
Josephine Wolff
Rochester Institute of Technology
jcwgpt@rit.edu

Abstract
With computer security spending on the rise, organizations seem to have accepted the notion that buying more—and more expensive—defenses allows them to better protect their computer systems and, accordingly, their business and customers. In the context of complex computer systems, however, defenses can also have the opposite effect, creating new, unforeseen vulnerabilities in the systems they are intended to protect. Advocacy for defense-in-depth and diverse security measures has contributed to creating this “more is better” mentality for defending computer systems which fails to consider the complex interaction of different components in these systems, especially with regard to what impact new security controls may have on the operation and functionality of other, pre-existing defenses. In this paper, we describe and give examples of several categories of perverse effects in defending computer systems and draw on the theory of unintended consequences and the duality of technology to provide some analysis of the origins of these perverse effects as well as some methods for avoiding them and directions for future research.

1. Introduction
Defending computer systems against intrusions and disruptions is a multi-billion dollar industry. With large-scale data breaches increasingly in the news and information technology security spending on the rise, many organizations appear to have bought into the idea that the best way to protect their computer systems, and the associated data and services, is to buy more—and more expensive—technical defense mechanisms. But more is not always better when it comes to defending complex systems. Unexpected interactions between different system components and new defense mechanisms can give rise to unforeseen vulnerabilities in the systems those defenses were intended to protect. Adding defenses to a computer system can actually undermine its security. These perverse effects may arise as the result of the way a single defense technology is implemented (or taken advantage of) by end-users. Such instances are referred to as user-interaction perverse effects. In other cases, measures intended to strengthen security backfire due to interactions between multiple defenses, or technical mechanisms, that in some fashion serve to counteract or otherwise interfere with each other. These instances we call technology-interaction perverse effects.

The potential for perverse effects to arise in the context of defending computer systems against security threats is often overlooked or ignored. In particular, the rhetoric of defense-in-depth and diversity for security has contributed to the dominance of the prevailing “more is better” mentality for defending computer systems. But this mentality vastly oversimplifies the complex environment computer security controls operate in, as well as the potential for those controls to backfire when they interact with other components of the systems they are intended to protect. Avoiding these types of defensive misfires requires careful consideration of the unintended consequences of security controls, especially with regard to what impact they may have on the behavior of both malicious and non-malicious users, as well as what impact they may have on the operation and functionality of other, pre-existing technical defenses.

In this paper, we consider the prevailing wisdom about the need for computer systems to include multiple defenses and many layers of protection and provide examples of how this mindset can fail to account for the complex interaction effects security controls may have both with users and other technical mechanisms. We then extend Orlikowski’s theory of the duality of technology [18] to the relationship between multiple technologies, not just the relationship between users and technology, and apply the existing taxonomy of perverse effects proposed by Merton [15] to the field of computer security to elucidate the different ways these perverse effects in defense can arise. We conclude with some recommendations for how to anticipate and avoid perverse effects in practice, as well as possible directions for future research to reduce perverse interactions of this nature.
2. Defense-in-depth and diversity for security

“We must accept the fact that no barrier is impenetrable,” Verizon’s 2013 Data Breach Investigations Report (DBIR) advises [1]. This idea that no single line of defense can serve as a silver bullet against all manner of attacks and infiltrations has guided the development of several categories of security controls that organizations should attempt to cover, as well as considerable advocacy for layering multiple defenses together to compensate for each other’s shortcomings. In [2], Anderson notes: “The theory behind multi-layered defense is straightforward. If the tools or techniques fail at one layer, the safeguards implemented at the other layers will compensate to prevent system compromise.” If the theory is straightforward, the implementation is anything but, since it’s not at all obvious which tools and techniques most effectively compensate for each other’s failings or how to assess the net impact of a set of aggregated defenses.

