Evaluating the Wisdom of Crowds in Assessing Phishing Websites

Tyler Moore and Richard Clayton

Computer Laboratory, University of Cambridge
15 JJ Thomson Avenue, Cambridge CB3 0FD, United Kingdom
`firstname.lastname@cl.cam.ac.uk`

Abstract. We examine the structure and outcomes of user participation in PhishTank, a phishing-report collator. Anyone who wishes may submit URLs of suspected phishing websites, and may vote on the accuracy of other submissions. We find that PhishTank is dominated by its most active users, and that participation follows a power-law distribution, and that this makes it particularly susceptible to manipulation. We compare PhishTank with a proprietary source of reports, finding PhishTank to be slightly less complete and significantly slower in reaching decisions. We also evaluate the accuracy of PhishTank's decisions and discuss cases where incorrect information has propagated. We find that users who participate less often are far more likely to make mistakes, and furthermore that users who commit many errors tend to have voted on the same URLs. Finally, we explain how the structure of participation in PhishTank leaves it susceptible to large-scale voting fraud which could undermine its credibility. We also discuss general lessons for leveraging the 'wisdom of crowds' in taking security decisions by mass participation.

1 Introduction

Phishing is the process of enticing people to visit fraudulent websites and persuading them to enter identity information such as usernames and passwords. The information is then used to impersonate victims in order to empty their bank accounts, run fraudulent auctions, launder money, and so on. Researchers have proposed many technical countermeasures, from mechanisms to detect phishing websites [17,28], through to schemes that prevent users from disclosing their secrets to them [20]. The primary response from the banks, in contrast, has been to initiate 'take-down' procedures, removing the offending content so that there is nothing there for a misled visitor to see [15].

Attackers remain an elusive target, setting up new websites as quickly as the existing ones are removed. So obtaining an updated feed of new websites requires constant vigilance and demands significant resources. Most banks and specialist take-down companies maintain their own feed. One group, called 'PhishTank' [18], has tried to leverage the 'wisdom of crowds' to generate an open source list that strives to be as complete and accurate as possible. Users are invited not only to provide the content but also to undertake the somewhat more menial task of verifying that entries are correctly classified.

Financial Cryptography and Data Security (FC) 2008, LNCS 5143, pp. 16–30.

PhishTank is part of a growing trend in turning to web-based participation to implement security mechanisms, from aggregating spam to tracking malware. In this paper, we study participation in PhishTank in order to better understand the effectiveness of crowd-based security more generally. In doing so, we make several specific contributions:

- we find participation in PhishTank is distributed according to a power law;
- we compare PhishTank's open list to a proprietary (closed) list, finding the closed list slightly more comprehensive, and faster in verifying submissions;
- we identify miscategorizations made in PhishTank;
- we determine that inexperienced users are far more likely to make mistakes;
- we find evidence that 'bad' users vote together more often than randomly;
- we explain how the structure of participation in PhishTank makes it especially vulnerable to manipulation;
- we outline several general lessons for implementing more robust crowd-sourced security mechanisms.

2 Data collection and analysis

2.1 Phishing website reporting and evaluation

We gathered phishing reports from PhishTank [18], one of the primary phishing-report collators. The PhishTank database records the URL for the suspected website that has been reported, the time of that report, and sometimes further detail such as `whois` data or screenshots of the website.

PhishTank has explicitly adopted an open system powered by end-user participation. Users can contribute in two ways. First, they submit reports of suspected phishing websites. Second, they examine suspected websites and vote on whether they believe them to be phishing. PhishTank relies on the so-called 'wisdom of crowds' [25] to pick out incorrect reports (perhaps pointing to a legitimate bank) and confirm malicious websites. Each report is only confirmed (and subsequently disseminated to anti-phishing mechanisms) following the vote of a number of registered users. The tally of as-yet undecided votes is not revealed to users until after casting a vote. This helps prevent information cascades where early opinions influence later ones [3].

Consistent with PhishTank's open policy, they publish a record of all completed votes. This includes the identifiers of the user who submitted the report, the result of the vote (is or is-not a phish), the users who voted, and the percentage of votes cast for and against categorizing the website as a phish. However, the records do not specify how each user voted.

