Pools, Clubs and Security: Designing for a Party Not a Person

Zheng Dong, Vaibhav Garg, L. Jean Camp, Apu Kapadia
School of Informatics and Computing
Indiana University
{zhdong, gargv, ljcamp, kapadia}@indiana.edu

ABSTRACT

Security solutions are often developed for individual end-users, typically identified as the weakest link. However, the inability to differentiate *good* security from snake oil and the misalignment of the cost of security investment vs. the liability of security breaches justify the rational imperative to free-ride. Thus, security solutions fail not because of technological or usability limitations, but due to economic constraints and lack of adoption. Existing research conceptualizes security as a public good suffering from underinvestment, or a private good with *externalities*, i.e. unintended consequences. We argue for a new paradigm of security solutions designed for communities rather than individuals. We leverage canonical economic theory of 'club goods' and 'common-pool resources' to encourage security through collective action and peer production. We operationalize these by providing concrete instantiations of security solutions. Investigating the paradigm of cooperation through community informs novel solutions that impinge on real world security and we advocate further research to enable this shift.

Categories and Subject Descriptors

K.6.0 [**General**]: Economics; K.6.5 [**Security and Protection**]: Invasive software

General Terms

Security, economics

Keywords

Peer production, social networks, trust, computer security

1. INTRODUCTION

Current security solutions target individual end-users, often identified as the weakest link [8, 54, 59]. Yet individuals do not capture much of their investment in security, both because of the externalities that impinge on others [12] as well as the public-good nature of some investments [11]. For example, the prevention of a DDoS attack on the Department of Defense wouldn't have directly affected the bystanders who unwillingly participated in Anonymous's Low Orbit Ion Canon (LOIC) botnet. Thus, the economics of altruistic investment, without cooperation, would lead to underinvestment, i.e. without sufficient incentives, the investment in security is suboptimal. In this paper we argue for a paradigm shift by proposing solutions for communities and encouraging security through collective action and *peer production*, i.e. a collaborative approach to creating goods and services.

Individually rational decisions often lead to suboptimal group outcomes [52], e.g. overpopulation and global warming [25]. Security solutions targeting individual end-users are similarly limited [3]. For example, the misalignment of cost and liability implies that while the cost of security investment is borne by individuals, the liability of security incidences, such as DDoS, is experienced by others [3]. The proposed solutions include legislative measures, such as graduate response [50], which, though well meaning, may be counterproductive not just due to the potential for abuse [41] but also due to the lack of security education [22]. Due to the information asymmetry between security software vendors and consumers, the end users may not be able to distinguish systems that are more secure from those that provide only basic protection, thereby leading to a *market for lemons* [1] as we see in used-car markets. Even if individuals are capable of knowing the quality of security software with help from a clear indicator, individuals are still willing 'to ride freely' [59]. Therefore, individual incentives to purchase security from a firm are not effective, as in many cases this requires pure altruism and in other cases it requires overinvestment (e.g. above optimal) in a public good.

Without adequate incentives security solutions are subject to the 'free-rider problem'. When the reliability of a system is dependent on the best effort of a group, only the individuals with the highest cost-benefit ratio contributes, while others are rationally inclined to withhold investment [59]. However, information systems are inherently value laden [30] and can be restructured [36]. Thus, security solutions can be engineered as clubs preventing the potential for hidden action [3]. The individual disincentive to invest and 'free ride' [59] is overcome by the peer production of a 'security club' or 'common secure computational resources' [16], e.g. secure common bandwidth.

Thus, we argue for a paradigm shift, focusing the design of security solutions for communities and not just individu-

NSPW' 12, September 18–21, 2012, Bertinoro, Italy.

als. The success of peer production through collective action is well documented in other domains. A classic example in software is that of the Linux kernel [6]. Similarly, in security Camp [9] developed the Net Trust system, where individual website blacklists could be uploaded and shared among groups of friends without additional sharing of histories or comments.

From an economic perspective our innovation lies in considering security as a *club good* or *common-pool resource*. Both of these economic notions suggest that security could be implemented in a community setting, but carry a slight difference; as a *common-pool resource*, every individual could participate as long as the resource constraint is satisfied, but as a *club good* people are invited to join a community and may be excluded from participation. We also introduce two motivating examples for this innovation: *cooperative subversion detection* and *community patching*.

The paper is organized as follows: Section 2 introduces the economic paradigm we leverage such that security investment could approach the optimal level; Section 3 elaborates the economic background of implementing security as a club good; Section 4 discusses an instantiation for security as a *club good*; Section 5 illustrates an alternative situation when security can be considered as a *common-pool resource*; Section 6 provides an example of *common-pool resource* implementation in security; Section 7 elaborates additional peer-based implementations; and Section 8 summarizes our contributions and concludes the paper. We are proposing the design of security as a club good and show how this might be done to encourage communities to invest in patching and engage in cooperative version detection.

2. ECONOMIC PARADIGMS

In this section, we introduce basic economic theory to better explain this new security paradigm. We argue that the reconceptualization of security as a *club good* or *common-pool resource* would improve individual investment in security. We build upon work by Ostrom [51] which illustrates that were security inherently a public good, then the proposed design paradigm would remain an efficacious choice. After the theoretical foundation we provide two illustrative instantiations as high-level proofs of concept.

