begun developing strategies and capabilities for conducting military operations using cyber weaponry.

Yet at the core of this ferment lies a vexing ambiguity: the world is developing an operational framework for cyber warfare without having any workable legal definition of what constitutes a cyber weapon. To date, our definition is an extensional one – we characterize a

[*] Senior Research Associate, George Washington University's Cyber Security Policy and Research Institute and fellow with New America's Cyber Initiative.

[**] Professorial Lecturer in Law, George Washington University, Washington, DC; Principal, Red Branch Consulting, PLLC; Distinguished Visiting Fellow, Homeland Security Studies and Analysis Institute.

[1] *U.S. Cyber Command Factsheet,* U.S. STRATEGIC COMMAND,http://www.stratcom.mil/factsheets/2/Cyber_Command/.

[2] TALLINN MANUAL ON THE INTERNATIONAL LAW APPLICABLE TO CYBER WARFARE (Michael N. Schmitt gen. ed. 2013) [hereinafter TALLINN MANUAL].

[3] Robert Lemos, *In Case of Cyber Attack: NATO Members Ready to Pledge Mutual Defense,* ARSTECHNICA, (Sept. 4, 2014), http://arstechnica.com/security/2014/09/in-case-of-cyberattack-nato-members-ready-to-pledge-mutual-defense/.

[4] NEIL ROBINSON ET AL., RAND CORPORATION, STOCKTAKING STUDY OF MILITARY CYBER DEFENCE CAPABILITIES IN THE EUROPEAN UNION (MILCYBERCAP) (2013), http://www.rand.org/pubs/research_reports/RR286.html.

cyber tool (such as malware) as a weapon if it is used in a warfare-like manner. Thus, for example, the drafters of the *Tallinn Manual* characterize a cyber weapon by the effects it may have, rather than by its nature or components, or means of operation or construction.[5]

But that sort of *ex post* definition is unsatisfactory for a host of reasons. The most salient, of course, is that *ex ante* decisions about the manufacture, sale, stockpiling, and use of particular tools cannot be dependent on a definitional determination that is made only after use. A host of possible policy responses, including, for example, arms limitations treaties, are critically dependent on the capacity to define a weapon with legal precision in a manner that can be employed at some stage prior to actual use. Yet, as we have said, no definition of a cyber weapon exists today other than a definition that operates only through practical application. Moreover, such a definition "may provide a basis for a more objective determination of the nature of activities in cyberspace,"[6] and would form the basis of a larger cyber weapons proliferation control enterprise.

The ambiguity of such a definition is compounded by the component nature of cyber tools. Like other tools, a particular version of malware is not unitary. Just as a gun is composed of many parts – a stock, barrel, and ultimately ammunition – so too, cyber tools have multiple elements of operation. Yet in seeking a cyber weapon definition it appears that many commentators fail to separate cyber tools into their component parts. Consequently, practical application is either over or under inclusive in determining the status of a particular tool and/or its component parts.

This article, we hope, will begin the process of resolving the underlying ambiguities. We start by using a technical, universal framework of the components of a cyber weapon. We characterize the weapon as necessarily containing three component parts – a Propagation method, an Exploit, and a Payload. For convenience we call this the PrEP framework.[7] We conclude that, at a technical level, the distinction between weaponry and non-weapon malicious software lies in the payload component of the tool, which must be capable of creating destructive digital or physical effects, but that the other components are essential aspects of the weapon that have a dual-use nature (that is, they equally have commercial, non-military uses). Our thesis is that the different components can and should be identified and managed using different policy frameworks and legal regimes.

To test our hypothesis, we take the PrEP framework and apply it in a classic area of law where an *ex ante* definition of weaponry and dual use technology already exists in a robust form: the area of export controls from the United States. As we discuss below, that regulatory system recognizes a distinction between tools or products that have dual use functions and those that are

[5] TALLINN MANUAL, *supra* note 2, at Rule 13, Commentary ¶ 4.

[6] Gary D. Brown & Andrew O. Metcalf, *Easier Said Than Done: Legal Reviews of Cyber Weapons*, 7 J. NAT'L SEC. L. & POL'Y 115, 128 (2014).

[7] Trey Herr, *PrEP: A Framework for Malware & Cyber Weapons*, 13 J. INFO. WARFARE 13 1, 87-106 (2014).

clearly military in nature without significant commercial utility. After examining the underlying law briefly, we demonstrate that the PrEP framework of cyber components is adaptable to the existing legal rules and maps readily into current restrictions.

As a result of that analysis, we conclude that a cyber weapon requires all three components to be defined as a military tool. But we also conclude that the subcomponents are subject to differential legal regulation based on their nature. The propagation method cannot be regulated as a discrete good but may be restricted through other legal measures intended to target criminal infrastructure. Exploits are inherently of a dual-use nature and could therefore be subject to export controls of the sort that govern this technology. By contrast, the analysis suggests that the payload component, where designed to create destructive effects, is inherently military in nature and as a policy matter may reasonably be subject to greater export control.

I. The PrEP Model

All malicious software can be thought of as a combination of three separate components: a propagation method, exploits, and payload.

- A propagation method is the means of transporting malicious code from origin to target. This could be as simple as a mass email for spear phishing attacks or as complex as carefully crafted "dropper" software.
- Exploits act to enable the propagation method and payload's operation, allowing attackers to take control of a piece of software or entire computing system.
- The payload is code written to achieve some desired malicious end such as to delete data or manipulate an industrial control system (ICS).

The three components each work in concert but have substantially different roles. The botnet infrastructure employed as a propagation method in spreading email attachments, for example, consists of tens of thousands of zombie computers under the control of a small group of people, and it is quite different from the payload being distributed: code contained in the attachment. The propagation method spreads the malicious tool while the payload is written to create some effect on a computer system. The exploit involved in hacking a common piece of software like Internet Explorer may be a tiny piece of code, written to take advantage of a flaw in a browser, and opens the door for a payload but does not achieve anything malicious by itself. Without one or several exploits, a payload would almost never be able to execute on a target computer. These exploits serve to manipulate the target system into giving malicious code access and user privileges in order to function. Each of the three PrEP components has a distinct purpose and works in combination with the others to create malware.

