SAND2014-3046C

SWARM: Signature Writing Analysts Researching Malware

Jeff W. Boote
Sandia National Laboratory
Livermore, CA
jwboote@sandia.gov

Carrie Gates
Dell Research
Round Rock, TX
carrie.gates@ca.com

ABSTRACT

Cybersecurity is an arms race — attackers create attacks, defenders respond. While defenders try to generalize detection, they have so far been unsuccessful. The best defense is to "keep the anti-virus signatures up to date".

Rather than attacking the problem of general attack detection, we aim to change the playing field by modifying the underlying economics. Antivirus companies must hire experts to analyze malware and develop signatures. Given the number of samples submitted, the cost of experts, and the time to analyze, A/V companies struggle to put out timely signatures. But what if defenders could tap a larger pool of talent... for free?

In this paper we will present the SWARM system — an environment for security experts and peers to compete and collaborate. The design is based on a trusted community for addressing cyber-security concerns, where users can develop a reputation for their work, as well as for their responses when helping others, thus also providing a training aspect to the system. We focus here on the A/V community and the need for malware signatures, but our approach should generalize.

1. INTRODUCTION

The art and science of cybersecurity puts defenders at a disadvantage. Defenders must constantly react to new attacks while determining how to enhance defenses against new vulnerabilities created by each new application. The current best practice utilizes defense in depth comprised of firewalls and application specific ingress/egress scanning of specific application space like email virus scanning. This allows security administrators enough information to detect most attacks, but is still vulnerable to 0-day attacks and other attacks that are designed to stay under detection thresholds.

General attack defense is hindered by the fact that it takes as much or more resources to defend as it does to attack. Because an arms race is won by attrition, it is important to gain an economic advantage.

Antivirus companies, for example, hire experts to analyze malware and develop signatures. Given the number of malware samples typically submitted (for example, McAfee reports finding nearly 1,000,000 unique signatures per day during Q2-2012 [1]), the cost of hiring experts, the time required to analyze malware, and the lack of qualified threat analysts, A/V companies struggle to keep current with adversaries and to put out signatures in a timely fashion. The New York times reported in January of 2013 "On average, it took almost a month for antivirus products to update their detection mechanisms and spot the new viruses." [2] Note that this is a full month after the virus has been reported to the anti-virus company. The industry does not have a 0-day problem — its problem is far worse.

But what would happen if defenders could tap a larger pool of talent ... for free?

In this paper, we present the SWARM (Signature Writing Analysts Researching Malware) system. The vision is to create a trusted cyber community where individuals cooperatively combat cyber threats for the collective community. Defenders compete with each other to address cyber threats urged on using community and crowdsourcing motivations. Conquests are shared in much the same way that people share triumphs in other communities such as: Nike+ (runners), Strava (cyclists), and World of Warcraft (gamers).

We leverage social community building and crowdsourcing techniques within a gaming culture to modify the economy of cyber-security in the following ways:

1. Provide a cheaper training ground for cyber-defenders, reducing the cost of developing knowledgeable cyber experts for defenders.

2. Reduce the cost of defense for national security issues by reducing the number of nuisance threats to national assets.

3. Provide a larger pool of talent for government and industry to tap for cyber-defense

4. Provide incentives for hackers to use their skills for defense rather than attack.

In this paper, we describe a system where malware samples are submitted and users analyze malware samples, with the goal of developing appropriate signatures. The system leverages reputation incentives, allowing users to gain reputation points based on speed and quality of submitted signatures that are validated and used in A/V engines. Reputation points allow users to become recognized experts in a community providing increased motivation (beyond just the technical challenge) for participation. We will discuss both how this system would be designed, and how we would address potential "gaming" of the system and other limitations, with a special focus on trust.

We focus specifically on the antivirus community and their need for malware signatures, but note that our approach should generalize to other cybersecurity problems. Thus we conclude the paper with a discussion on how this approach can potentially be expanded to other cybersecurity domains.