To explain our argument three fundamental properties of a classic good must be (briefly) introduced. The descriptions in this paragraph are highly simplified; first person is used to make these descriptions suitable for those who need only a cursory idea of the concepts to understand the proposal. Recommendations for classic readings are [15, 60].

Understanding the proposed paradigmatic shift requires understanding the nature of private goods as opposed to public goods and as further refined into *club goods* and *common-pool resources*. The subsequent concepts are critical to that understanding. Goods that will be produced at a socially optimal equilibrium have three characteristics: *exclusion*, *rivalry*, and *transparency*.

Exclusion means that I can keep you from using a good. Property rights are grounded in this right to exclude. Essentially, *rivalry* means that if I am using a good, you cannot. *Rivalry* may arise from consumption, e.g. eating a cookie or from non-divisibility, e.g. the inability to share pants simultaneously. Rivalrous consumption implies *exclusion* post-consumption, but *exclusion* is distinct. *Exclusion* applies when I can prevent you from using a good *regardless of whether I am consuming it or not*. *Transparency* requires that the nature and quality of the good are readily apparent. Effectively this may apply if the search costs are low. This may arise when the nature of the product is either inherently obvious, e.g., some chocolate fudge with neither nuts nor peanut butter. Alternatively, transparency arises when the product is simple and variations are easy to evaluate, e.g. it is simple to compare term life insurance policies on the Internet [29].

3. SECURITY AS A CLUB GOOD

Club goods are partially non-rival and are or can be made excludable. The classic example of a club good is that of a fenced-in swimming pool where members of the club pay for use and others are excluded. Such a facility is partially non-rival because my use does not exclude yours; in fact, use by one person alone may not be optimal for that (potentially lonely) person. However, there is the potential for crowding (resulting in decreased or absolute limits on the number of participants). Club goods are not associated with the underinvestment that characterizes public goods.

Determining the correct size of a club is a necessary component for the maximization of social welfare. Based on demand size and the nature of the good, the number of clubs differs. In theory, individuals will depart from and join clubs to reach that optimal point. In practice, iterative evaluation of this optimal point can be integrated into any implementation.

To implement security as a club good it must be the case that others can be prevented from accessing some significant components of the value of the investment in security. The club good need not capture all value or be perfectly non-rivalrous. It is possible to have a club good and improve investment with associated positive externalities. For example, ready availability of Boys and Girls Clubs can lower the rate of nuisance crimes for an entire neighborhood, as well as provide direct value to participants. While others are excluded from the direct educational value of the Boys & Girls Club, the entire neighborhood benefits from the secondary characteristics.

Recall that club goods are not associated with the underinvestment characterized of public goods. This implies that taking the components of security, which have the characteristics of a public good and creating a club good will result in significant increases in security investment. Creating a club good requires creating a design such that security has both rivalrous and excludable components and making the value of these components visible.

In terms of security itself the first point is that there already exist excludable elements in the presence of shared bandwidth and this simply needs to be made visible or transparent. In a shared last-mile scenario, the loss of bandwidth is local and inherently excludable. Thus, a machine that is engaging in spam or DDoS has a cost, and investment in ending this cost will result in benefit to those sharing the bandwidth.

The second excludable event is the protection of one's own machine. While malware can be used to spam others and implement DDoS attacks, it also attacks the host. As malware is known to install keyloggers and other mechanisms for password theft, the value for the individual investing in security is clearly greater than zero. This is illustrated in

the wild by the fact that, while there is underinvestment, there is certainly investment.

It is intuitively reasonable to exclude someone from a constructed social network, and it is widely available as a basic functionality in most online social networking websites. The current Facebook design allows a user to 'unsubscribe' from a friend's future activities (remove them from *news feed*), 'unfriend' a person, or completely 'block' a person. The actions can be taken easily by clicking on a button on the webpage and are reversible. Any user may report a malicious Facebook account; a suspicious account will be permanently removed if the report is confirmed. Similar mechanisms exist in Twitter and Google Plus. Considering that an individual may be part of several social networks (e.g. work, home, hiking), Google Plus has implemented the *circles* feature, which guarantees that content is delivered to the appropriate audience. Since certain content has already been made invisible to some subscribers, the implementation of *circles* could be considered as a form of effective user exclusion.

By using social networks we can create an excludable zone whereby security reputations are made visible. If it is possible to create reputations based on security or lack of security, exclusion is achieved as reputation systems are inherently excludable; you cannot use my reputation without my consent. (Of course, one could be duplicitous, yet exclusion does not require perfect enforcement; that burglary exists does not mean private property is not excludable.)

The third point is that the subversion of a participant in your social network can be particularly harmful to members of that immediate network because criminals regularly make use of social networks. As such, protecting yourself also protects your friends. While security has not been excludable at the individual level, the value of being part of a social network has certainly been leveraged by criminals. While the benefits do not accrue to the social network; the harm does. Specifically, the risks of masquerade attacks are compounded when there is a social connection to the individual who has suffered a machine subversion. The practice of leveraging social connections for diffusion and infection may have been introduced by the ILoveYou virus. In any case, ILoveYou is a classic example. The efficacy of leveraging social connections to increase the vulnerability of the target of a phishing attack was illustrated by Jakobsson et al. [31], and the use of 'Please send money' scams is a testament to the power of social ties to extract value from a target fraudulently. The benefits within the club are exactly equal to the value of social network ties in attacks that are mitigated or prevented.