The graphic below summarizes all three components of the PrEP framework and their relationship to a potential target:

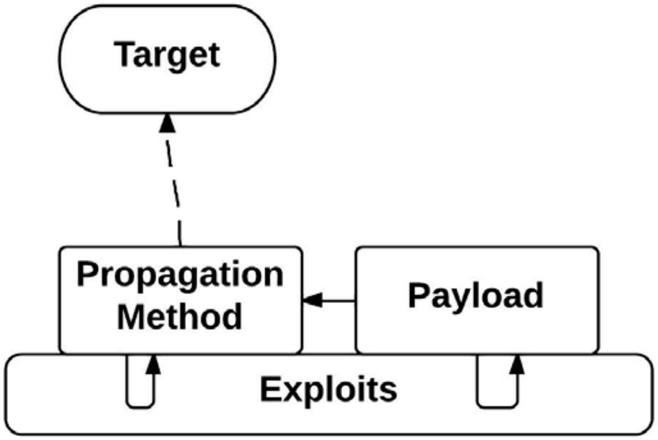

Figure 1: PrEP Framework

Both the propagation method and payload are enabled by a set of associated Exploits, which open the door for the other two components. All three are required to identify a collection of software as malware or a cyber weapon however.

This modular approach is not entirely new. The first edition of *Hacking Exposed* in 1999 laid out an "Anatomy of the Hack," a process-oriented or phase-based model that described the steps a hacker might take in executing an attack on some hypothetical target (including gathering intelligence in the Footprinting and Scanning steps and propagating code in the Gaining Access step).[8] More than a decade later, researchers at Lockheed Martin revised the phase-based model with a paper on the cyber "kill chain" that outlined major steps in the cyber-attack process, including "Weaponization" and "Actions on Objectives," and identified points along the chain at which defenders could interrupt an ongoing attack by persistent threats, labeled APTs.[9] The PrEP framework focuses not on process, but on the characteristics of the tools being used. This approach, suggesting that all malware can be conceptualized as the combination of three components, builds on previous work that delineated between software tools intended for access and those intended for malicious compromise of computer systems.[10]

A. Propagation Method

[8] *See generally* STUART MCCLURE, JOEL SCAMBRAY & GEORGE KURTZ, HACKING EXPOSED: NETWORK SECURITY SECRETS AND SOLUTIONS (1st vol. 1999).

[9] Eric M. Hutchins, Michael J. Cloppert & Rohan M. Amin, Intelligence-driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains, Address before the International Conference on Information Warfare and Security (Mar. 17-18, 2011), *in* PROC. 6TH INT'L CONF. INFO. WARFARE AND SEC. (2011), *available at* http://www.lockheedmartin.com/content/dam/lockheed/data/corporate/documents/LM-White-Paper-Intel-Driven-Defense.pdf.

[10] *See* Herbert S. Lin, *Offensive Cyber Operations and the Use of Force*, 4 J. NAT'L SEC. L. & POL'Y 1, 63 (2010).

Propagation is the act of delivering code to the target system; it could constitute an email attachment, compromised web site, or USB stick to jump a physical air gap between computers. It may have multiple stages and could include a so-called "dropper," or code written specifically to land on a target system and phone home to a designated IP address to download other components. This flexibility comes from propagation being the least software intensive piece of malware. Anything that can hold or transmit data can propagate malware. A variety of propagation methods have been observed including compromised websites,[11] email attachments,[12] and removable storage media.[13]

Propagation methods can be classified according to their scale and specificity. Scale determines the total possible target pool, i.e. how many computers and devices from the global population are accessible. Malware which propagates over the internet, for example, is likely to be found much farther afield than malware spread over compromised storage media. The scale of a compromised web site may be tremendous (if the site in question is Google) or tiny (if it is an old GeoCities page). Contemporary botnets such as Kelihos have up to tens of millions of slave machines and present an excellent means of propagating to targets indiscriminately.[14] A propagation method which targets all internet connected computers (large scale) is thus different from one which targets only users connected to a particular company's network (small scale).

Specificity measures the targeting constraints placed on a weapon, determining how much of the possible target pool is of interest or 'active.' These could be technical limitations, focusing on a particular operating system or software version, or based on personal information like account credentials or the presence of certain filenames. Phishing and spear phishing techniques demonstrate the difference in specificity: phishing emails are targeted at any recipient while spear phishing emails contain personal details or relevant information about a particular recipient. Specificity can help to contain the spread of malware infections, lowering the likelihood of detection and limiting defensive response. This capability is especially important for espionage malware from actors like Unit 61398, a Chinese PLA group featured in Mandiant's report on APT 1 in 2013.[15] This group was effective in that it was able to operate undetected and exfiltrate data from companies. Specificity matters where targets are substantively distinct. For malware trying to build a botnet, for example, one computer is largely like any other. Espionage malware on the other hand, is typically concerned with a particular group of

[11] Michael Mimoso, *Watering Hole Attack Hits US Department of Labor Website*, THREAT POST (May 1, 2013, 4:30 PM), http://threatpost.com/watering-hole-attack-claims-us-department-of-labor-website/100081.

[12] Tillman Werner, *Botnet Shutdown Success Story: How Kaspersky Lab Disabled the Hlux/Kelihos Botnet*, SECURELIST (Sept. 28, 2013, 11:53 PM), http://www.securelist.com/en/blog/208193137/Botnet_Shutdown_Success_Story_How_Kaspersky_Lab_Disabled_the_Hlux_Kelihos_Botnet.

[13] Sergei Shevchenko, *Agent.btz - A Threat That Hit Pentagon*, THREAT EXPERT BLOG (Nov. 30, 2008), http://blog.threatexpert.com/2008/11/agentbtz-threat-that-hit-pentagon.html.

[14] Brian Krebs, *Researchers Clobber Khelios Spam Botnet*, KREBS ON SECURITY (Mar. 28, 2012), http://krebsonsecurity.com/2012/03/researchers-clobber-khelios-spam-botnet/.