2. SYSTEM DESIGN

An overview of the system architecture is provided in Figure 1. The basic concept is that malware samples are submitted to a repository. These samples can come from end users directly into the SWARM system, or they can come via A/V companies who are collecting malware samples. Users (e.g., the "Analyst" in SW**A**RM) can select malware samples to analyze as if they were an A/V researcher — that is, with the goals of determining what malware family a particular piece of malware belongs to and developing an appropriate signature. Once a signature has been developed, it gets submitted to the beta signature repository. A/V companies can then collect new signatures for new malware (or malware for which no current signature exists) from the beta signature repository, validate those signature using professional A/V researchers, and then, if the signatures prove to be good, update the signature databases for existing products.

In order to encourage users to participate in such a system, there needs to be incentives applied. In this particular instance, we intend to apply reputation-like systems where users who submit signatures that are provided quickly, validated, and used in an A/V engine, gain reputation points and become known as the experts within a community. This echoes what has happened on github[1], a collaborative open source development platform, where users become known as experts based on their responses to questions.

Obviously, a participating A/V company will not want to use signatures from the system without reviewing those signatures first. However, we expect that the review of submitted signatures — particularly if the thoroughness of the review is combined with the reputation of the submitter — will be less costly than the development of a signature from scratch in-house.

Trust is an important part of this system. It is always possible to game the trust of others in order to later exploit them. For example, an individual can gain a trusted reputation for providing good signatures, just to later provide a signature that does not actually detect the malware it proposes to detect. In order to avoid these situations, it would be recommended that participating A/V companies perform a comparison against some top number of submitted signatures, thus requiring collusion in order for such an attack to succeed.

The community will be anchored by a web-based presence providing a place to share tools, indicators, and training material. Members will also get ongoing feedback regarding their performance. We envision feedback provided in a highly interactive game-like environment. This simultaneously collaborative and competitive atmosphere will provide opportunities to develop a reputation while helping others learn. The system we describe takes into account the design recommendations for peers and crowds in security settings as identified by Dong and Camp [3].

2.1 Analysis

The analysis of malware samples will be performed using a crowdsourcing approach, where crowdsourcing is defined as "the act of exporting tasks traditionally performed by one or more employees to an indefinite group of persons or a community through an open call." [4] Malware samples will be checked into a semi-public repository. Users can sign up to be analysts; however, before they can gain access to the malware repository, they will need to complete online training to demonstrate their understanding of malware and the need for a secured environment that prevents the release of any downloaded malware, along with an understanding of the ethical component of such research. Upon completion of the training, the analyst receives limited access to the malware repository. As the analyst demonstrates competence (via an increased reputation score), he gains increasing access to malware samples.

When a new sample is added to the system, alerts are sent to the analysts who have access to the sample (e.g., via email or SMS messages, based on individual user preference). Analyst access is determined based on a combination of reputation (those analysts with better reputations gain quicker access to new samples) and analyst preference (e.g., perhaps based on the malware metadata, such as the geographic area from which it was submitted or the platform target of the malware). Analysts can then choose to download the sample and begin to reverse engineer the malware in order to determine the malware family to which it belongs (or if it is completely new) and thus design an appropriate and effective detection signature.

It is this use of multiple analysts to examine the same piece of malware that is the crowd-sourcing aspect of the architecture. We note that this is different from the open source model, as we make use of many people each working competitively (and also perhaps collaboratively a la an agile programming approach) to solve a given problem. In contrast, open source tends to have individuals each working on distinct problems.