Considering security as a club good does not necessarily mean that an individual needs to explicitly pay for participation. Ideally, individual contributions in a small community could serve as a primary source of security enforcement; external advisory sources of security (e.g. dialog notifications of anti-malware software, security warnings of browsers) would be utilized to inform reputations, but only as a complement to individual contributions.

In summary, security has components of a public good. Other domains have illustrated that public goods can be made into club goods by changes in governance. Building on this, we argue that categories and components of security threats can be made club goods by changes in the design of these systems. Recall that the advantage of redesign to change a public good into a club good is that the individual has more incentive to invest in a club good than in a public good.

4. COOPERATIVE SUBVERSION DETECTION

As an instantiation of *security as a club good*, we propose an innovative mechanism in which a small-scale network is constructed with machines (including computers, smartphones, tablets, etc.) owned by members of a community. Each machine in the network constantly tests for potential subversion on other machines and assists with malware removal when an infected machine is discovered. Note that only members in the community could participate in this mechanism, and it requires an explicit invitation mechanism for a new individual to join the community (i.e., majority vote or blackballing).

Detection is based on the assumption that individuals are self-similar and mainly relies on a comparison to observed historical patterns. For example, imagine the primary Internet use of Alice's phone is checking emails and the daily data usage is approximately 500 KB. Other machines in the network may identify it as an irregular observation when Alice's phone suddenly reports a daily data usage of 20 MB.

The innovation of this scenario lies in peer production in detecting subverted machines. Each machine plays the role of *claimant* and proves to other machines that it has not been subverted. Each machine also plays the role of *verifier* in another round to decide if a claimant's statement is valid. For example, Alice, Bob, and Carol each have a machine in the network. As the detection process starts, Alice's machine (claimant) first proves to Bob and Carol's machines (verifiers) that it has not been subverted. Regardless of the result, Bob's machine (claimant) then proves to Alice and Carol's machines (verifiers) that it has not been subverted in the next round. Note that being a claimant in one round does not exclude a machine from being a verifier in another round.

Peer production is distinct from crowdsourcing in that crowdsourcing is organized by a firm for the ends of the firm. Crowdsourcing implies obtaining information and feedback from customers or potential customers. Peer production is self-organized. While firms may profit from the results of peer production (e.g., RedHat and Linux) there is not a firm to organize nor exclude others from the value resulting from the efforts of the crowd.

Regarding the detailed design of this scenario, we divide the entire process into five phases: introduction, rejoining, attestation, verification, and recovery. Table 1 summarizes the tasks and purposes of each phase. A central server manages the introduction and rejoining phases while the other three phases may be conducted in a distributed manner. A device list is maintained on each machine. New entries will be added when an introduction or rejoining phase has completed. To delete an entry a departing device can actively report to its peers, or it can be removed passively after it has been inaccessible for more than one round until rejoining.

We also track unusual device departures and absences from the network to account for a situation in which a subverted machine physically presents but cannot send attestation messages on the pre-set schedule and ensure that no participating machine presents but does not respond to the required attestation process.

Table 1: Phases of Detailed Design

Phase Name	Tasks	Notes
Introduction	Proximity authentication Communication key assignment	For newly joined machines
Rejoining	Historical challenge Communication key assignment	For returned machines
Attestation	Runtime info collection Message broadcasting	For claimant machines
Verification	Previous status comparison Subverted machine identification	For verifier machines
Recovery	Recovery package search, delivery and execution	For machines identified as subverted

The introduction phase begins when a machine enters the constructed network for the first time. Normally, this happens when a new member joins the community or an existing member purchases a new device. In the introduction phase each device first passes a *proximity authentication*: the new device is given a challenge that only the device owner could solve. Applicable *proximity authentication* mechanisms include *Seeing-is-believing* [40], *Amigo* [61], and *Emsemble* [33]. This process ensures that a machine is controlled by its owner instead of a remote party. A communication key is assigned to the newly joined machine at the end of introduction phase.

A rejoining phase is needed as some machines may leave the network and return later. This phase starts with a historical challenge, which asks questions about previous activities of the machine. Historical challenges could be implemented as password-based [37], certificate-based [57], or biometric-based [62]. Upon successfully passing the historical challenge, a new communication key is assigned to the returned device for attestation and verification phases.

In the attestation phase each machine collects its current runtime information, such as active processes, active ports, and inbound/outbound traffic. This information is incorporated into an attestation message. Encrypted attestation messages are then sent to other machines on the device list along with a digital signature. This process repeats on a pre-set schedule.

Upon receipt of an attestation message, a machine checks the digital signature of the message and decrypts the message once the sender's identity has been verified. The verification process starts with a comparison of the claimant's current status and its previous attestation. Significant variations are identified based on a local tolerance setting. Verification resulting from this phase is generated and communicated with other verifiers. Once a majority of verifiers determines that a claimant has been subverted, a recovery phase is entered. Several collaborative rating algorithms may be utilized to calculate the verification result [27, 49, 24].

Considering the storage and power limits of mobile devices, the attestation messages of each device are only stored on a few neighboring machines. For example, Bob and Dave's machines keep a copy of today's report from Alice's machine, while Carol deletes Alice's report immediately after the verification phase for Alice. Similarly, Bob's historical attestations may only be stored on Alice and Carol's machines.