[15] *See* MANDIANT, APT1: EXPOSING ONE OF CHINA'S CYBER ESPIONAGE UNITS 2-3, 6 (2013), http://intelreport.mandiant.com/Mandiant_APT1_Report.pdf.

companies or individuals; the Chinese group profiled by Mandiant primarily targeted defense contractors and companies with access to advanced military technology.

Stuxnet, a prominent example of cyber weaponry, was designed to impede Iran's uranium enrichment operations and included a two-stage propagation method as computers at the targeted facility were networked together but likely not directly connected to the internet.[16] A separate set of PCs, running Windows and configured to communicate directly with the Siemens-built Programmable Logic Controllers managing the centrifuges, were air-gapped (physically separated) from all other computers, with no direct link to the Internet.[17] The first propagation stage, including two exploits targeting system vulnerabilities not yet publically disclosed (so-called "zero-day" or "0day" exploits), manipulated each computer's network services into replicating the malware to all connected computers. In all, this first stage propagation had at least five different propagation techniques associated with it, including a peer-to-peer communications and update protocol as well as the Microsoft local print network software.[18] In order to jump the gap to those computers with access to the centrifuges, Stuxnet had a second propagation method, employing a small software package that lived on removable USB sticks and would infect all machines with which it came into contact. This allowed Stuxnet to jump between computers even when they were kept physically separate.[19] The coordinated efforts of the local network and USB stick propagation methods proved highly effective in delivering Stuxnet's payload.

Malware's propagation method serves as a delivery vehicle for the payload and associated exploits. Stuxnet relied on small-scale methods like local area networks and removable storage media, to spread, focused on a limited number of computers, mostly in the Middle East, and activated only where it found the proper version of a Siemens control system present.[20]

B. Exploits

Aiding the propagation method and payload are software programs called exploits that take advantage of vulnerabilities in computer systems or surrounding networks. Vulnerabilities are aspects of a computer system which allow third parties to effect unintended operations.[21] Exploits typically target the operating system, key applications such as the browser, or the firmware of a computer or

[16] NICOLAS FALLIERE, LIAM O. MURCHU & ERIC CHIEN, SYMANTEC, SYMANTEC SECURITY RESPONSE: W32.STUXNET DOSSIER 3 (2011), http://www2.gwu.edu/~nsarchiv/NSAEBB/NSAEBB424/docs/Cyber-044.pdf.

[17] Ralph Langner, *Stuxnet: Dissecting a Cyberwarfare Weapon*, 9 SEC. & PRIVACY, IEEE 49 (2011) [hereinafter Langner, *Stuxnet*].

[18] Falliere, Murchu & Chien,*supra* note 16, at 25.

[19] Langner, *Stuxnet*, *supra* note 17, at 49.

[20] RALPH LANGNER, TO KILL A CENTRIFUGE: A TECHNICAL ANALYSIS OF WHAT STUXNET'S CREATORS TRIED TO ACHIEVE 11 (2013).

[21] PETER MELL & TIMOTHY GRANCE, NATIONAL INSTITUTE OF STANDARDS & TECHNOLOGY, THE NIST DEFINITION OF CLOUD COMPUTING 7 (2011), http://csrc.nist.gov/publications/PubsSPs.html (follow "SP 800-145" hyperlink).

network component to provide the digital grease necessary for other components of malware to operate successfully.

Vulnerabilities may be purposefully included features or simple bugs in otherwise functional code. Their presence does not affect system operation but instead acts as a narrow window through which an exploit may be written. For example, a program that expects to retrieve a static image file but fails to check the supplied file type might return an executable software program instead. The procedure to retrieve an image was intentional but failing to check the file type allows a third party to execute malicious software. The Love Letter virus of 2000 relied on the fact that Windows 2000 and XP hid known file extensions when parsing file names from the right to the left. The virus file hid itself, an executable program, outside the filename which ended in .txt - LOVE-LETTER-FOR-YOU.TXT.vbs.[22] While this design convention did not constitute a 'flaw' per se, it was used by third parties to effect unintended operations in the software. Vulnerabilities may also be introduced directly to hardware through compromises in chip design or manufacture somewhere along the supply chain.[23]

Part of the Stuxnet propagation method used an exploit in the Windows Print Spool network service to spread between computers. From an infected machine, Stuxnet would submit a specially formatted print request to another, uninfected, machine. Instead of supplying data to print, the request would trigger a remote procedure call, injecting Stuxnet's code from the original infected computer over the print network to the target machine.[24] Here the propagation method was the network-based remote procedure call, but the exploit was necessary to take advantage of a vulnerability in the Windows print software.

Exploits are code, written to take advantage of these features or errors in software and enable the operation of other malware components, either the propagation method or payload. Failing to distinguish the two can cause analytic confusion. Indeed, some literature combines the payload's function into the exploit and uses the combination as a verb, to 'exploit' a system.[25] This approach limits our understanding of the variety of payload types and functions employed by malware authors, apart from the rapidly growing market for software vulnerabilities. The two are logically distinct both in sequence and form - exploits are written to a particular vulnerability present in target software while payloads are written to achieve a particular effect. They reveal different types and quantities of information about potential targets and are used by malware differently.

Distinguishing between the two becomes especially important when thinking about their development. A payload is written to achieve a desired effect so its focus is on the code's output. An

[22] *Information about the VBS.LOVELETTER Worm Virus*, MICROSOFT (Jan. 29, 2007), http://support.microsoft.com/kb/282832.

[23] *See generally* Georg T. Becker et al., *Stealthy Dopant-Level Hardware Trojans*, *in* 4 J. CRYPTOGRAPHIC ENGINEERING 19, 21 (2014).

[24] Falliere, Murchu & Chien, *supra* note 16, at 2.