We examined reports from 200 908 phishing URLs submitted between February and September 2007. Voting was suspended for 24 254 of these because the websites in question went offline before a conclusive vote could be reached. In these cases, we could only determine who submitted the record and not who voted on it. We gathered completed votes for the remaining 176 366 submissions. 3 798 users participated by submitting reports and/or voting.

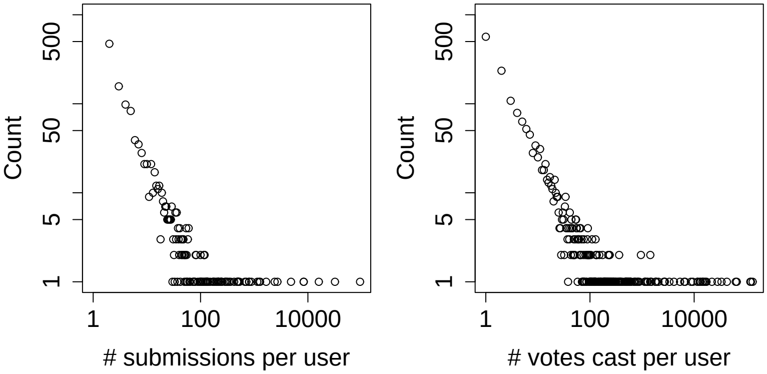

Fig. 1. Density of user submissions (left) and votes (right).

In all, 881 511 votes were cast, implying an average of 53 submissions and 232 votes per user. However, such averages are very misleading. Small numbers of users are responsible for the majority of submissions and votes. The top two submitters, adding 93 588 and 31 910 phishing records respectively, are actually two anti-phishing organizations that have contributed their own, unverified, feeds of suspect websites. However, neither verifies many submissions. The top verifiers have voted over 100 000 times, while most users only vote a few times.

Many of the leading verifiers have been invited to serve as one of 25 PhishTank moderators. Moderators are granted additional responsibilities such as cleaning up malformed URLs from submissions.[1] Collectively, moderators cast 652 625 votes, or 74% of the total. So while the moderators are doing the majority of the work, a significant contribution is made by the large number of normal users.

2.2 Power-law distribution of user participation rates

The wide range of user participation is captured in Figure 1. Noting the log-log axes, these plots show that most users submit and vote only a handful of times, while also indicating that a few users participate many times more.

In fact, the distribution of user submissions and votes in PhishTank are each characterized by a power law. Power-law distributions appear in many real-world contexts, from the distribution of city populations to the number of academic citations to BGP routing topologies (see [16] for a survey). More precisely, the probability density function of a power law corresponds to $p(x) \propto x^{-\alpha}$, where α is a positive constant greater than one. Power-law distributions have highly skewed populations with 'long tails', that is, a limited number of large values appear several orders of magnitude beyond the much-smaller median value.

[1] Moderators also, on some rare occasions, use their powers to pre-emptively remove obviously incorrect submissions.

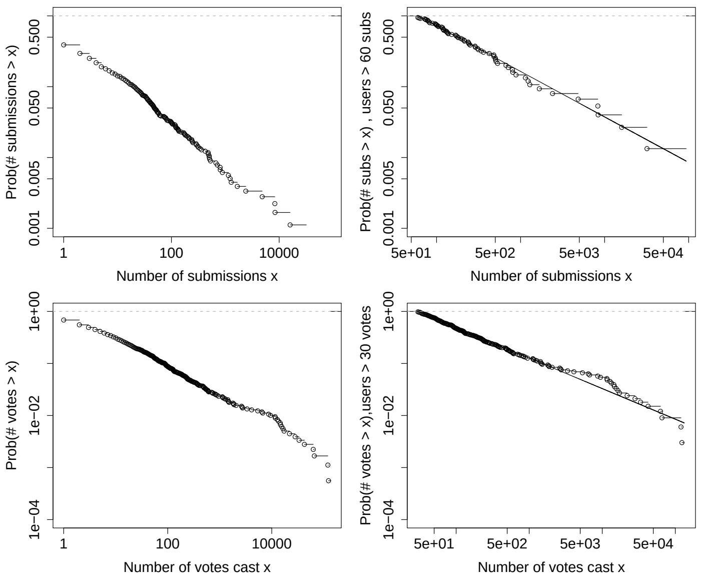

Fig. 2. Complementary CDF of user submissions (top left) and votes (bottom left). Tail of submission CDF with power-law curve fit (top right), $\alpha = 1.642$ and the number of submissions per user at least 60. Tail of vote CDF with power-law curve fit (bottom right), $\alpha = 1.646$ and the number of votes per user at least 30.