Once an analyst has developed what he feels to be an accurate signature capable of detecting specific malware (or, perhaps, modified another signature to detect both that malware as well as other previously identified malware from the same family), he submits this signature to a beta signature repository. In addition to the signature, the analyst is

[1] `https://github.com`

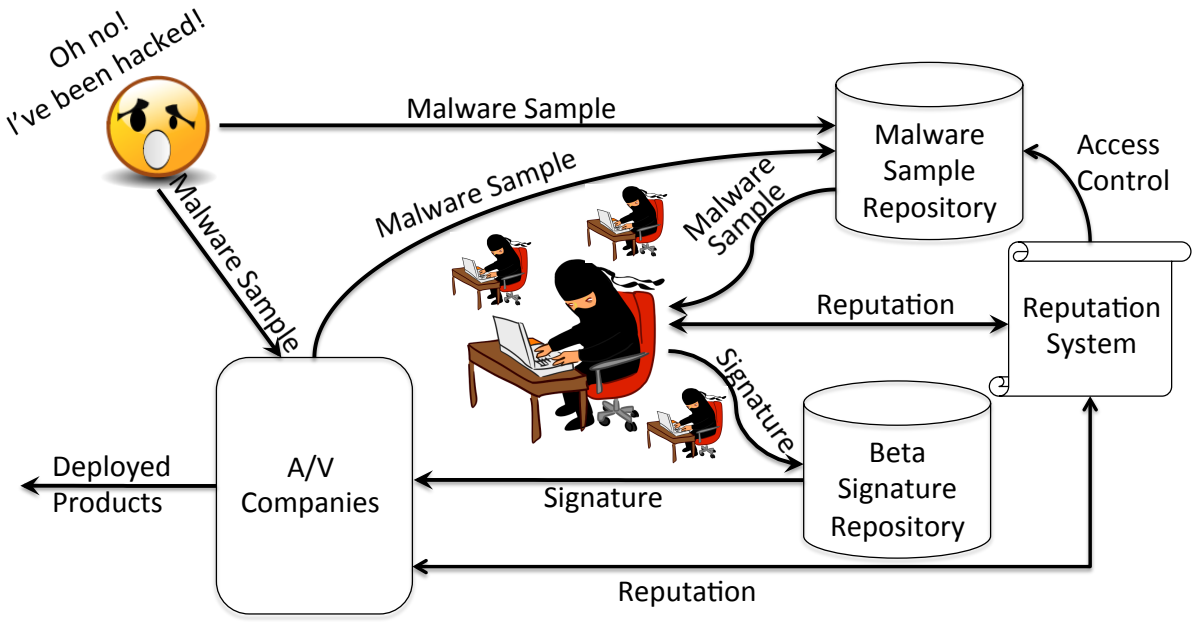

Figure 1: Overview of the architecture for the SWARM system.

expected to provide documentation indicating the malware family, comments on the signature, and a link back to the malware sample repository. At this point, A/V companies will be automatically notified that a new signature has been submitted for review (see section 2.3 for details).

2.2 Reputation System

A key component of the SWARM system is a reputation system. The reputation system will provide a competitive, game-like environment that challenges analysts to obtain KOM (king-of-the-mountain) status. This is similar in concept to Strava[2], a web site that caters to cyclists. In Strava, users create routes that are made public. When a cyclist bikes one of these routes, he can later compare his statistics (e.g., time, power, heart-rate) to those of other cyclists over the same route. This provides a competitive environment where cyclists try to be KOM. Simultaneously, it provides incentive for cyclists to improve, along with tangible end goals. Thus there is effectively a training component as well.

In our system, reputation is divided into two parts:

1. reputation from generation of exceptional signatures, and
2. reputation from training, where answers to other analyst questions are judged.

In the first instance, reputation is gained based on evaluations from A/V companies to the signatures that are submitted by analysts. We cover the role of A/V companies in greater detail in the next subsection; however, one of the outputs of A/V companies is a "grading" of submitted signatures, with better grades resulting in increased reputation scores. This grading is based on a combination of factors, including the grade for the signature itself (e.g., simplicity, completeness, elegance, lack of false positives generated, etc.), as well as the speed with which the signature was developed and submitted (e.g., based on the relative time to find a solution, rather than absolute time — thus allowing for more complex malware samples to take longer to analyze), and the quality and completeness of the documentation provided. As with all reputations, this will necessarily be a subjective measure. Conversely, poor signatures will adversely affect an analysts reputation score.