[25] Peter Bright, *Massive SQL Injection Attack Making the Rounds—694K URLs So Far*, ARSTECHNICA (Mar. 31, 2011, 7:54 PM), http://arstechnica.com/security/news/2011/03/massive-sql-injection-attack-making-the-rounds694k-urls-so-far.ars.

exploit is written to a particular target, focusing on the target software's structure and function. This distinction, between writing code to the target and writing to the effect, make for divergent practices in developing, selling, and integrating exploits and payloads. Exploits are highly fungible, and can be integrated with different malware components depending on need. Payloads are more time intensive to repurpose; though still code, they were written to achieve a narrow range of effects. This difference helps underline the role exploits play in malware, opening the door for malicious code to propagate to and execute on a target system.

Exploits can be sorted into several distinct categories:

- Access
- Escalation of Privileges
- Code Execution

Each category functions to support a different component of the PrEP model; *access* exploits work with the propagation method while *escalation* and *code execution* exploits are typically employed by the payload.[26] Categorizing exploits in this functional manner can help create context and detail for exploits beyond current classification methods which are typically related to discovery date and target. The most popular means of differentiating exploits is to determine which, if any, are zero-days that take advantage of vulnerabilities present since the software's launch -- but this has nothing to do with the exploit, only our collective awareness of its existence. The Common Vulnerability Scoring System (CVSS), which rates vulnerabilities on a 1-10 scale based on several factors, is an improvement though not without limitations.[27]

Exploits are the most commodity-like component in malware; bought and sold on the web, their need to be specific to a target system creates tremendous value with quoted prices ranging into the hundreds of thousands of dollars.[28] Increasingly, companies such as VUPEN and a sizable collection of freelancers are selling newly discovered vulnerabilities and developed exploits to governments, including the NSA, and non-state actors rather than the original software vendors like Google.[29]

[26] MIKKO HYPPONEN, *The Exploit Marketplace*, *in* THE FOG OF CYBER DEFENCE 231-234, 232 (Jari Rantapelkonen & Mirva Salminen eds., 2013), *available at* http://www.doria.fi/handle/10024/88689.

[27] Patrick Toomey, *CVSS-Vulnerability Scoring Gone Wrong*, NEOPHASIS LABS SECURITY BLOG (Apr. 25, 2012), http://labs.neohapsis.com/2012/04/25/cvss-vulnerability-scoring-gone-wrong/; Carsten Eiram, et. al., *The CVSSv2 Shortcomings, Faults, and Failures Formulation*, RISKED BASED SECURITY (Feb. 26, 2013), https://www.riskbasedsecurity.com/2013/02/cvssv2-shortcomings-faults-and-failures-formulation/ (follow "The full letter may be accessed here." hyperlink).

[28] Andy Greenberg, *Shopping For Zero-Days: A Price List For Hackers' Secret Software Exploit*, FORBES (Mar. 23, 2012), http://www.forbes.com/sites/andygreenberg/2012/03/23/shopping-for-zero-days-an-price-list-for-hackers-secret-software-exploits/.

[29] Danielle Walker, *NSA Sought Services of French Security Firm, Zeroday Seller Vupen*, SC MAG. (Sept. 18, 2013), http://www.scmagazine.com/nsa-sought-services-of-french-security-firm-zero-day-seller-vupen/article/312266/; David Fidler, *Zero-Sum Game: The Global Market for Software Exploits*, ARMS CONTROL LAW (July 18, 2013), http://armscontrollaw.com/2013/07/18/zero-sum-game-the-global-market-for-software-exploits/.

C. Payload

A payload is the core content of malware: malicious software designed to execute on a computer system and achieve some predefined goal such as compromising password files or deleting data. It could take advantage of code libraries present on a target system in order to execute and may combine several modules, each with a different but complementary purpose. The payload is malware's *raison d'être* and can vary widely in sophistication. The amateur Wank worm, which infected computers running Digital Equipment Corporation (DEC)'s VMS operating system in the late 1980s, executed a simple operation to change user's passwords and display an anti-nuclear slogan. Stuxnet's payload by contrast, was designed to manipulate machinery into destroying itself.[30] The payload of a piece of malicious software executes on a computer system in order to create some effect; to alter data, remotely activate a camera, create a 'backdoor' for future access, or damage hardware - these actions manipulate the intended function of an information system to achieve the effects desired by an attacker.

These payloads are often what come to be identified as malware - we think of those intended for financial fraud or to perpetrate distributed denial of service (DDOS) attacks. Payloads may have one or several different modules, each with a different function, such as espionage or creating destructive effects, which require differing levels of technical expertise and resources. Stuxnet's payload was designed to damage physical equipment without alerting nearby staff. One module infected control software for Iranian enrichment centrifuges to change their rotation speed while a second payload module opened and closed valves controlling the flow of uranium hexafluoride gas to other centrifuges, disrupting the multi-stage enrichment process.[31] Both payload modules were designed to speed the growth of metal fatigue in, and eventually break, the centrifuges. Stuxnet also had an obfuscation module that took normal diagnostic information on the centrifuge rotation speed and fed it into the machine's control software so that nothing would appear amiss until the machines had broken down.[32] This deception of facility staff was critical to Stuxnet's operation, as the destructive payload required several months to achieve its effect.

A malware's payload is the core functionality: malicious software that creates a desired effect on a target computer. Payload modules tend to fall into a number of different categories, the design of which vary in terms of technical sophistication and resources required to develop. Spread by a propagation method, the payload is enabled by a set of exploits that help inject it into the target system and execute.

II. Applying the PrEP Model to Export Controls

[30] *WANK Worm on SPAN Network*, CERT (Oct. 17, 1989), http://www.cert.org/historical/advisories/ca-1989-04.cfm.

[31] DAVID ALBRIGHT, PAUL BRANNAN & CHRISTINA WALROND, INSTITUTE FOR SCIENCE AND INTERNATIONAL SECURITY, STUXNET MALWARE AND NATANZ: UPDATE OF ISIS DECEMBER 22, 2010 REPORT 7 n.17 (2011), http://isis-online.org/uploads/isis-reports/documents/stuxnet_update_15Feb2011.pdf.

[32] LANGNER, TO KILL A CENTRIFUGE, *supra* note 20, at 8-10.