2.1. Existing computer defense taxonomies and catalogs

Two of the more detailed and comprehensive attempts to define different layers, or categories, of defense come from the National Institute of Standards and Technology (NIST) Special Report 800-53 on security controls for federal information systems [17] and the Consensus Audit Guidelines (CAG) [4], also called “Twenty Critical Security Controls for Effective Cyber Defense,” a publication of best practice guidelines for computer security compiled by a consortium of more than 100 contributors from US government agencies, commercial forensics experts, and pen testers. NIST 800-53 identifies eighteen "families" of controls, ranging from access control and media protection to personnel security and identification and authentication, while the CAG lists twenty (very different) critical control categories, including data loss prevention, boundary defense, wireless device control, and malware defenses.

The huge variation between the families that NIST 800-53 identifies and the critical control categories listed in the CAG indicates how little consensus there is in this space over even the broad classes of defense that organizations need to ensure they cover, much less the specific mechanisms for doing so. The number of high-level categories listed in both documents (eighteen in NIST 800-53, twenty in the CAG) further suggests just how many different dimensions and layers there are to consider and protect. The 2013 DBIR, for instance, states: “most organizations should implement all 20 of the Critical Security Controls to some level.”

Others offer less specific guidance for crafting a multi-layer, multiple-technology defense. In [19], Parker writes: “We can achieve defensive depth by implementing a sequence of controls that will be serially encountered by a perpetrator who is working toward a specific target. An attacker may give up when he faces one control after another. ... The violation of one control station should immediately alert a security administrator and result in reinforcement of the other controls.” Parker conceives of the layering of these defenses along the dimension of time—as soon as an intruder penetrates one, he comes up against another. This assumes a fairly linear and static progression, however, in which defenders can always be certain they know exactly how a perpetrator will progress through the target system.

Parker also notes that security controls should work “in concert” with other controls so that “failure of one control does not result in a successful attack or loss of the others.” He concludes his discussion of defensive depth with a note of caution: “When you apply this principle, be sure to analyze the role of control objectives in relation to other safeguards and their environments.” The warning is warranted—layering multiple controls may have unforeseen consequences or cause emergent vulnerabilities and, furthermore, we may not know in which order an attacker will encounter these controls—but Parker does not address the essential underlying question of how to analyze different controls in relation to each other and assess their overall impact.

2.2. Defense-in-depth

In [21], Tirenin and Faatz describe a closely related notion of strategic cyber defense, which they call “defense-in-depth,” that is closely related to Parker’s concept of “defensive depth.” Defense-in-depth, by their definition, involves layering multiple defensive techniques “in such a manner that the weaknesses of some can be mitigated by the strengths of others,” a strategy that the authors equate to being able to build trustworthy systems from untrustworthy components. The key to designing these defense-in-depth systems lies in implementing “orthogonal” security controls, that is, controls with independent, or different, vulnerabilities, Tirenin and Faatz argue. “An attack against multiple, independent defenses requires both the expertise and the time to overcome all of them, rather than only one,” they note. “The additional time and effort required may be the key to enabling detection of the attack in-progress, enabling the assessment and response necessary to prevent accomplishment of the attack objective.”
As opposed to Parker’s conception of defense layers as being sequentially encountered by an attacker, Tirenin and Faatz’s definition focuses on the ability of these layers to plug the holes of any individual defense. However, it is not immediately apparent how to either characterize the full set of vulnerabilities associated with a specific control or determine whether that set is fully covered by the other layers. In other words, it’s not easy to identify orthogonal controls or characterize their orthogonality. Other research has advocated in a related vein for holistic defense approaches that combine both detective and reactive defense mechanisms [9], or defenses that operate at the application layer as well as the underlying network layer [3], but provides similarly minimal guidance for how to construct these multiple layers of defense and understand their interaction in a clear or systematic manner.