Malware removal tools are stored on different machines (in most cases on computers rather than mobile devices). Once a subverted machine is detected, recovery packages are first searched from verifiers' machines. Through the network, available malware removal tools are then delivered to the subverted machine with the protection of encryption and digital signatures. After the execution of the recovery package, a machine needs to re-enter the network through the introduction phase. Alternatively, the machine could rejoin the communication through human interactions.

We argue that these processes could be executed automatically by an application installed on each machine, while the execution of the designed processes could be protected and enforced by software-based attestation mechanisms [32]. It is true that these could be overcome by rootkits that personalize and customize traffic based on the machine subverted. Such customization and personalization would increase the cost of an attack. It would also change the nature of botnets from roughly uniform machines, which can be easily controlled, marketed, and utilized to a network at once more difficult to describe, market, and manage. These would be significant changes in the cost of ecrime.

Security as a club good has clearly been incorporated into this implementation. *Exclusion* was achieved by providing subversion detection services to community members only. Inside the community, however, each machine conducts and receives subversion detection without affecting other participants. That is, this mechanism is *non-rivalrous*. We discussed an example of implementation in a previous research paper [18].

5. SECURITY AS A COMMON-POOL RESOURCE

Common-pool resources are non-excludable but rivalrous [48]. The canonical example is fisheries. Alice cannot prevent Bob from fishing in the same river. Thus, fisheries are non-excludable. However, a fish caught by Alice cannot be caught by Bob, hence fisheries are rivalrous. Hardin's canonical paper, entitled "Tragedy of the Commons" [25], argues that without public or private intervention,
common-pool resources are not sustainable. Hardin uses the example of herding pastures. It would be rational, he argues, for every herder to have the maximum number of cattle they can afford graze on the pasture. While the positive utility of adding another animal to the pasture commons positively affects the individual herder, all share in the negative externality of overgrazing.

This argument is a generalization of the prisoner's dilemma [52]. Consider a scenario in which two criminal accomplices

are imprisoned in separate rooms with no means to communicate. If either of the prisoners defects, then they get lower sentences. If both prisoners defect, neither one gets the benefit. If, however, both cooperate they still would get lower sentences. Here the individual rational strategy is to defect, leading to a suboptimal Nash equilibrium [35], where neither of the prisoners enjoy shorter sentences. However, the optimal strategy would be for the prisoners to cooperate.

Hardin argues that similar to the prisoners the herders would adopt individually rational strategies, leading to overgrazing and thus destruction of the commons. He argues that in order to ensure that the commons is sustained, there needs to be public or private intervention. Though Hardin uses the example of local pastures, his argument is targeted at global issues such as overpopulation. Given the actors involved in the dialogue, such issues can be contentious even when the actors agree to communicate, as we observe with the various discussions on global warming.

Regarding security, Herley et al. [26] argue that phishing suffers from the tragedy of the commons. They argue that phishers have a limited number of phishable dollars (or common-pool resources) that they can exploit. However, more and more phishers contest for this 'shared' resource. Thus, ever increasing competition means that phishing does not provide easy money, but is rather *a low effort and low reward* endeavor.

Unfortunately, Hardin's argument, meant to examine overpopulation, was taken at face value and applied to areas such as fishing [21] and forestry [34] where public or private appropriation of these resources often has led to suboptimal results [21]. Even when the intervention was well intended, public/private bodies did not have the granularity and depth of knowledge possessed by local stakeholders, leading to issues such as monoculture and even outright destruction [7, 4]. These interventions were particularly unfortunate, as many of these resources had traditional local institutions that had evolved over decades, if not centuries, whose goal was to ensure sustainability through cooperation and detecting/preventing defection.

Elinor Ostrom studied such institutions to argue for nonmarket and non-state solutions for the tragedy of the commons [46, 47]. The classic problem in a public commons is that each individual has the incentive to use as much of the common good as possible, yet over-utilization destroys the value of the whole resource. Ostrom argues that a range of social and cultural structures can be created to manage commons absent a market, and argues for the characteristics where these solutions are applicable. Similar but more limited observations have been made in the design of practical reputation systems.

Ostrom identifies a five-dimensional framework to facilitate the governance of the commons through local and immediate stakeholders, rather than through external intervention [17]: 1) the possibility of even temporary exclusion, 2) moderate rates of change in the social network, 3) ability to monitor resources, 4) existence of reputation within the community, and 5) the existence of social norms.

We argue that with the construction of a patching community, the security instantiation introduced in the following section addresses three of Ostrom's five conditions: 1) moderate rates of change in the social network, 2) reputations, and 3) the monitoring resources . Unlike *club goods* which were discussed in Section 3, we do not emphasize the notion of a *social network*, and *exclusion* is not a required component to the *common-pool resource* design. That is, any individual could participate as long as the resource constraint is satisfied, even if he/she does not know most of the community members.

A similar approach has previously been applied to information resources to provide alternative solutions to intellectual property issues [28]. Hess et al. [28] distinguish between ideas, the representation of ideas as *artifacts*, and the availability of artifacts through *facilities*. They note the increasing frequency of 'intellectual land-grab' by private entities and how it is being countered by initiatives such as Creative Commons, morphing the scholar's role from that of a passive appropriator to an active provider. For example, public bodies such as the NSF now expect a research dissemination plan to complement grant proposals, thereby making the notion of contributing to the information commons salient for scholars and the academic community.