To test the strength and utility of the PrEP framework, we have imagined how the definition would apply in the real world context of export controls under U.S. law. If the framework were to produce unusual or counter-intuitive results when matched to an existing, stable, legal structure, we would have doubts about the accuracy or utility of the model. Conversely if our PrEP framework maps well onto existing export control law, we might consider such mapping a confirmation of the model's validity.

A. An Introduction to EAR and ITAR

Of course, if we are going to match the PrEP framework against existing export control law and regulation, a working knowledge of those laws and regulations is necessary. What follows is an abbreviated overview of American export control law – with apologies to practitioners who will see in this summary a number of generalizations and simplifications that, while generally true, are subject to many exceptions, exemptions and limitations which we happily elide for the sake of clarity.

To begin, exports are regulated and controlled under two distinct regulatory systems: the Export Administration Regulations (EARs), administered by the Department of Commerce,[33] and the International Traffic in Arms Regulations (ITARs), administered by the Department of State.[34] The EARs regulate commercial items for export and also items that might be considered "dual use," or capable of both commercial and military uses. By contrast, the ITARs regulate the export of "military articles" and information and defense services.

The EARs operate through what is known as the Commerce Control List (CCL).[35] Items are added to the list if they have a dual use and are potentially exploitable as an instrument of military technology but also have commercially significant uses. For items on the CCL, American exporters require a license that permits the shipment of the goods. The same structure applies in the ITARs. Items are added to the U.S. Munitions List (USML) if they are considered a defense article,[36] and may not be exported without a license.

It is worth noting that under both the EARs and the ITARs the term "export" is defined – and it encompasses both the actual shipment or transmission of an item outside the United States or, a status known as a "deemed export." This latter category consists of release or disclosure of the item or information to a foreign national, whether within or outside the United States through means like writing or a private conversation. In addition, EAR does not distinguish between software transferred via physical shipment and digital distribution.[37]

[33] Export Administration Regulations, 15 C.F.R. §§ 730-774 (2014).

[34] International Traffic in Arms Regulations, 22 C.F.R. §§ 120-130 (2014).

[35] Commerce Control List, 15 C.F.R. § 774, Supp. 1 (2014).

[36] *See* 22 C.F.R. § 121.1 (2014).

[37] Jeffrey Richardson, *Is Your Software Transmission Subject to U.S. Export Controls under the EAR?*, MILLER CANFIELD (May 3, 2013), http://www.millercanfield.com/resources-alerts-845.html ("The EAR . . . makes no

Though there is much additional detail to the structure of the EARs and ITARs, it is, for the most part, irrelevant to our consideration in this paper. Rather the critical question is to dig down in detail on precisely which products make the CCL or USML and are therefore subject to regulation. As with the questions posed initially in this paper, the application of the EARs and ITARs turns on a threshold definition that delimits the scope of regulation.

Sadly, however, that exercise is to some degree tautological. Items are subject to the EARs and ITARs if they are listed as controlled on the CCL or USML; yet, the standards for inclusion on those lists is remarkably imprecise and indefinite. Here, for example, is how the EAR is described in regulation:

> The export control provisions of the EAR are intended to serve the national security, foreign policy, nonproliferation of weapons of mass destruction, and other interests of the United States, which in many cases are reflected in international obligations or arrangements. Some controls are designed to restrict access to items subject to the EAR by countries or persons that might apply such items to uses inimical to U.S. interests. These include controls designed to stem the proliferation of weapons of mass destruction and controls designed to limit the military and terrorism support capability of certain countries. The effectiveness of many of the controls under the EAR is enhanced by their being maintained as part of multilateral control arrangements. Multilateral export control cooperation is sought through arrangements such as the Nuclear Suppliers Group, the Australia Group, and the Missile Technology Control Regime. The EAR also include some export controls to protect the United States from the adverse impact of the unrestricted export of commodities in short supply.[38]

The devil, then, is in the application of this broad statement of purpose to more particular EAR requirements. With respect to software (except encryption software, which is subject to special rules),[39] the Department of Commerce has generally determined that publicly available software does not meet the standards for listing on the CCL and is therefore not subject to control.[40] Software and information is "published" when it is available for general distribution, either for free or at a price that does not exceed the cost of reproduction and distribution. The converse of this is that most non-public software of the sort we consider in this paper is putatively subject to the EAR licensure requirements and thus eligible for inclusion on the CCL.

The ITARs are perhaps slightly less tautological in their definition, but not especially so. An article may be listed on the USML if it:

distinction between software transferred through the physical shipment of tangible items and electronic transmissions.").

[38] 15 C.F.R. § 730.6 (2014).

[39] *Id.* § 742.15.

[40] *Id.* § 734.7.

> (a) Is specifically designed, developed, configured, adapted, or modified for a military application, and (i) Does not have predominant civil applications, and (ii) Does not have performance equivalent (defined by form, fit and function) to those of an article or service used for civil applications; or
>
> (b) Is specifically designed, developed, configured, adapted, or modified for a military application, and has significant military or intelligence applicability such that control under this subchapter is necessary.[41]

Thus, the central question becomes one of the specificity of the military purpose.

B. Cyber Weapons and the EAR and ITAR

All cyber weapons are malware, but very few pieces of malware are cyber weapons. Under the PrEP framework, cyber weapons still require all three components to function but are differentiated from malware by a payload designed to create destructive digital or physical effects. Destructive digital effects damage the integrity or availability of an information system - deleting data or disrupting a critical network service. To label software as having a military use, it must create or tangibly support the deployment of destructive effects. These could be short term, where deleted data is restored from backup, or near permanent, where a payload is designed to damage a device's firmware. Destructive physical effects manipulate a piece of equipment, like a centrifuge or generator, causing permanent damage or destruction. The Aurora test at Idaho National Labs deployed a cyber weapon into the industrial control system of a multi-ton generator causing it to shake on its foundations, eventually destroying the machine.[42] Destruction can amount to physical damage or loss of data integrity such as deletion or corruption.