The intuitive argument put forth in favor of the robustness of ‘crowd-sourced’ applications like PhishTank’s phish verification mechanism is that the opinions of many users can outweigh the occasional mistake, or even the views of a malicious user. However, if the rate of participation follows a power-law distribution, then a single highly active user’s actions can greatly impact a system’s overall accuracy. This is why a power-law distribution invalidates the standard Byzantine Fault Tolerance view of reliability [11]: the subverting of even a single highly active participant could undermine the system. In Section 5, we study how the skewed structure of participation rates in PhishTank could cause trouble.

Figure 2 (top left) plots the complementary cumulative distribution function (CDF) of user submissions. Both axes are logarithmic in scale. Figure 2 (bottom left) plots the CDF for the number of votes. Power-law distributions appear as a straight line on log-log axes, so visual inspection suggests that PhishTank data is likely to be distributed in this way. We have examined the tails of the voting and submission distributions to determine whether the data are consistent with a power-law tail.

The CDF for a power-law distribution is given by:

$$Pr(X > x) = \left(\frac{x}{x_{\min}}\right)^{-\alpha+1}$$

For the submission data, we tested the tail by considering only those users who submit at least $x_{\min} = 60$ times, while we set $x_{\min} = 30$ for the voting data. We estimated the best fit for α using maximum-likelihood estimation. We then evaluated the fit by computing the Kolmogorov-Smirnov test. The results are given in the following table:

	Power-law distribution		Kolmogorov-Smirnov	
	α	$x_{\min}$	D	p-value
Submissions	1.642	60	0.0533	0.9833
Votes	1.646	30	0.0368	0.7608

Given the large p-values and small D values from the Kolmogorov-Smirnov test, we can say with high confidence that both the submission and voting data are consistent with a power-law distribution. Figure 2 (top and bottom right) presents the CDF for the tails of the submission and voting data, respectively, along with a line showing the power-law fit.

2.3 Duplicate submissions in PhishTank

PhishTank asks its users to vote on every unique URL that is submitted. Unfortunately, this imposes a very large and unnecessary burden on its volunteers. The 'rock-phish' gang is a group of criminals who perpetrate phishing attacks on a massive scale [14]. Instead of compromising machines for hosting fake HTML in an ad-hoc manner, the gang first purchases a number of domains with meaningless names like `lof80.info`. They then send email spam with a long URL of the form `http://www.bank.com.id123.lof80.info/vr`. This URL includes a unique identifier; all variants are resolved to a particular IP address using 'wildcard DNS'. Up to 25 banks are impersonated within each domain. For a more complete description of rock-phish attacks see [15].

Transmitting unique URLs trips up spam filters looking for repeated links, and also fools collators like PhishTank into recording duplicate entries. Consequently, voting on rock-phish attacks becomes very repetitive. We observed 3 260 unique rock-phish domains in PhishTank. These domains appeared in 120 662 submissions, 60% of the overall total. Furthermore, 893 users voted a total of 550 851 times on these domains! This is a dreadfully inefficient allocation of user resources, which could instead be directed to speeding up verification times, for example.

Further duplication must also be addressed in the remaining 80 246 submissions. In many instances several URLs have been submitted that correspond to webpages from different stages within the same phishing attack. By ignoring any part of the URL following the right-most `/`, we arrive at 75 501 unique URLs. Of course, there may be a very small number of cases where this consolidation treats multiple distinct phishing websites as one. However, the benefits in reducing workload seem to outweigh this unlikely occurrence.

3 Comparing open and closed phishing feeds

PhishTank is not the only organization tracking and classifying phishing websites. Other organizations do not follow PhishTank's open submission and verification policy; instead, they gather their own proprietary lists of suspicious websites and employees determine whether they are indeed phishing. We have obtained a feed from one such company. In this section, we examine the feeds of PhishTank and the company to compare completeness and speed of verification.