In the second instance, reputation is gained based on responses to questions from more junior analysts. The ability to ask questions and have experts respond provides a training component to the SWARM system. As junior analysts start, they gain access to older malware samples that they can analyze in order to learn more about malware analysis. The SWARM system provides them with an environment where they can ask questions, and more senior analysts can answer those questions. The senior analysts gain reputation points based on the quality of the answers (as judged by the junior analyst who asked the question, as well as by peers). This education can extend to senior analysts providing feedback on signatures generated by junior analysts on older malware, and this feedback then rated as well. Additionally, the junior analyst can start gaining reputation points based on submitted signatures. As the junior analyst becomes more competent, he can also start answering questions, increasing his own reputation. As his reputation score increases, he gains greater access to the malware samples repository, so he can eventually become a senior analyst with full access to provide signatures for newly submitted mal-

[2]`http://www.strava.com`

ware samples. Thus the system has built-in incentives for answering questions and increasing reputation. Further, we have observed similar systems work, such as github, which is described in Section 3.

2.3 Productization

The A/V companies play an important role in the SWARM system, as the greatest benefit is obtained through their participation. Indeed, without their participation, SWARM becomes merely a training ground.

A/V companies provide three functions in this system:

1. they can provide malware samples to be analyzed to the malware repository,
2. they test beta signatures provided by analysts for release in their signature database, and
3. they provide feedback to analysts in the form of reputation scores.

Focusing on the second item, it would be easy for A/V companies to simply take any beta signatures provided and, assuming the reputation score of the analyst submitting the signature meets some minimum threshold, simply add the signature to their database. However, in order to maintain control over the quality of submitted signatures (as well as over their own reputation), A/V companies are encouraged to review any submitted signatures to ensure that it performs as expected (e.g., correctly identifies the malware in question) with an acceptable performance rate (speed and accuracy).

The advantage of this for the A/V companies is that they have farmed out signature generation to a potentially large pool of A/V researchers and analysts for just the cost of verification of signatures provided (and potentially monetary award to the analyst who submitted the signature that was later included in the signature database). Non-A/V companies have employed similar approaches to improve their cybersecurity posture — Facebook's "bug bounty" program (described later) provides Facebook with a significantly larger pool of penetration testers, for example.

The A/V company is also encouraged to provide feedback to the analysts. In particular, lower reputation scores (and appropriate comments) should be provided to any analysts who submit signatures that fail to be sufficiently robust to be included for release in the signature database. At the same time, a high reputation score should be provided to the first analyst to submit a signature that detects some new piece of malware for which no signature currently exists.

3. INCENTIVES

In order for the SWARM system to be used, incentives need to be provided for both analysts and A/V companies.

For analysts, incentives include: (1) the technical challenge, (2) the possibility of establishing a reputation, and (3) potentially financial gains and employment opportunities. First, we aim to attract hacker types who are motivated by technical challenges. We provide an outlet for their creativity and skills in a white-hat environment where their work might also end up being used to improve security for numerous end users via the A/V companies. We further hope that this will provide an outlet for hackers who might otherwise use their skills within the blackhat community.

Secondly, analysts have the opportunity to develop a reputation within the community for excellence, not only in terms of the signatures they generate, but also in terms of contributing back to the community in the form of education and answering junior analysts' questions. Analysts can use their reputation scores for self-satisfaction and in the form of a competition. But, thirdly, analysts can also leverage their reputation for financial gain. First because we hope the A/V companies will provide modest financial rewards for submitted signatures that are later used in their products (similar to how Facebook's bug bounty program works, which is described in Section 4). But also because we expect analysts with high reputation scores will also be able to leverage those scores to find appropriate employment (e.g., within the A/V companies themselves, for example). Towards this end, we note that many people are currently posting their github reputation scores on their LinkedIn profiles.