Importantly, espionage tools and other malware that create loss of confidentiality should not be considered weapons; while the internet and networked information systems have proven a tremendous boon for intelligence collection and information theft, using software to compromise and steal data is no more the use of a weapon than stealing hard copy files. Including espionage in any weapons definition creates something too broad to be useful. Only malware that includes a payload to create *destructive* effects should be considered a cyber weapon. Using this definition of a weapon, the sections below evaluate the relative dual-use and military applicative nature of each of the PrEP components in turn, followed by their combination as a cyber weapon.

Propagation Method

A propagation method is the means of conveying code to a target from the attacker. It may be a physical device like a compromised USB stick or a web-based technique like the zombie machines of a botnet. A propagation method is not a tradable good but rather a form of

[41] 22 C.F.R. § 120.3 (2014).

[42] Lori A. Burkhart, *Cyber Attack! - Lessons Learned: Aurora Attack*, FORTNIGHTLY, Jan. 2008, at 1, http://www.fortnightly.com/fortnightly/2008/01/cyber-attack-lessons-learned-aurora-attack.

infrastructure. For web-based distribution, this may be a legitimate network like the Window's Update service or illegitimate like the Kelihos botnet. Where malware is distributed over the internet, as is usually the case, the commodity to be sold or rented is access, e.g., access to a botnet or a compromised web server.

This fact alone makes application of export control regulations difficult. It would be possible to treat access to a botnet as a service, but given that construction of such infrastructure is already illegal, it is likely to do little more to dissuade their use.[43] This could easily overlap with legitimate uses as well; many software providers build in a propagation method to their software for updates and new releases. The recognition of a botnet as infrastructure requires a legal approach more akin to regulating the sale of property rather than exporting goods, where the legal object is seizure of property rather than restrictions on the transfer of goods. The propagation method thus is a form of infrastructure not easily traded or exported and should not be considered under the CCL or USML.

Exploits

Exploits are code written to take advantage of a vulnerability in a piece of software, and they are the critical enabling component of malware and cyber weapons – they provide the basis for access and the ability to execute malicious code. Exploits pose a challenge to law as they present a compelling set of competing interests to balance.[44] On one side is the penetration testing (pen-testing) community who use tools that identify vulnerabilities, and execute exploits to take advantage of them, in order to test client system's security and identify gaps in protection. This "white hat" hacking is a healthy, legal, and necessary part of the larger information security process.

Opposite of these defensive security efforts are states and non-state groups who use exploits to gain access to third party computer systems in order to steal information or cause damage. The exploits they use may be the same as those employed in pen-testing and indeed, it is in the information security community's interest to identify as many avenues of attack as possible. The ability to differentiate between malicious and legitimate exploits is only really possible when they are combined with a propagation method or payload – only these components can effectively be deemed malicious or not prima facia.

Exploits then represent the ultimate dual use good; written for legitimate purposes, exploits can be used as part of pen-testing to assess the security of computers and networked information systems. Used maliciously, exploits are a critical enabling tool for malware and cyber weapons. Exploits should be considered dual use items and thus placed on the CCL

[43] *See* 18 U.S.C. § 1030 (2012) (criminalizing "unauthorized access" of a computer).

[44] *See generally* Jennifer Stisa Granick, *The Price Of Restricting Vulnerability Publications,* 9 INT'L J. COMM. L. & POL'Y 1 (2005).

because of their applicability to a wide range of activities, both legitimate and illegal. Defining the particular guidelines for vendors on how to implement such sweeping controls, however, is a thorny problem that will rest on the particular language and definitions used; as such, it is beyond the immediate scope of this paper.[45]

Payloads

The payload of a piece of malware or a cyber weapon is the means to create some desired effect on a target computer; payloads that do not cause destructive effects should not be considered under either of the two export control regimes. Including non-malicious software, such as a calculator program or code that changes the color of a computer background, risks ensnaring legitimate penetrating testing tools under a restrictive legal regime. Less flippant but equally non-lethal examples can be found in espionage or surveillance software like the core of Hacking Team's Galileo Remote Control System, which serves as a distributed surveillance platform targeting desktop and mobile operating systems.[46] While this sort of software can be and often is used by repressive regimes to target dissidents,[47] surveillance and espionage malware does not cause destruction to information or physical object and thus cannot be covered under anything but an unreasonably expansive reading of current U.S. export control law. Software cannot be considered a threatening or harmful item without the capability to cause destructive effects. Thus, we restrict our analysis to payloads that are capable of causing damage to digital or physical systems, in conjunction with a propagation method and exploits.[48]

[45] By way of outline, current implementation of prohibitions on software "weapons" is through an ad-hoc methodology. To the extent considered at all, malware is listed in a miscellaneous category (e.g. Category XXI under ITAR). Firms who want greater clarity before exporting an item may seek review by the US government prior to export.

[46] Joseph Cox, *The FBI Spent $775K on Hacking Team's Spy Tools Since 2011*, WIRED.COM (July 6, 2015, 3:00 PM), http://www.wired.com/2015/07/fbi-spent-775k-hacking-teams-spy-tools-since-2011/.

[47] *See HRW Says Saudi Govt Targeting Dissidents with Malware*, YAHOO NEWS, (June 27, 2014), http://news.yahoo.com/hrw-says-saudi-govt-targeting-dissidents-malware-210153373.html; Nicole Perlroth, *FinSpy Software Is Tracking Political Dissidents*, N.Y. TIMES (Aug. 30, 2012), http://www.nytimes.com/2012/08/31/technology/finspy-software-is-tracking-political-dissidents.html.

[48] For those who see espionage as a means of war, and espionage tools as weapons, this means that the scope of malware payloads which could be classified as a military item under ITAR or EAR is broader than we contemplate here – but it does not change the overall analysis we offer. For reasons we discuss in the text, we think such an expansion would strain ITAR and EAR beyond their present form. For more on the definition of "destructive" cyber weapons, see Herr, *supra* note 7; Robert Fanelli & Gregory Conti, *A methodology for cyber operations targeting and control of collateral damage in the context of lawful armed conflict*, *in* PROCEEDINGS OF THE 4TH INTERNATIONAL CONFERENCE ON CYBER CONFLICT 319-331, 322 (NATO CCD COE Pub. 2012) ("'Cyber weapons,' and those wielding them, must be capable of operating in accordance with the principles of *jus in bello*. This entails the capability to direct effects at valid military targets using controlled amounts of force and to minimize collateral damage."); Thomas Rid & Peter McBurney, *Cyber-weapons*, 157 RUSI J. 6, 7 (2012) ("[W]e understand a weapon as a tool that is used, or designed to be used, with the aim of threatening or causing physical, functional, or mental harm to structures, systems or living things.").