6. COMMUNITY PATCHING

While many existing patching mechanisms rely on central servers provided by software vendors, the current design does not align with the incentive of patching and therefore leads to low participation rates from end users. We introduce a distributed patching scheme designed for community members (such as friends or colleagues). Three phases are designed for this mechanism: device introduction, vulnerability detection, and patching. Like the previous scenario of cooperative subversion detection, all phases in this scheme are operated by an application (*app*) installed on participating machines.

During the member introduction phase we begin with machine(s) owned by a solitary individual. Inviting another member's machine to join by email expands the network. Specifically, an existing member sends an email invitation through the app and the invitee replies either *Accept* or *Decline*. Upon confirmation the app sends a second email in which a registration link is embedded. The invitee follows the link and registers the machines he/she wishes to join the network. Finally, the inviter is notified when machines from new participants are added. Note that any individual could be invited given that the maximum number of users has not been reached.

In some cases the app has not been installed on the machine when a registration invitation is received. Following the registration link in this situation first directs the invitee to the app download page; the machine registration process cannot be started unless the app has been successfully installed. Encryption, digital signatures, time stamps and nonce could protect the communication between machines. These features could potentially prevent a message from being forged or replayed.

A device ID is assigned when a machine registers through the installed application. This ID is linked to the physical address of a device (e.g. MAC address), and does not alter with software changes (e.g. operating system reinstallation). Device IDs are utilized during the entire process: from machine introduction to patching.

The vulnerability detection phase is based on a P2P-based 'good worm' design. Specifically, each device keeps track of active participating machines and a vulnerability database through the app. A well-designed 'good worm' is then released by the scanner. Similar to the propagation mecha-

nism of malicious worms, a 'good worm' rapidly reproduces itself and actively scans the neighboring machines for known system vulnerabilities. Once a vulnerable machine is detected, both the scanner and machine being scanned will be notified about the incident and a patching phase is entered.

Upon discovery of vulnerability a patching request is generated by the scanner and broadcast to the entire network. As a result each machine in the network needs to perform a self-scan and requests a patch as needed.

Two forms of patch distribution approaches may be implemented. The first requires that the vulnerable machines keep a copy of the patch after the recovery process. Assuming that Alice received a new patch A and found Bob's machine vulnerable, then once she patches Bob's machine, Alice also makes sure that Bob has a copy of the patch applied. In the next round, Bob might fix Carol and Dave's machines with the same patch, and deliver it to both Carol and Dave. This approach is faster than the conventional centralized patching paradigm, and we could expect an exponential growth for the number of the patched machines.

Considering some patches are more urgent than others, we designed an express distribution approach that marks an importance value on each patch. Imagine the same situation in which Alice patches Bob's machine but this time with an *urgent* new patch. Instead of waiting until next round, Bob actively scans all other machines and sends the patch to all neighbors that he can reach. Note that the importance of the patch decreases with time. The logic behind the decrease is that a newer version of the patch may have been released, which could potentially cover functionalities of an old patch. Therefore, an *urgent* patch may be considered as *common* after a week, thereby failing to qualify for an express delivery any longer. Patches could be distributed in a more efficient way based on importance levels such as these.

Once the recovery process has been completed, a previously vulnerable machine broadcasts a confirmation message to the network. Consider a situation in which a malicious participant, Eve, claims to the network that she has discovered vulnerability A on Bob's machine. Then Eve pretends to be Bob and reports that a patch has been applied to fix A (when, in fact, no patch has been installed). Would Eve keep Bob unpatched this way? No. The reason is that other machines are also detecting their peers; even though Bob's machine claims itself to be patched, it does not prevent other machines with the patch from verifying the claim. Since detection and patching history is available to all members, a malicious machine could be easily discovered and a vulnerable machine (in this example, Bob's machine) will eventually be patched.

Simulating the behaviors of worms is analogous to an anti-theft exercise organized by a responsible neighbor. Therefore, with the help of peer production in a small community, we can create a secure commons. While this is not a silver bullet, this approach targets a subclass of security problems that can be solved through this new community-based or collective action paradigm of security solutions.

In terms of privacy and security, the relationship between privacy, anonymity, and data sharing is a long-contested topic in the PETS literature. Information sharing can be seen as a cost or as a social component. In the case of information sharing as a cost, the transition from public good to club good will create an incentive to pay that cost. In the case that information sharing is a social component, individuals are often more ready to share information with chosen networks of friends than with centralized services, firms, or government [13, 55, 2].

Further, it is possible to implement a version of this that combines crowdsourcing with peer production. For example, there could be a version of this proposal where the ISP organizes the groups and manages the reputation. Certainly the lack of ISP action in the security market (although arguably rational [53]) indicates that such an effort is needed. Further, without peer as opposed to corporate network monitoring, ISPs would be required to invest in security without being able to profit from this investment due to the lemons' market nature of security (i.e., the lack of transparency). Finally, as an aside but a potentially important one, adopting such monitoring could put ISPs at risk of losing their Safe Harbor under the DMCA.

7. ADDITIONAL IMPLEMENTATIONS

The theoretical approach to the design of security as a club good was motivated by completed projects which utilize peer input, but did not follow a common approach. Neither of these was grounded in the general design theory above, but both informed its development.