3.1 Phishing website identification

We compared the feeds during a 4-week period in July and August 2007. We first examine ordinary phishing websites, excluding rock-phish URLs. PhishTank reported 10 924 phishing URLs, while the company identified 13 318. After removing duplicates, the numbers become much closer: 8 296 for PhishTank and 8 730 for the company. The two feeds shared 5 711 reports in common. This means that 3 019 reports were unique to the company's feed, while 2 585 reports only appeared in PhishTank. Hence, although neither feed is comprehensive, the company's feed contains a wider selection of websites than PhishTank achieves.

For rock-phish URLs the difference is starker. PhishTank identified 586 rock-phish domains during the sample, while the company detected 1 003, nearly twice as many. Furthermore, the company picked up on 459, or 78%, of the rock-phish domains found in PhishTank, and detected 544 that PhishTank had missed.

By examining the overlap between the feeds, we can gain some insight into the company's sources. The overlap for all phishing reports corresponded to 9 380 submissions to PhishTank. 5 881 of these submissions, 63% of the total overlap, came from a user called *PhishReporter*, that we understand to be an anti-phishing report collation organization in its own right. This certainly implies that the company and PhishTank both receive a feed from *PhishReporter*. However, the remaining reports are more widely distributed, coming from 316 users. Unfortunately, we cannot say with any certainty whether these reports were also given to the company or if they were independently rediscovered.

It is noteworthy that both feeds include many phishing websites which do not appear on the other. This observation motivates the case for a universal feed shared between the banks and the various anti-phishing organizations.

3.2 Phishing website verification

Given that prompt identification and removal of phishing websites is a priority, a feed's relevance depends upon the speed with which websites are reported and subsequently verified. Requiring several users to vote introduces significant delays. On average, PhishTank submissions take approximately 46 hours to be verified. A few instances take a very long time to be verified, which skews the average. The median, by contrast, is around 15 hours.

We also found that unanimous votes were verified slightly quicker than votes where there was disagreement on average, but that conflicting votes had a much shorter median (7 hrs). URLs confirmed to be phishing were verified a few hours faster than those determined not to be a phishing website. The precise values are given in the following table:

Verification time	All entries	Conflict	Unanimous	Is-phish	Not-phish
Mean (hours)	45.6	49.7	45.8	46.1	39.5
Median (hours)	14.9	6.6	27.8	14.1	20.6

We also compared the submission and verification times for both feeds during the four-week sample. On average, PhishTank saw submissions first, by around 11 minutes, but after an average delay of just 8 seconds the company had verified them.[2] However, PhishTank's voting-based verification meant that they did not verify the URLs (and therefore did not disseminate them) until 16 hours later. For the rock-phish URLs, we compared the earliest instance of each domain, finding that overlapping domains appeared in PhishTank's feed 12 hours *after* they appeared in the company's feed, and were not verified for another 12 hours. The time differences between feeds are summarized in the following table:

ΔPhishTank	Ordinary phishing URLs		Rock-phish domains	
− Company	Submission	Verification	Submission	Verification
Mean (hrs)	−0.188	15.9	12.4	24.7
Median (hrs)	−0.0481	10.9	9.37	20.8

To sum up, voting-based verification introduces a substantial delay when compared to a unilateral verification.

4 Testing the accuracy of PhishTank's crowd decisions

Having compared the breadth and timeliness of PhishTank's reports to the closed source, we now examine the correctness of its users' contributions. Unfortunately, since the closed phishing feed does not provide a record of invalid submissions, we cannot compare its accuracy to PhishTank's. We first describe common causes of inaccuracy and discuss their prevalence. We then demonstrate that inexperienced users are far more likely to make mistakes than experienced ones. Finally, we show that users with bad voting records 'cluster' by often voting together.

4.1 Miscategorization in PhishTank

The vast majority of user submissions to PhishTank are indeed phishing URLs. Of 176 654 verified submissions, just 5 295, or 3%, are voted down as invalid. Most of these invalid submissions appear to be honest mistakes. Users who do not understand the definition of phishing submit URLs from their spam, while others add URLs for other types of malicious websites, such as those involved in

[2] We suspect that verification of any particular URL is in the hands of an individual on-duty employee, who often submits and verifies in a single operation.

advanced fee fraud (419 scams). However, a number of carefully-crafted phishing websites have also been miscategorized and 'foreign-language' websites are sometimes classified incorrectly. Most commonly, an obscure credit union or bank that uses a different domain name for its online banking may be marked as a phish.