2.3. Diversity for security

In the absence of any clear guidance regarding how best to combine different types of defenses, a more general—and much easier to implement—principle has emerged around employing multiple different types of controls, or “diversity for security,” essentially in hopes that by covering as many bases as possible, one may rather haphazardly stumble upon some useful defensive depth. Research on combining the signature detection engines of many different antivirus products, for example, shows significant gains in detection capability over using just a single antivirus product [5, 20]. This might seem like a good example of clearly orthogonal defenses—the holes in one antivirus program (i.e., the missing signatures) could be at least partially filled by the signatures provided in the others—but, in fact, the United States Computer Emergency Readiness Team (US-CERT) explicitly advises users not to install more than one anti-virus program, in its Security Tip (ST06-009) on Coordinating Virus and Spyware Defense. “You may feel that the more anti-virus and anti-spyware programs you install on your computer, the safer you will be ... However, by installing multiple programs in an attempt to catch everything, you may introduce problems,” the US-CERT website warns [14].

In [8], describing the results of a set of red team experiments around four scenarios each with increasing layers of added defenses, Kewley and Lowry conclude, similarly, that some additional defenses may actually be counterproductive in preventing intrusions. “The expected result was that as each layer of protection was added, the adversary’s work factor would increase incrementally ... The actual results were quite different,” they write of the experiments. “Interestingly enough, it actually took the adversary less time to capture the flag at configurations two, three, and four than it did for the baseline configuration. As the three layers of security were added, they actually introduced a new control surface for the adversary to exploit ... rather than adding to the assurance of the system, the complex interactions between the security layers were used against the blue team.”

The fallacy of expecting more layers of defense to automatically lead to greater protection may derive in part from physical security analogies that give rise to the notion of defense-in-depth. “Traditional thinking lends itself to the philosophy that the more layers of protection you add, the more secure a target will be. For example, you can lock the doors and windows, put a moat around the castle, put alligators in the moat and a guard dog in the yard, a fence around the moat, and an armed guard in the watchtower to create layers of physical security,” Kewley and Lowry note in [8]. “In the cyber world, however, multiple layers of defense do not necessarily add together to create a higher level of assurance.” Too much defense, in other words, can be just as dangerous as too little.

3. Theoretical framework for perverse effects in defense

There are two broad categories of perverse unintended consequences caused by computer defense mechanisms: those that arise from a single defense mechanism being used in some unexpected or undesirable fashion by end-users, and those caused by an adverse reaction among multiple different defense technologies. In short, a defense may backfire in its intended function to better protect a system due to its interactions either with people or with other defenses. The latter type of perverse effects, in which user interaction with a defense technology serves to negate or reverse the technology’s intended function, aligns with the theory of the duality of technology, which holds that technologies may simultaneously constrain and enable users. Orlikowski explains: “while personal action of human agents using technology has a direct effect (intended and unintended) on local conditions, it also has an indirect effect (often unintended) on the institutional environment in which the agents are situated” [18]. Similarly, a defense technology may constrain users’ security even as it enables a particular form of protection because it can have both a direct, intended effect (e.g., requiring users to change their passwords every 90 days) as well as an unintended effect on an institutional environment (e.g., encouraging people to write down new passwords on post-it notes stuck to their desks).

Orlikowski’s conception of the duality of technology focuses on the interactions between technology and people—specifically the people who...
design and use it. But it can also be extended to consider the interactions between multiple technologies. When a new technical component is introduced to a system it may both constrain and enable the other components. Thus, a new antivirus program may enable detection of a larger assortment of malware even as it flags the malware signatures stored in another antivirus program as malicious and attempts to shut down that other component. A technology, in other words, may similarly exhibit both direct effects on its local technical environment and unintended, indirect effects on its broader systemic environment.

The duality of computer defense technology is central to the potential for perverse effects, but while the original notion of duality centers on technology’s “dual nature as objective reality and as socially constructed product,” understanding the duality of computer defenses requires us to also understand a technology’s nature as a “systemically constructed product,” that is, a technology shaped in part not just by its designers or by its users, but also by the other technologies that surround it. User-interaction perverse effects result from the duality of a defense’s objective reality and social use, or construction. Technology-interaction perverse effects, however, result from the duality of a defense’s individual objective reality and its systemic use, or the ways it reacts to and changes the other technical components of a shared system.