For A/V companies, there is a financial incentive to participate. In particular, it should be cheaper to review and test submitted signatures from reputable users than to completely develop signatures in-house. This has the further advantage of signatures being available to the general public more quickly, which is a benefit both to the general public and to the reputation of the A/V company. By having multiple experts work on signature development, there is also the potential for the development of better signatures that can detect larger classes of malware, which again also benefits the A/V company and the general public.

4. PROOF POINTS

While this system has not yet been developed, there are proof points that indicate the possibility for success of a crowdsourcing approach to cybersecurity. Two examples include Facebook's "bug bounty" program[3], where Facebook will pay a bounty (minimum $500 reward) for security vulnerabilities discovered in its software, thus leveraging a very large number of penetration testers for a very small sum. Google employs a similar tactic, called the Vulnerability Reward Program[4]. In both cases, these companies are essentially using CrowdSourcing approaches to outsourcing certain aspects of their security posture. Further, we note that a modest sum is provided, which provides increased incentive for people to participate in the CrowdSourcing.

Another security-related system is PhishTank[5], which employs a crowdsourcing approach to identifying phishing websites. First launched in 2006, PhishTank has received over 1.8 million submissions to date, with over seven million votes, identifying nearly 10,000 online, valid phishes. In

[3] `https://www.facebook.com/whitehat/`
[4] `http://www.google.com/about/appsecurity/reward-program/`
[5] `http://www.phishtank.com`

this case, an active community identifies potential phishing emails, and participation is completely open. Web of Trust, or WoT[6], is a similar system to PhishTank, but focuses on being a reputation system for websites, rather than specifically detecting phishing.

GitHub[7] is another system that has leveraged reputation motivation to encourage development of software by identifying individuals as "experts". GitHub provides a software repository system — both enterprise versions and a freeware version that can be used for opensource projects — that currently hosts 12.9 million code repositories online. This includes a community of approximately 4.5 million developers, each with a profile (that many have started using as the "new resume"). The community includes activity streams, where users can watch for updates from specific developers, and face-to-face meet-ups.

While finding usage statistics for github is difficult, according to Wikipedia[8], they surpassed SourceForge and Google Code within about four years. GitHub was launched in April 2008, reached 100,000 users after 15 months, and one million users after three years. By September 2012, they had 2.1 million users. They reached three million users by January 2013, and 4.5 millions users by June 2013. This indicates that there is an inflection point at which point popularity increased dramatically, although initial growth was comparatively slow.

5. LIMITATIONS

There are three main limitations to this approach that need to be considered and addressed: sustainability, viability and trust.

5.1 Sustainability

For this system to work, it needs to first reach a critical mass of participants and usage — this critical mass is is unknown. For GitHub, as described above, the inflection point appears to be near one million people (reached after about three years). Given the more specific nature of the SWARM system and the fewer number of qualified participants, the actual number of analysts required is likely considerably lower (and should, perhaps, be based on the percentage of qualified individuals participating in terms of estimating a critical mass); however, the amount of time required for the system to reach critical mass is still likely comparable, indicating that SWARM will need many months and potentially years in order to be a viable system.

Additionally, while Google and Facebook are given as example systems that are similar in nature, we note that (1) malware analysis is considerably more specific than penetration testing requiring a different skillset, and (2) breaking into Google and Facebook is more likely to be considered "cool" than reverse engineering a piece of malware, and so it may be more difficult to recruit potential analysts.

A final issue is that the community will need to be self-sustaining. This will require ongoing development of the site, tools, indicators and training materials, all done by the community itself. Thus proper incentives need to be in place to help start and support the community. While considerable research exists regarding motivation, along with research on crowd sourcing and open source, it is still unclear what support and incentives need to be provided to guarantee the formation and continuation of such a specific community.

5.2 Viability

Related to the notion of sustainability is that of viability. The goal of this system is to magnify A/V resources and, if successful, serve as a role model for building similar communities in other areas of cybersecurity in order to magnify national resources. But given the issues raised under sustainability, there is no guarantee that SWARM would be viable. It might be the case that it never gains sufficient popularity (or even any popularity).