Yet there is even dissent among moderators as to what exactly constitutes a phish: 1.2% of their submissions are voted down as invalid. For example, some moderators take the view that so-called 'mule-recruitment' websites should be categorized phishing because they are used to recruit the gullible to launder the proceeds of phishing crime. Other mistakes may just be the result of fatigue, given that the moderators participate many thousands of times.

In addition to invalid submissions that are correctly voted down, submissions that are incorrectly classified present a significant worry. Identifying false positives and negatives is hard because PhishTank rewrites history without keeping any public record of changes. As soon as a submission has received enough votes to be verified, PhishTank publishes the decision. Sometimes, though, this decision is reversed if someone disputes the conclusion. In these cases, voting is restarted and the new decision eventually replaces the old one. Once we realized this was happening, we began rechecking all PhishTank records periodically for reversals. In all, we identified 42 reversals. We found 39 false positives – legitimate websites incorrectly classified as phishing – and 3 false negatives – phishing websites incorrectly classified as legitimate. 12 of these reversals were initially agreed upon unanimously!

We first discuss the false positives. 30 websites were legitimate banks, while the remaining 9 were other scams miscategorized as phishing. Sometimes these were legitimate companies using secondary domains or IP addresses in the URLs, which confused PhishTank's users for a time. However, several popular websites' primary domains were also voted as phish, including eBay (`ebay.com`, `ebay.de`), Fifth Third Bank (`53.com`) and National City (`nationalcity.com`). Minimizing these types of false positives is essential for PhishTank because even a small number of false categorizations could undermine its credibility.

Unsurprisingly, there are many more false positives than false negatives since the vast majority of submitted phishes are valid. However, we still observed 3 false negatives. Most noteworthy was incorrectly classifying as innocuous a URL for the rock-phish domain `eportid.ph`. Five other URLs for the same domain were submitted to PhishTank prior to the false negative, with each correctly identified as a phish. So in addition to the inefficiencies described in Section 2.3, requiring users to vote for the same rock-phish domain many times has enabled at least one rock-phish URL to earn PhishTank's (temporary) approval.

4.2 Does experience improve user accuracy?

Where do these mistakes come from? It is reasonable to expect occasional users to commit more errors than those who contribute more often. Indeed, we find strong evidence for this in the data. The left-hand graph in Figure 3 plots the rates of inaccuracy for submissions and votes grouped by user participation rates. For instance, 44% of URLs from users who submit just once are voted down as

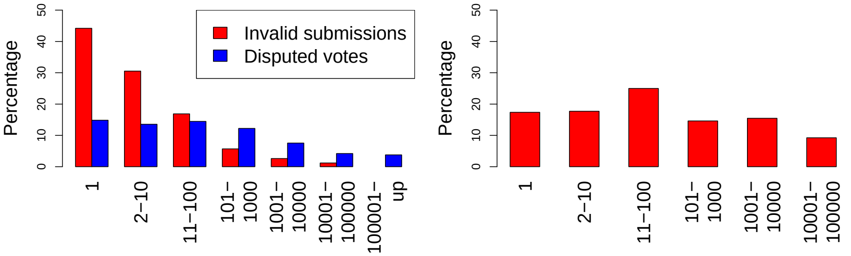

Fig. 3. Inaccuracy of user submissions and votes according to the total number of submissions and votes per user, respectively (left). Proportion of all invalid user submissions grouped by number of submissions (right).

invalid. This steadily improves (30% of submissions are invalid from users who submit between 2 and 10 URLs, 17% invalid for users with between 11 and 100 submissions), with the top submitters incorrect just 1.2% of the time.

A similar, albeit less drastic, difference can be observed for voting accuracy. Unfortunately, we cannot determine with certainty whether a user has voted incorrectly (i.e., voted a submission as a phish when the majority said otherwise, or vice versa). This is because PhishTank does not publicly disclose this information. So we are left to devise a proxy for incorrectness using votes where there is disagreement (i.e., a mixture of yes/no votes). This is a reasonable approximation given that nearly all submissions (97%) are decided unanimously.

Users voting fewer than 100 times are likely to disagree with their peers 14% of the time. This improves steadily for more active users, with the most active voters in conflict just 3.7% of the time, in line with the overall average.