Both types of perverse effects fit the more general definition of unanticipated consequences proposed by Merton in [15] in which “consequences result from the interplay of the action [the addition of a new defense] and the objective situation, the conditions of action [the users and existing defenses].” It is worth distinguishing between the two classes for the purposes of discussing computer security controls, however, because the human conditions and the technical conditions of the environment a new defense is introduced to are subject to different degrees of external control and predictable behavior. Furthermore, separating the scenarios in which multiple technical defenses specifically interfere with each other may contribute to the as-yet murky understanding of how to construct effective multi-layer defense strategies (or, at least, how not to construct very ineffective ones).

3.1. User-interaction perverse effects

The interactions between an individual defense mechanism and end-users can cause perverse effects in a variety of different ways due to both the behavior of non-malicious and malicious users. Consider the set of non-malicious end-users who do not intend to subvert the utility of a newly implemented defense. These users may, nonetheless, negate any potential benefits of a given defense by altering their behavior due to either a disproportionate perception of the increased security afforded them by the new control or the challenges associated with using the new defense mechanism properly.

The psychological effects of added security mechanisms on perceptions of security have not been measured, but in [16] Mitnick cautions that “Security is too often merely an illusion … many information technology (IT) professionals hold to the misconception that they’ve made this companies largely immune to attack because they’ve deployed standard security products—firewalls, intrusion detection systems, or stronger authentication devices such as time-based tokens or biometric smart cards.” His main point is that attackers can often use stolen credentials obtained via social engineering tactics to bypass these security products (hence the illusion). However, the ability to steal credentials does not, in itself, mean that the deployed technical measures have actually made the protected system less secure, merely that they have not made it completely secure.

On the other hand, if organizations or individual users believe so firmly in the protective power of their security controls that they then decide to store more sensitive information on their systems, or forego other defensive measures, or be more careless in protecting their own devices and credentials and following set security protocols, these added layers of defense may backfire. To this end, [16] asserts, “A security code improperly used can be worse than none at all because it gives the illusion of security where it doesn’t really exist.” In other words, a new layer of defense may backfire simply by presenting an illusion of security so compelling as to make non-malicious users lower their guard.

Peoples’ use of security controls is better understood than their perceptions of the protections afforded by these defenses, and this area of computer-human interaction research suggests another possible avenue for perverse effects of these technologies: the challenges associated with user implementation. For instance, [10] finds that there is a correlation between the use of higher-entropy passwords and storage of those passwords (either written down or in a computer file). Thus, a password policy intended to make users’ accounts significantly more secure can backfire by causing more people to write down their difficult to guess and equally difficult to remember passwords on easily accessible post-it notes. The impact of user behavior on defense effectiveness is not limited to passwords; in a study of corporate firewalls, [22] found that nearly 80 percent exhibited “gross mistakes” in their configuration.

Security mechanisms have come under considerable fire for being poorly designed and difficult to use, while the general population has been
rondly criticized as too careless and ignorant to implement security measures properly. The objective in identifying these user perception and implementation issues that can lead to perverse defensive results is not to assign sole blame or responsibility for remedying these problems to either the end-users or the security control designers, but rather to point out the significant unintended impacts that defense technologies can have on user behavior, and vice-versa.

The tendency of non-malicious users to alter their behavior in ways that counteract security controls is an unwitting, if not insignificant, source of perverse results. By contrast, malicious actors may be able to exploit new defensive technologies in much more intentional and direct ways. In [6] Geer notes that several security products, including firewalls, anti-spam software, and intrusion prevention and detection software “have been found to have potentially dangerous flaws that could let hackers gain control of systems, disable computers, or cause other problems.” He describes three categories of vulnerabilities in security products: vulnerabilities that give malicious actors exploitable information about a system, vulnerabilities that allow intruders to enter a system, and vulnerabilities that let successful intruders expand the access they’ve gained to system resources. Reference [6] gives examples of such flaws in products from a variety of vendors, including Internet Security Systems, Zone Labs, Check Point, and Symantec, and cites a statistic from Fred Cohen, managing director of security consulting firm Fred Cohen & Associates, that “security vendors each generally average from three to 12 remotely exploitable critical vulnerabilities per year.” These vulnerabilities are due to rushed production timelines, inadequate debugging, and increasing complexity. Geer argues in [6], noting that as security firms add functionality to their products to “gain a competitive edge and meet the demands of users who assume that more features make their systems safer,” the resulting “feature bloat” may introduce more vulnerabilities since “a combination of functions might cause problems that individual ones would not.”