It might also be the case that SWARM becomes popular amongst a subset of users who are more interested in abusing the system, such as gaining access to malware in order to leverage it for writing new malware. The system as described is open, so that non-professionals might learn how to analyze malware, and begin contributing to supporting the cybersecurity community. There are no background checks performed on individuals requesting access, and so it is possible that potential attackers will gain equivalent access.

This system is also only viable with the support of the A/V community, and specifically A/V companies. Initial investment from them in terms of A/V researcher time is required, and a lack of interest or commitment will kill the system in its infancy.

5.3 Trust

Any system designed will require some level of trust by the participants. Promoting trust requires that sufficient controls be in place to provide reasonable protection to the participants. Participants in the system could be harmed in the following ways:

1. Analysts could become victims in the reputation system if they become targeted by some other person or group. At the low end of the spectrum, they might find any answers they post attacked by someone on a personal level with insults. On a higher end of the spectrum, someone might become the victim of an attack specifically aimed at lowering their reputation scores, and potentially therefore jeopardizing employment possibilities.

2. The victim in Figure 1 might also submit inappropriate malware samples. This could take the form of samples that serve to waste people's time, to trying to write sample malware specifically aimed at doing harm to people in the SWARM system. This might be mitigated to some extent by including victims in the reputation database.

3. The system itself, given the repository of malware, could be abused by someone either with legitimate access or who has gained inappropriate access. This

[6] `http://www.mywot.com`
[7] `https://github.com`
[8] `https://en.wikipedia.org/wiki/GitHub#Statistics`

attacker could collect the malware in the repository, along with accompanying documentation, to then modify themselves for adversarial purposes, or to sell on the black market. Having a collection of malware in one location that has somewhat more open access than is typically found within A/V companies and researchers provides a potential target.

4. The A/V companies need to ensure that appropriate controls are in place so that they do not rely on trusting any single analyst and the signatures he submits, as this would allow an attacker to gain a strong reputation to then abuse that reputation by corrupting a submitted signature, or potentially even larger parts of the signature database. Mitigating strategies will need to be put into place, such as crowdsourcing submitted signatures (by comparing the top N submitted signatures for example) or by ensuring that all submitted signatures are reviewed by at least one additional person before being added to the signature database and released.

6. COMPARISON TO RELATED RESEARCH

In general, the application of crowdsourcing to the security domain is rare.

It was first suggested by Methusula Cebrian Ferrer at EICAR 2010 that crowdsourcing is a technique that could be applied to A/V research and cybersecurity in general [5, 6]. In her presentation, Ferrer asks the question of whether crowdsourcing techniques might be applicable in this round. In contrast, our paper provides a detailed system architecture for such a system, including a discussion on motivations and incentives.

Fink *et al* [7] have discussed the application of crowdsourcing (in conjunction with machine learning and natural language processing) to the detection of cybersecurity threats, and describe their crowdsourcing architecture in more detail in [8]. Their sytem focuses on detecting scam websites and cross-site request forgery. In contrast to our approach, they do not provide any training aspects nor reputation systems, but rather use end user votes and comments to gather general information on website reputation, supplemented with machine learning approaches for actual detection of web-based threats.

Burguera *et al* [9] have suggested that crowdsourcing be used in order to *collect* malware, focusing on the Android platform, but once collected they apply automated techniques (specifically application behavioral analysis) to the detection of the malware itself. The use of automated techniques for malware analysis that result in A/V signatures has been discussed in the academic literature; however no widely applicable technique is currently available due to the evolving nature of malware (e.g., increased obfuscation, self-detection of virtual environments).