Payloads intended to cause destructive digital or physical effects are defense items and thus should fall under the USML, specifically Category XXI, Miscellaneous Articles, which covers:

> (a) Any article not specifically enumerated in the other categories of the U.S. Munitions List which has substantial military applicability and which has been specifically designed . . . or modified for military purposes. The decision on whether any article may be included in this category shall be made by the Director, Office of Defense Trade Controls Policy.[49]

The conventional argument for dual use items, as with exploits above, rests on the presence of both a potential military application and genuine commercial use. This test doesn't cover payloads capable of causing damage or disruption; there is no legitimate commercial application as even penetration testing stops short of causing damage to client systems. The act of accessing a protected system or network is proof positive of a defensive lapse; products of this access, like executing code to create new users, writing into different portions of memory, modifying running programs, are adequate demonstrations of a target system's insecurity. Producing damaging effects is neither necessary nor useful to a testing client. While there are certain minimal exceptions to this logic (commercial software to securely wipe and repurpose computing equipment, for instance), these can be dealt with by individual exception.

Cyber Weapon

Considering each of the three components of malware is important, but there is an additional possibility that comes with the combination of the three where the payload is capable of destructive digital or physical effects. In this fourth potential category, a cyber weapon represents the combination of capabilities to move to a target, gain access, and execute destructive code all in a single package. Such a collection of capabilities, akin to different mechanical components which make up a firearm, represents malicious capability that can be considered a weapon. Thus it is only in combination with a destructive payload that an exploit or propagation method could ever be considered a military application and be listed under the USML. This fourth category, malware, is considered in addition to the previous evaluation of each constituent component.

To demonstrate this logic's application, consider Metasploit, the popular open source penetration testing framework developed by H.D. Moore and now maintained by Boston-based information security company Rapid7. Begun as a collection of tools to help identify and apply relevant exploits to targeted systems for research and pen-testing applications, the Metasploit Framework has evolved to encompass a range of hacking tools including exploit configuration

[49] 22 C.F.R. § 121.1(c) (2014).

modules, payload selection and application features, obfuscation modules to evade intrusion detection systems, and graphical collaboration tools. The range of payloads and exploits used by the program are updated regularly, and its open source nature makes the Framework a widely used and popular tool within the pen-testing community. The flip side to this sophistication and accessibility is Metasploit's popularity amongst some malicious hackers, or black hats,[50] as well as the less talented but more populous "script kiddies."[51] Metasploit gets to the core contradiction of pen-testing and the difficulty of restricting malware through legal means like export controls; the act of such testing, up to a point, is largely indistinguishable from malicious hacking.

Metasploit at its core is a growing collection of exploits and non-destructive payloads complemented by an array of tools and basic programs to combine and apply these components to target systems. Two important facts separate this popular pen-testing tool and malware. First, without being packaged with a propagation method to be deployed on a target machine, Metasploit cannot constitute malware. Second, unless it is used with payloads that can create destructive digital or physical effects, Metasploit does not constitute a cyber weapon and the payloads it contains are not covered under the USML. Neither Metasploit nor its attendant payloads can be considered malware or goods to be controlled under ITAR or EAR. The exploits used in the Framework are considered dual use items and so may be subject to listing under the CCL. It is therefore possible that the larger framework may also need to be considered a controlled item. This determination lies with the Department of Commerce however, which has been reluctant thus far to restrain trade in software and other digital goods.

Conclusion

There is a great deal of difficulty involved in treating the ideas expressed as malicious software like any other tangible good. Bearing this in mind, we would like to close with three points.

First, this is not the Crypto Wars redux. In the 1990s, a battle raged between independent cryptographic researchers and Federal authorities over fears that the inclusion of sophisticated cryptographic algorithms into everyday software would create an irrevocable handicap for the law enforcement and intelligence communities by allowing hostile states and non-state actors to

[50] It is important to note the term "hacker" encompasses anyone who modifies or tinkers with systems, whether hardware or software. It was originally associated with some of the most respected computer engineers and technical entrepreneurs of the 20th century, but modern use has integrated a pejorative inflection. Malicious hackers, or "black hats," are but one subset of a much larger community of developers, tinkerers, and engineers that includes luminaries such as Steve Wozniak and Bill Gates.

[51] Script kiddies are relatively unsophisticated users who make crude use of a more skilled programmer's malicious code. *Know Your Enemy: The Tools and Methodologies of the Script Kiddie*, HONEYNET PROJECT (July 21, 2000), http://old.honeynet.org/papers/enemy/.

keep their communications safe from prying American eyes. The resulting statutory restrictions on the type and sophistication of cryptographic implementations allowed in exportable software generated strong resistance from researchers and the security community who challenged the laws during a period that later became known as the Crypto Wars. One researcher and author of the prominent algorithm PGP, Phil Zimmermann, took the source code for his idea, wrote it into a book, and shipped it abroad in order to place it into the public domain and thereby gain free speech protections for the work.[52]

The prospect of using export control regulations to contain the spread of cyber weapons is not the same as the Federal government's battle against exporting cryptography. Cryptographic algorithms are an embedded product; we find them baked into all manner of software and hardware applications. One of the defining ironies of the Crypto Wars were t-shirts and bumper stickers, made popular at the DEFCON conference, labeled "this [item] is a munition" and followed by code for some piece of an encryption algorithm.[53] Cryptographic implementations were not standalone goods but processes that would become integrated into a large percentage of software applications in the ensuing decade. Malware's exploits and payloads are not an embedded product but purposefully built tools (components), which combine to create some malicious effect. Restricting their exchange does not therefore implicitly target some larger category of goods.