The increasing complexity of corporate defenses and combination of more security controls may also spur organizations to implement centralized defense management systems that allow for easier monitoring and management of a diverse set of defenses. Such mechanisms may facilitate the jobs of not only internal systems security workers but also external intruders looking for an access point to a protected system. By centralizing the management and configuration of all security controls under a single system, an organization may—in the words of Kewley and Lowry—create “a new control surface for the adversary to exploit” [8] that can actually be used against the defenders. Centralizing control of layers of diverse defenses in some sense negates the security value of that diversity by creating a single, centralized point of failure for all of them.

While software vulnerabilities and centralized defense management systems can backfire by helping intruders figure out how to gain access to the systems they’re intended to protect, other forms of defense may provide additional information that attackers can exploit to their advantage. In some cases this information may simply be the existence of new layers of defense, indicating to the intruder that there is something worth protecting, and therefore worth stealing. This may not be relevant for intruders motivated by financial gain seeking to expend as few resources as possible in their pursuit of profitable information, since they may simply look for less well-protected data. On the other hand, well-funded espionage organizations may have precisely the opposite reaction to encountering strong defenses since they may have significant resources and their interest is in identifying and accessing the most useful and relevant intelligence information, rather than making money.

Finally, malicious actors may be able to take advantage of additional information provided by security mechanisms to perpetrate a completely different type of attack than the defense was intended to prevent. For instance, the Domain Name System Security Extensions (DNSSEC) implemented to prevent the use of forged or manipulated DNS data required the addition of a new type of DNS record, RRSIG records, which contain the digital signatures that can be used to verify that the provided DNS data has not been forged. These digital signatures were intended to protect users from cache poisoning attacks, but they have also raised concerns in its potential to worsen a different type of attack—DNS amplification attacks. By spoofing the source address of DNS queries, an attacker can direct the response to that query to a target server and flood it with the amplified response of the DNS records. While these amplification attacks existed before the implementation DNSSEC, the addition of the RRSIG records can increase the amplification factor even more, leading to higher volume attacks. It’s uncertain to what extent DNSSEC actually exacerbates the problem of amplification attacks, though estimates of how much DNSSEC increases the amplification effect range from a factor of two [7] to seven [11].

### 3.2. Technology-interaction perverse effects

Identifying sources of perverse unintended consequences of individual defense mechanisms may provide some insight into how to avoid these issues in the future, but it offers little guidance for how to—or
as the case may be, how not to—combine different defenses for an effective defense-in-depth configuration. To that end, it is instructive to look at the few known ways in which multiple defenses can actually counteract each other and layers of defense serve to lessen the cumulative protection. One category of these negative reactions concerns defenses tagging other defenses as malicious and trying to disable them. This problem goes back to the US-CERT website’s warning not to install multiple antivirus programs on a single system, which notes that “in the process of scanning for viruses and spyware, anti-virus or anti-spyware software may misinterpret the virus definitions of other programs. Instead of recognizing them as definitions, the software may interpret the definitions as actual malicious code. Not only could this result in false positives for the presence of viruses or spyware, but the anti-virus or anti-spyware software may actually quarantine or delete the other software” [14].

Anti-virus programs’ propensity to attack each other could stem from several factors, including the possibility that the manufacturers of these products are trying to squash their competition by removing competing products from machines, as well as US-CERT’s assertion that the signature detection engines may trigger each other. More generally, many of the capabilities and characteristics of anti-virus and anti-malware software are also suggestive of malware—for instance, their ability to scan and delete other programs and the fact that they cannot be easily switched off—so it is not altogether surprising that they might be regarded with suspicion by other anti-malware programs.