The first system was Net Trust [10]. Net Trust, which utilized the homophily of social groups to detect malicious web sites, was grounded in social informatics studies of trust [5, 14, 23, 56, 45, 44] and the economics of phishing [43, 42].

Net Trust was designed to identify malicious sites based on the fact that phishing sites and malware sites have very short lifetimes. Thus, unvisited sites should be considered sources of risk. A later empirical study illustrated that with only ten friends in a group, 95% of all clicks would be on links clicked by the person or one of these ten friends, which rises to 99% with forty friends [19]. This system also loaded authoritative lists of sites identified as malicious and interrupted connections to these. (However, that interruption could be over-ridden.)

The identification of sites as new and suspicious was intended to change the fight against phishing and malware distribution sites. The current mechanism is to allow phishing and malware sites to be instantiated, and then defenders chase the sites in order to identify and remove them. The economic change in this case would require sites to be visitable and visited for some time in order to be able to engage in an attack. This strategically changes the game between defenders and attackers for the benefit of the defenders.

Net trust is:

1. grounded in a microeconomic theory and social theories of trust that are embedded in
2. an interaction that has been subject to repeated user testing, which can
3. yield a privacy-enhanced social browsing application, which will enable
4. informed trust decisions based on social context that can be used to test
5. the value of Net Trust as a contextual signaling mechanism.

After completing the analysis and, to some extent reviewing our work, the confluence of social context and information sharing clearly reflected a peer-to-peer construction of

a security system. When a person was unsure of a site's reputation, she was able to get instant feedback from her social network that they have specified. The application graphically displays each friend's opinion on the site as well as an aggregate trust score. That is, private comments may be left as well as the implicit ratings resulting from an individual's web history. The application allows (nearly requires) a person have multiple identities coupled with multiple social networks. Overall, the interface displays the person's social-network, the opinions of the social network, and the aggregate score of the network on reputation of the site from the social network and any rating agency. The system appears as a toolbar above the tabs, the display changing with each tab click. The final result was a system that enabled a person to make a quick and informed decision on the reputation of a site with respect to his or her social network. The detailed architecture and data structures, illustrating the provision of privacy, are available at [58]. The second system was budget-based detection of the insider threat [39]. This second system, which utilized a budget-based mechanism to detect the insider threat, was grounded in contract theory [38]. Insider attacks are often possible due to the failure of the rigid, binary, and atomic nature of existing access control mechanisms. In these mechanisms, whether an access should be authorized or not is decided independently of other accesses. For example, in a multi-level security (MLS) policy without categories, if an employee is given security clearance at the Secret level, the employee can access all documents at that level. There is no limit on how many documents the employee can access, even though the vast majority of employees only need to access a small portion of the documents. On the other hand, if an employee is not given security clearance at that level, then the employee can access no document at that level.

In dynamic environments where an employee may need to access a wide range of resources, it is simply impossible to predict all resources an employee legitimately need to access. Given the binary nature of the access control mechanism, one has two choices. First, an organization can under-specify the policies, causing many legitimate accesses to be denied by the policy, requiring extra mechanism (such as break-glass) to enable. Alternatively, the organization can over-specify policies, exposing a vast quantity of information to each employee and thus enabling malicious insiders to abuse the access privileges.

To resolve this we proposed a risk budget mechanism whereby individuals received aggregate budgets, and each action incurred some type of risk charge. Within the risk budget mechanism, employees can no longer abuse their privileges without bearing any cost. As an example, consider an Internet commerce researcher whose job demands daily Internet surfing. Suppose the employee has a daily risk budget B_i for downloading documents from the Internet. He can visit a website w_j that costs him risk points p_j to perform the downloading, which costs him another p_k. Alternatively, he can visit another website w_u that requires p_u for visit and p_v for document downloading. The prices p_j, p_k, p_u and p_v are set by the organization based on its perception and evaluation of potential risks. Assuming $B_i > (p_j + p_k) > (p_u + p_v)$, we expect employee i voluntarily chooses the second website, which incurs lower risks, under our risk budget mechanism. Similarly, if the employee visits an order of magnitude more websites than any other employee, even if each is low risk, the employee is taking more aggregate risks.

The crowdsourcing nature of this work is in that each employee generates a baseline for all employees in the same category. Groups of employees, simply by choosing their own risk behaviors, inform the organization not only of outliers but of the normal distribution of risks by the organization. In this case risk management is effectively crowdsourced to the insiders themselves. This is based on a very explicit assumption that there are very few employees who seek to cause harm, even risk bringing down, their employer. The example of data breaches illustrates that employees who are trying to fulfill their work tasks, even working at home, may be a significant source of (oblivious) risk.

In other work we are redesigning this mechanism to explicitly utilize the approach described here [20]. The most basic change in design is that employees will have group as well as individual budgets. The changes in design also include an analysis of optimal group size and group formation based on organization and task diversity. For example, employees may self-select, thus resulting in more homogenous groups. Alternatively, employees may be assigned. Individual reputations may be private (as with the initial design), shared within a group, shared across the organization. Similarly, group reputations may be private, shared within a set of similar groups, or shared across the organization.