Research has also been done on the "wisdom" of trusting crowds with security results. In particular, Moore and Clayton [10] evaluated the submissions to PhishTank, discussing how it is particularly vulnerable to manipulation. However, it was noted by Chia and Knapskog [11] that it is possible to implement countermeasures as done in WoT. Further, WoT also uses a reputation system, unlike PhishTank, that counters the ease of manipulation of such a system. We note that SWARM leverages many of the same features as WoT, such as reputation, in order to reduce the risks in using crowd sourcing for security purposes.

7. CONCLUSIONS

One of the difficulties in cyber security is that there are insufficient numbers of qualified defenders, while attackers can be few in number while still inflicting considerable damage. We described a system for combating this imbalance — focusing specifically on the anti-virus community — by leveraging crowd-sourcing techniques to increase the amount of resources used for solving cybersecurity issues (e.g., analyzing malware and providing A/V signatures) while also providing a training ground to allow people to learn how to analyze malware. The SWARM system provides a reputation system to allow analysts to gain a reputation within the A/V community for understanding malware, analyzing malware and quickly writing appropriate signatures for detection, and for training new analysts.

In this paper, we provided a detailed architecture for the SWARM system, along with a discussion on motivation for using this system. We described related systems in other domains as proof-points for how SWARM might work, along with a discussion on the possible limitations of this system.

The SWARM system focuses specifically on the case of malware analysis, rather than general cyber-security; however, we expect that the concept can generalize to other areas within cyber-security. One possibility here is forensics, although the community might require a higher level of "vetting" of applicants before they are approved for participation in such a system. In general, however, it is expected that SWARM-like systems can be developed for other specific areas within cybersecurity, although it might be more difficult to develop a generic cybersecurity system that follows the same model. What is more likely is that, once a sufficient number of subsystems have been developed and deployed, a meta-system could be developed that contains these subsystems, thus allowing potential defenders to gain a more generic cybersecurity reputation based on their reputations within the different subsystems.

[1] M. Labs, "Mcafee threats report: Second quarter 2012," http://www.mcafee.com/us/resources/reports/rp-quarterly-threat-q2-2012.pdf, 2012, last visited: 15 March 2013.

[2] N. Perlroth, "Outmaneuvered at their own game, antivirus makers struggle to adapt," http://www.nytimes.com/2013/01/01/technology/antivirus-makers-work-on-software-to-catch-malware-more-effectivel html, 2013, last visited: 19 June 2013.

[3] Z. Dong and L. J. Camp, "Peersec: towards peer production and crowdsourcing for enhanced security," in *Proceedings of the 7th USENIX conference on Hot Topics in Security*, ser. HotSec'12. USENIX Association, 2012.

[4] J. Howe, *Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business.* New York, NY: Crown Publishing Group, 2008.

[5] M. C. Ferrer, "Is there a future for crowdsourcing security?" http://www.eicar.org/files/meths_eicar2010.pdf, 2010, last visited: 19 June 2013.

[6] E. Willems, "Eicar 2010: Rainy days in paris," *Virus Bulletin (June 2010)*, 2010.

[7] E. Fink, M. Sharifi, and J. G. Carbonell, "Application of machine learning and crowdsourcing to detection of cybersecurity threats," in *Proceedings of the DHS Science Conference*, 2011.

[8] M. Sharifi, E. Fink, and J. G. Carbonell, "Smartnotes: Application of crowdsourcing to the detection of web threats," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, 2011, pp. 1346–1350.

[9] I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani, "Crowdroid: behavior-based malware detection system for android," in *Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices*, ser. SPSM '11, 2011, pp. 15–26.

[10] T. Moore and R. Clayton, "Financial cryptography and data security," G. Tsudik, Ed. Berlin, Heidelberg: Springer-Verlag, 2008, ch. Evaluating the Wisdom of Crowds in Assessing Phishing Websites, pp. 16–30.

[11] P. H. Chia and S. J. Knapskog, "Re-evaluating the wisdom of crowds in assessing web security," in *Proceedings of the 15th international conference on Financial Cryptography and Data Security*, ser. FC'11. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 299–314.