Another way in which defense mechanisms may counteract each other’s effectiveness is by interfering with the ability of the other to function properly. For instance, an intrusion detection or prevention system responsible for monitoring a system’s traffic for anomalies or intruders may be effectively foiled by using a strong encryption scheme on internal traffic. This is related to Kewley and Lowry’s finding in [8] that IPsec actually degrades the utility of firewalls in their red team experiments. They note, “The inclusion of IPsec in this case provided an avenue for the adversary to exploit and thus get through the boundary firewall,” but add that this result may vary depending on the system being protected and “in other network configurations, the use of IPsec would in fact add a layer of security.” While both encryption and intrusion detection can serve a valuable defensive purpose individually, in combination they may actually aid intruders, who can take advantage of the encryption to hide from an intrusion detection or prevention mechanism.

Concerns about the potential of defenses to backfire extend all the way to the policy world, where President Obama signed into law P.L. 113-6, the Consolidated and Further Continuing Appropriations Act for Fiscal Year 2013 on March 26, 2013. Section 516 of the Act included restrictions banning the Departments of Commerce and Justice, the National Aeronautics and Space Administration, and the National Science Foundation from purchasing any computer equipment from China unless “the head of the entity, in consultation with the Federal Bureau of Investigation or other appropriate Federal entity” had made a risk assessment of potential “cyber-espionage or sabotage ... associated with such system being produced, manufactured or assembled by one or more entities that are owned, directed or subsidized by the People’s Republic of China.”

The provision was intended to protect U.S. infrastructure from espionage attempts and infiltration by Chinese spies, but in a letter to Congress, several technology industry associations referred to these requirements as setting a “a troubling and counterproductive precedent” which could “impede the U.S. government’s ability to protect itself through use of the latest cutting-edge IT products.” Congress’ actions may have been dictated by the best of intentions, but their results could be disastrous not just for American technology firms, but also for the security of U.S. computer networks, according to the April 2013 letter [12].

4. Closing doors without opening more

There are some general lessons to be drawn out of the cases and categories of defenses backfiring and counteracting each other—many of which have already been identified and addressed. Concerns about non-malicious end-users unintentionally degrading the effectiveness of security controls hint at the necessity of better designed, more user-friendly defenses, as well as better end-user education about the actual degree of security afforded by network protections and training in their proper implementation. And perhaps the best—and least easily taught—protection against the illusion of security is simply a deeply engrained and unwavering paranoid mentality. The interaction between defenses and malicious users provides some more counter-intuitive and probably less realistic possibilities, including manufacturing less complex, more thoroughly debugged defenses along a slower production timeline, or maintaining a decentralized security posture of several independently operated and managed controls.

The potential of especially strong defenses to actually attract intruders in some cases indicates the importance of sometimes keeping security mechanisms as secret possible—as opposed to the strategy of broadcasting them loudly in hopes of deterring attackers, which may be better suited to organizations worried about financial crime whose
attackers are likely to be more resource-conscious. In [2], Anderson makes a similar recommendation, writing that “If you can convince your enemy that the system/information that is valuable has no value whatsoever, then there is no need to hide it.” Lowry & Maughan, in [13], also note that that “the most likely indicator of attack is detailed information gathering. Protection of this information will likely prevent an attack or, in the case of a determined adversary, cause him to engage in riskier (more likely to be observed) behavior.” Dealing with defenses that either tag each other as malicious entities or interfere with each other’s ability to perform their designated functions requires choosing between the functionality of the conflicting controls based on the requirements of the system being defended—or accepting the downgraded functionality of both, if it seems superior to either individually.

4.1. Sources of unanticipated consequences

In [15], Merton identifies five sources of unanticipated consequences resulting from purposive social action: ignorance, error, fulfillment of short-term interests at the expense of long-term interests, basic values, and self-defeating prophecies. Ignorance, or uncertainty, is defined by Merton to include only those circumstances in which “the interplay of forces and circumstances ... are so complex and numerous that prediction of them is quite beyond our reach.” In other words, cases where the unintended consequences of an action could not reasonably have been predicted based on existing knowledge.