8. CONCLUSIONS

We argue that current approaches targeted at incentivizing individual users for improving the security of systems are limited as they view security problems through the lens of individuals' investment in public or private goods. We advocate for a paradigm shift to thinking of security as a community resource, i.e., as 'club goods' or 'common-pool resources', and show how these economic theories can be applied to improve the security of systems for groups of people. If people are incentivized to improve the security health of the community thought a shared endeavor, then there is the potential for greater personal investment in security. We have already operationalized these economic models for two instantiations, thereby demonstrating the strong potential for this new paradigm for improved security. A deeper understanding of this cooperative paradigm can have significant impact on real-world security, and we advocate further research to enable this shift.

9. ACKNOWLEDGEMENTS

We thank John McCurley for his editorial comments. This material is based upon work supported by the National Science Foundation under Grant 0916993. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

10. REFERENCES

[1] G. Akerlof. The market for 'lemons': Quality uncertainty and the market mechanism. *The quarterly journal of economics*, pages 488–500, 1970.

[2] K. Albrecht and L. McIntyre. *Spychips: How major corporations and government plan to track your every move with RFID*. Nelson Current, Nashville, TN, 2005.

[3] R. Anderson and T. Moore. The economics of information security. *Science*, 314(5799):610–613, 2006.

[4] J. E. M. Arnold. *Devolution of Control of Common Pool Resources to Local Communities: Experiences in Forestry*, pages 163–195. Oxford University Press, 2001.

[5] R. Axelrod and W. Hamilton. The evolution of cooperation. *Science*, 211(4489):1390–1396, 1981.

[6] Y. Benkler. Coase's penguin, or linux and the nature of the firm. *The Yale Law Journal*, 112(3):369—446, 2002.

[7] F. Berkes. Fishermen and 'the tragedy of the commons'. *Environmental Conservation*, 12(03):199–206, 1985.

[8] R. Böhme and T. Moore. The iterated weakest link–a model of adaptive security investment. 2009.

[9] L. Camp. Designing for trust. *Trust, Reputation, and Security: Theories and Practice*, pages 203–209, 2003.

[10] L. J. Camp. Reliable, Usable Signaling to Defeat Masquerade Attacks. In *WEIS: Workshop on the Economics of Information Security*, 2006.

[11] L. J. Camp. Reconceptualizing the role of security user. *Daedalus*, 140(4):93–107, 2011.

[12] L. J. Camp and C. Wolfram. Pricing security. In *Proceedings of the CERT Information Survivability Workshop*, pages 31–39, 2000.

[13] C. M. Cheung, P.-Y. Chiu, and M. K. Lee. Online social networks: Why do students use facebook? *Computers in Human Behavior*, 27(4):1337 – 1343, 2011.

[14] J. S. Coleman. *Foundations of Social Theory*. Belknap Press, Campbridge, MA, 1990.

[15] R. Cornes and T. Sandler. *The Theory of Externalities, Public Goods, and Club Goods*. Cambridge University Press, 1996.

[16] G. Danezis and R. Anderson. The economics of resisting censorship. *Security & Privacy, IEEE*, 3(1):45–50, 2005.

[17] T. Dietz, E. Ostrom, and P. C. Stern. The Struggle to Govern the Commons. *Science*, 302(5652):1907–1912, Dec. 2003.

[18] Z. Dong. Enabling users to self-manage networks: Collaborative anomaly detection in wireless personal area networks. In *Proceedings of Workshop on Usable Security*, Mar. 2012.

[19] Z. Dong and L. J. Camp. The decreasing marginal value of evaluation network size. *SIGCAS Comput. Soc.*, 41(1):23–37, Oct. 2011.

[20] Z. Dong and L. J. Camp. Peersec: Towards peer production and crowdsourcing for enhanced security. In *The 7th USENIX Workshop on Hot Topics in Security (HotSec)*, 2012.

[21] D. Feeny, S. Hanna, and A. McEvoy. Questioning the assumptions of the "tragedy of the commons" model of fisheries. *Land Economics*, pages 187–205, 1996.

[22] V. Garg, T. Koster, and L. J. Camp. Macroeconomic analysis of the weakest link: A case study of spambots. submitted to ACM Transactions of Information and System Security, 2012.

[23] H. Golberg and Shostack. Privacy ethics and trust. *Boston University Law Review*, 81(2):407–422, April 2001.

[24] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4:133–151, 2001.

[25] G. Hardin. The tragedy of the commons. *Science*, 162(3859):1243–1248, 1968.

[26] C. Herley and D. Florêncio. A profitless endeavor: phishing as tragedy of the commons. In *Proceedings of the 2008 workshop on New security paradigms*, pages 59–70. ACM, 2009.

[27] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 230–237, New York, NY, USA, 1999. ACM.

[28] C. Hess and E. Ostrom. Ideas, artifacts, and facilities: Information as a common-pool resource. *Law and Contemporary Problems*, 66(1/2):111–145, 2003.

[29] J. Hunter and J. Hunt. Term life insurance on the internet: An evaluation on on-line quotes. *Washington, DC: Consumer Federation of America*, 2001.

[30] L. Introna and H. Nissenbaum. Shaping the web: Why the politics of search engines matters. *The information society*, 16(3):169–185, 2000.

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

[32] M. Jakobsson and K.-A. Johansson. Practical and secure software-based attestation. In *Lightweight Security Privacy: Devices, Protocols and Applications (LightSec), 2011 Workshop on*, pages 1 –9, Mar. 2011.