Second, there is some limited use to be gained from the use of export controls to circumscribe the behavior of firms; however, targeting malware and associated tools creates a curious balancing problem between two distinct communities. Open Source software developers and the information security community are accustomed to sharing tools widely as a means of development. Burdensome regulations rooted in a tangible conception of the goods to be restricted are not likely to have much impact, if for no other reason than the goods in question are computer code and thus readily transmissible across the globe. There are few, if any, technically feasible solutions to control the flow of code between borders and likely none from this pool which are politically palatable.

Within the United States, however, the defense contracting industry is tied to Federal requirements through contracting processes. While restricting code's movement across borders here is just as difficult, the reliance of these companies on Federal dollars creates a point of leverage for the application of export regulation to control behavior indirectly. More importantly,

[52] Eric Geller, *The rise of the new Crypto War*, DAILY DOT (July 10, 2015), http://www.dailydot.com/politics/encryption-crypto-war-james-comey-fbi-privacy/. *See generally* PHILIP R. ZIMMERMAN, PGP SOURCE CODE AND INTERNALS (1995).

[53] *See* David Loundy, *Is Your T-Shirt a Lethal Weapon?*, DAVID LOUNDY'S E-LAW WEB PAGE (1996), http://www.loundy.com/Roadside_T-Shirt.html (describing a t-shirt imprinted with three lines to implement RSA in the *perl* programming language).

it is possible that the cyber defense work being done by these firms on behalf of the US defense and intelligence communities creates a compelling government interest to make sure this technology is not sold to suspected or overtly hostile end-users.

Lastly, export control may well not be the best policy mechanism to stymie the development of ever more sophisticated cyber weapons by state actors or to restrain the growth in sophistication and variety of malware developed by individuals and criminal groups. The sheer number of means to exchange software across rooms, campuses, and borders make restrictions on code's movement difficult at best. There is some use, however, in the discussion and debate around how such regulations might be crafted as it raises fundamental questions about the nature of the cyber security environment and its participants while offering the prospect for international coordination through vehicles like the Wassenaar Arrangement.[54]

Key to addressing the use of export controls for malware is the question of what components are being targeted. The existing Wassenaar language refers to controls on "intrusion software" designed for 1) the extraction of data or information, from a computer or network capable device, or the modification of system or user data; or 2) modification of the standard execution path of a program or process in order to allow the execution of externally provided instructions.[55] These clauses map well onto the PrEP framework; the first capturing some functionality of a payload, the second corresponding to exploits.

The rule to harmonize the Export Administration Regulations with Wassenaar has been drafted and circulated for comment by the Department of Commerce but it does not cover the malware components defined under "intrusion software" directly.[56] Rather it targets "'systems,' 'equipment,' 'components,' or 'software,' 'specially designed' for the generation, operation or delivery of, or communication with, 'intrusion software'" and "'technology' for the 'development,' 'production,' or 'use' of equipment or 'software' controlled under the definition of intrusion software.[57] These amount to development tools associated with malware and the propagation methods and vulnerabilities employed in their operation and construction respectively.[58]

[54] "The Wassenaar Arrangement has been established in order to contribute to regional and international security and stability, by promoting transparency and greater responsibility in transfers of conventional arms and dual-use goods and technologies, thus preventing destabilising accumulations. Participating States seek, through their national policies, to ensure that transfers of these items do not contribute to the development or enhancement of military capabilities which undermine these goals, and are not diverted to support such capabilities." *Introduction*, WASSENAAR.ORG, http://www.wassenaar.org/introduction/index.html.

[55] COLLIN ANDERSON, CONSIDERATIONS ON WASSENAAR ARRANGEMENT CONTROL LIST ADDITIONS FOR SURVEILLANCE TECHNOLOGIES 9 (2015).

[56] Wassenaar Arrangement 2013 Plenary Agreements Implementation: Intrusion and Surveillance Items, 80 Fed. Reg. 28853 (proposed May 20, 2015).

[57] *Id.*

[58] *Id.*

This form of regulation is of debatable utility in imposing practical restrictions on the flow of information products and will tend to enable the activities of established companies while penalizing researchers and independent actors. Vulnerability and exploit information, responsibly disclosed, plays a key role in the information security lifecycle and any overt or inadvertent imposition of disincentives on the disclosure process could well contribute to less secure software for the public. These debates, especially over the role and scope of export controls, are ongoing within the national security and regulatory community.

The use of export controls to restrict the development and sale or distribution of software, malicious or otherwise, has attracted a great deal of controversy[59] but could present an opportunity for coordinating policies on the development and use of malware within the United States and abroad. The PrEP framework offers a more complete basis on which to compare and build law around the development and sale of malicious software components and broadening a discussion focused, perhaps to excess, on zero-day vulnerabilities alone.[60] The role and rule of law in a market for software has long been a subject of debate, largely because of the recurrent gaps that exist between technical and policy practitioners. This essay is an attempt to bridge those communities and suggest a technically informed legal framework for policymaking on cyber weapons and malware.

[59] *See, e.g.*, Dennis Fisher, *Governments Need to Discuss Use of Cyber Weapons*, THREATPOST (Feb. 6, 2014), http://threatpost.com/governments-need-to-discuss-use-of-cyber-weapons/104097 (quoting Eugene Kaspersky as saying that "world governments will have to sit down together eventually and hash out the issue of cyber weapons and whether they should be used at all."); *Expert Warns of the Growing Trade in Software Security Exploits*, HARVARD LAW TODAY (Oct. 30, 2012), http://today.law.harvard.edu/expert-warns-of-the-growing-trade-in-software-security-exploits/ (noting the absence of any cyber export controls in United States).

[60] *See* Paul Stockton & Michele Golabek-Goldman, *Curbing the Market for Cyber Weapons*, 32 YALE L. & POL'Y REV. 239 (2013) (discussing the market for zero-day exploits). *See generally* Mailyn Fidler, Anarchy or Regulation: Controlling the Global Trade in Zero-Day Vulnerabilities (May 2014) (unpublished thesis, Stanford University Center for International Security and Cooperation) (on file at http://searchworks.stanford.edu/view/zs241cm7504).