Extending Merton’s sources of unanticipated consequences to the perverse effects of computer defenses, the domain of ignorance seems to map most closely to the ways in which certain controls influence non-malicious user behavior. While there has been research done on the obstacles to implementation and usability problems of some security mechanisms, it remains an active area of study, indicating how much there is left to learn about the ways in which people do and don’t perceive, value, and implement security measures for computer systems. Predicting the ways in which end-users will adopt, adapt, implement, and circumvent defense mechanisms is still, to a great extent, beyond our reach.

Merton’s second category of unanticipated consequences stem from errors in which actors fail to “recognize that procedures which have been successful in certain circumstances need not be so under any and all conditions.” This classification applies to the two categories in which multiple defenses degrade a system’s overall security, either by fighting each other as perceived malicious entities or inhibiting the operation of the other. In such cases, the defenders believe incorrectly that procedures like anti-malware software and IPsec that have been successful in certain circumstances will still be so in the presence of other types of defenses, failing to recognize the interactive effects.

The third factor Merton identifies as causing unintended consequences is “imperious immediacy of interest” in which an actor’s “paramount concern with the unforeseen immediate consequences excludes the consideration of further or other consequences of the same act.” This may apply to the implementation of centralized defense management systems—intended to satisfy an organization’s short-term need to simplify its security operations at the expense of its long-term interest in deflecting intruders—as well as the vulnerabilities in poorly tested and developed security products that are brought to market too quickly, and with too much complexity, to satisfy manufacturers’ short-term sales goals at the expense of their long-term interest in gaining a reputation for providing effective defenses. More generally, when the immediate interest of a defender is simply adding to their assembled defenses—on the assumption that more defense will better protect their system, or at the very least, better shield them from being held liable for any breaches—this may eclipse a more considered analysis of what defense would be most useful and least inhibitory in the context of the existing defensive ecosystem. In this sense, any perverse effects resulting from defensive decisions spurred by a “more is better” ethos may fall into this category on some level.

Basic values can lead to unintended consequences in “instances where there is no consideration of further consequences because of the felt necessity of certain action enjoined by certain fundamental values,” Merton asserts. As an example of this phenomenon he cites Weber’s analysis of the Protestant work ethic and capitalist spirit in which “active asceticism paradoxically leads to its own decline through the accumulation of wealth and possessions entailed by decreased consumption and intense productive activity.” While the notion of basic values maps less neatly onto the categories of computer defense, it does relate in some sense to the risks associated with attracting intruders by layering on especially strong defenses. In such cases, the “felt necessity” of defending especially sensitive information as much as possible may override concerns about whether these measures may have further consequences, such as attracting the attention of spies.

The notion of self-defeating prophecies, in which “public predictions of future social developments are ... not sustained precisely because the prediction has become a new element in the concrete situation, thus tending to change the initial course of developments,” do not have a clear analogy to computer defense mechanisms. However, the space where there is
perhaps some potential for such an effect is in the policy realm. Public predictions like those voiced by the technology industry associations that regulating technology imports from China may weaken the United States’ security posture could conceivably cause other, domestic security firms to address the gap by manufacturing new defenses replicating the capabilities of those coming from Chinese companies. This change in the “initial course of developments” could be further encouraged by the promise of not having to compete directly with Chinese manufacturers, whose ability to sell products within the U.S. would be severely compromised by the new law.

4.2. Anticipating and avoiding perverse effects in defense

While Merton’s five sources of unintended consequences of purposive social action provide an interesting lens through which to view the causes of perverse, unintended effects of computer defense mechanisms, they do not offer much guidance for how to avoid these effects. Certain considerations and repeated themes do emerge from these cases, however, suggesting some questions that should be asked of a new defense prior to implementation. These include:

- What responsibilities will this defense impose on non-malicious end-users? How might users react to those responsibilities?
- What new information (labels, signatures, etc.), if any, is being added to the system as part of the defense? How might that information be subverted or used by malicious users?
- Could the protections being extended to non-malicious users and networks also help protect malicious intruders and their activities?
- What are the underlying assumptions necessary for the operation of any existing defenses in the system? How might a new defense alter those assumptions?
- What new attack surfaces or access points (including trusted update clients, centralized management interfaces, programs with the ability to scan and alter files, programs that are relied on to monitor and alert users to any anomalous behavior, authentication credentials, etc.) are created by this defense?