[33] A. Kalamandeen, A. Scannell, E. de Lara, A. Sheth, and A. LaMarca. Ensemble: cooperative proximity-based authentication. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, MobiSys '10, pages 331–344, New York, NY, USA, 2010. ACM.

[34] D. Klooster. Institutional choice, community, and struggle: A case study of forest co-management in mexico. *World Development*, 28(1):1–20, 2000.

[35] D. Kreps, P. Milgrom, and J. RobertsRobert. Rational cooperation in the finitely repeated prisoners' dilemma. *Journal of Economic Theory*, 27(2):245–252, 1982.

[36] L. Lessig. *Code and other laws of cyberspace*. Basic books, 1999.

[37] I.-E. Liao, C.-C. Lee, and M.-S. Hwang. A password authentication scheme over insecure networks. *J. Comput. Syst. Sci.*, 72(4):727–740, June 2006.

[38] D. Liu, L. J. Camp, and X. Wang. Using budget-based access control to manage operational risks caused by insiders. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, 1:29–45, 2010. Previously version published at MIST 2010.

[39] D. Liu, X. Wang, and L. Camp. Mitigating inadvertent insider threats with incentives. In R. Dingledine and P. Golle, editors, *Financial*

Cryptography and Data Security, volume 5628 of *Lecture Notes in Computer Science*, pages 1–16. Springer Berlin / Heidelberg, 2009.

[40] J. M. McCune, A. Perrig, and M. K. Reiter. Seeing-is-believing: using camera phones for human-verifiable authentication. *Int. J. Secur. Netw.*, 4(1/2):43–56, Feb. 2009.

[41] T. Meyer and L. Van Audenhove. Graduated response and the emergence of a european surveillance society. *info*, 12(6):69–79, 2010.

[42] T. Moore and R. Clayton. An empirical analysis of the current state of phishing attack and defence. In *In Proceedings of the 2007 Workshop on the Economics of Information Security (WEIS 07)*, June 2007.

[43] T. Moore and R. Clayton. Financial cryptography and data security. chapter Evaluating the Wisdom of Crowds in Assessing Phishing Websites, pages 16–30. Springer-Verlag, Berlin, Heidelberg, 2008.

[44] P. Nikander and K. Karvonen. Users and trust in cyberspace. In *Revised Papers from the 8th International Workshop on Security Protocols*, pages 24–35, London, UK, UK, April 2001. Springer-Verlag.

[45] H. Nissenbaum. Securing trust online: Wisdom or oxymoron. *Boston University Law Review*, 81(3):635–664, June 2001.

[46] E. Ostrom. *Neither market nor state: Governance of common-pool resources in the twenty-first century.* International Food Policy Research Institute, 1994.

[47] E. Ostrom. Coping with tragedies of the commons. *Annual review of political science*, 2(1):493–535, 1999.

[48] E. Ostrom. How types of goods and property rights jointly affect collective action. *Journal of Theoretical Politics*, 15(3):239–270, 2003.

[49] C. Patrikakis, D. Kyriazanos, and N. Prasad. Establishing trust through anonymous and private information exchange over personal networks. *Wireless Personal Communications*, 51:121–135, 2009.

[50] K. Peter. The graduated response. *Fla. L. Rev.*, 62:1373, 2010.

[51] A. R. Poteete, M. A. Janssen, and E. Ostrom. *Working Together: Collective Action, the Commons, and Multiple Methods in Practice*. Princeton University Press, Apr. 2010.

[52] A. Rapoport and A. Chammah. *Prisoner's dilemma: A study in conflict and cooperation*, volume 165. Univ of Michigan Pr, 1965.

[53] B. Rowe. Isps as cyberecurity providers. In *The Ninth Workshop on the Economics of Information Security (WEIS)*, 2010. (Rump Session Presentation).

[54] M. Sasse, S. Brostoff, and D. Weirich. Transforming the 'weakest link' — a human/computer interaction approach to usable and effective security. *BT technology journal*, 19(3):122–131, 2001.

[55] K. Shankar, L. Camp, K. Connelly, and L. Huber. Aging, privacy, and home-based computing: Designing for privacy. *Pervasive Computing, IEEE*, PP(99):1, 2011.

[56] B. Shneiderman. Designing trust into online experiences. *Communications of the ACM*, 43(12):57–59, December 2000.

[57] M. R. Thompson, A. Essiari, and S. Mudumbai. Certificate-based authorization policy in a pki environment. *ACM Trans. Inf. Syst. Secur.*, 6(4):566–588, Nov. 2003.

[58] A. Tsow and L. J. Camp. A privacy-aware architecture for sharing web histories. *IBM Journal of Research & Development*, 2009.

[59] H. Varian. System reliability and free riding. *Economics of Information Security*, pages 1–15, 2004.

[60] H. R. Varian. *Microeconomic Analysis, Third Edition.* W. W. Norton & Company, 3rd edition, Feb. 1992.

[61] A. Varshavsky, A. Scannell, A. LaMarca, and E. De Lara. Amigo: proximity-based authentication of mobile devices. In *Proceedings of the 9th international conference on Ubiquitous computing*, UbiComp '07, pages 253–270, Berlin, Heidelberg, 2007. Springer-Verlag.

[62] C. Vielhauer. *Biometric User Authentication for IT Security: From Fundamentals to Handwriting (Advances in Information Security)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.