The central lesson of the perverse effects in computer system defense that have been witnessed thus far seems to be that we still understand very little of the dual nature of defensive technology both in relation to users and in relation to the larger technical systems in which it operates. This suggests both a disconnect in the impacts that the designers of technology intend their creations to have and the ways that those creations actually do impact the social environments where they are used as well as a disconnect in the intentions of the designers of different defensive technologies. In some cases, the designers of a defensive technology fail to predict how characteristics of an institutional environment and user community will constrain or facilitate their implementation; in others, those same designers fail to appreciate the slightly different intentions and impacts of the designers of other defensive technologies—and how those impacts may be at odds with the intentions of their own design, even though both may have in mind the same broader goal of greater security. Both the failure to appreciate how the design of a defense may be at odds with its users’ goals and the failure to appreciate how the design of a defense may be at odds with the goals of the designers of other defenses operating in the same system are potential sources of perverse effects.

4.3. Directions for future research

This paper proposes a framework drawn from Merton’s sources of unintended consequences and Orlikowski’s duality of technology for understanding how perverse effects arise in computer defense. It leaves open many questions about what types of defenses cause these effects in practice and why, and how best to avoid or counteract them. Future research in this area could serve to further our understanding of when and why perverse effects arise in defending computer systems and how they may be most effectively mitigated. For instance, further research in the area of computer-human interaction on the behavioral impacts of defensive technologies such as password policies, email filters, and browser warnings could contribute to a clearer understanding of user-interaction perverse effects by shedding light on how defenses may unintentionally shift user behavior to weaken security. Psychological studies on the perceived impact of such measures could also further our understanding of whether such behavioral shifts are due to a false sense of security—and to what extent that false sense exists—or rather to implementation and usability challenges posed by the defense’s design. This distinction is crucial for trying to address these behavioral influences: designing defenses that are more usable may be effective in the latter case, but might do little to mitigate users’ false sense of security and subsequent carelessness.

In the area of technology-interaction perverse effects, more research is needed to measure the impact of these effects and the extent to which they are due to intentional competitive agendas on the part of security vendors (e.g., one antivirus firm’s attempts to supplant the product of a competitor) or misaligned security objectives (e.g., the objective of
confidentiality—supported by encryption—and the objective of threat detection, hindered by that same encryption of traffic). This distinction again has crucial implications for trying to address perverse effects: one explanation calls for potential coordination of competing firms, another requires a prioritization of different, competing security goals on the part of defenders. Research in this area might include assessing the actual impacts of adding additional antivirus programs, encryption packages, firewalls, or other defensive technologies to computer systems and testing them in different combinations to see which configurations yield the best results.

As computer systems and their defenses become increasingly complex, these questions about how and why perverse effects arise become increasingly difficult and important to answer. “Complexity is the bane of network security,” Kewley and Lowry assert in [8], adding, “While components and systems become more complex, they become more difficult to secure. To secure something, you must thoroughly understand how it behaves, where the vulnerable holes are, and how to plug them. As operating systems and applications add features to make themselves more ‘user friendly’ and compatible with various devices, they become infinitely complex. When these infinitely complex components begin communicating with other components in complicated ways, securing them becomes a significant challenge. … Individual components in themselves can be for the most part trustworthy, however when multiple components are added to a networked system, the interactions become complex and … new vulnerabilities are often introduced.”

Simply dodging, or minimizing, those new vulnerabilities—no easy feat in itself—does not answer the question of how best to layer and combine different defenses but it may, at the very least, help answer the less daunting question of how to avoid the worst modes of defense-in-depth, and how to move beyond a simplistic model of computer security in which our governing principle is that more security is always better.

5. References