These results suggest that the views of inexperienced users should perhaps be assigned less weight when compared to highly experienced users.[3] However, we must note that simply ignoring low-contribution users would not eradicate invalid submissions and votes. Since most contributions come from experienced users, many of the errors can be traced to them as well. The right-hand graph in Figure 3 groups user submissions together logarithmically, then plots the proportion of all invalid user submissions each group contributes. For instance, users submitting once contribute 17% of all invalid submissions. Users with fewer than 100 submissions collectively make 60% of the mistakes, despite submitting less than 7% of the phishing candidate URLs.

4.3 Do users with bad voting records vote together?

We now consider whether bad decisions reinforce themselves. More precisely, we ask whether users with bad voting records are likely to vote on the same phishing reports more often than randomly.

[3] Developers at PhishTank tell us that they have never treated users equally, but weigh their votes according to the user's accuracy over time.

We define a *high-conflict user* as one where a large fraction of votes f_{HC} cast are in conflict. We denote the set of all high-conflict users as HC, and the set of votes for user A as V_A. T is the set of all phishing submissions, and $V_A \subset T$.

We denote high-conflict users as those where the majority of their votes are in conflict ($f_{HC} = 0.5$). Of 3 786 users, 93 are in high conflict. We now explore the relationship between these users.

We can empirically measure the observed overlap between high-conflict votes using the following formula:

$$\text{overlap}(HC) = \sum_{A \in HC} \sum_{B \in HC, B \neq A} |V_A \cap V_B|$$

If there is no relationship between the users, then we would expect their interactions to be random chance. Hence, we can develop a measure of the expected overlap[4] in this case:

$$E(\text{overlap}) = \sum_{A \in HC} \sum_{B \in HC, B \neq A} \sum_{i=1}^{min(|V_A|,|V_B|)} i \times \frac{\binom{|V_A|}{i} \times \binom{|T|-|V_A|}{|V_B|-i}}{\binom{|T|}{|V_B|}}$$

If the overlap observed, overlap(HC), is greater than the overlap expected, E(overlap), then the high-conflict voters have tended to vote with each other more often than randomly. In our data, overlap(HC) $= 254$, while the expected overlap, E(overlap) $= 0.225$.

In other words, the rate of overlap in high-conflict voters is approximately one thousand times higher than would be the case if there was no connection between how high-conflict voters select their votes.

What are the implications? While it is possible that these high-conflict users are deliberately voting incorrectly together (or are the same person!), the more likely explanation is that incorrect decisions reinforce each other. When well-intentioned users vote incorrectly, they have apparently made the same mistakes.

5 Disrupting the PhishTank verification system

We now consider whether PhishTank's open submission and voting policies may be exploited by attackers. Recently, a number of anti-phishing websites were targeted by a denial-of-service attack, severely hindering their work in removing malicious sites [12]. Hence, there is already evidence that phishermen are motivated to disrupt the operations of groups like PhishTank. But even if enough bandwidth is provisioned to counter these attacks, PhishTank remains susceptible to vote rigging that could undermine its credibility. Any crowd-based decision mechanism is susceptible to manipulation. However, as we will see, certain characteristics of user participation make PhishTank particularly vulnerable.

[4] We are grateful to Jaeyeon Jung for correcting an earlier version of this formula.

5.1 Attacks and countermeasures

We anticipate three types of attacks on PhishTank:

1. Submitting invalid reports accusing legitimate websites.
2. Voting legitimate websites as phish.
3. Voting illegitimate websites as not-phish.

We can envision two scenarios where an attacker tries to manipulate PhishTank. The *selfish attacker* seeks to protect her own phishing websites by voting down any accusatory report as invalid. Such an attacker shares no empathy with other phishing attackers. The selfish attacker attempts to avoid unwanted attention by only allowing a few of her own websites through (attack type 3 above). The attacker's strong incentive to protect herself even when it causes harm to others is a novel property of PhishTank's voting system.

The *undermining attacker* does not bother with such subtleties. Instead, this attacker seeks to harm the credibility of PhishTank, which is best achieved by combining attacks 1 and 2: submitting URLs for legitimate websites and promptly voting them to be phish. This attacker may also increase the confusion by attempting to create false negatives, voting phishing websites as legitimate.

Detecting and defending against these attacks while maintaining an open submission and verification policy is hard. Many of the straightforward countermeasures can be sidestepped by a smart attacker. We consider a number of countermeasures in turn, demonstrating their inadequacy.

One simple countermeasure is to place an upper limit on the number of actions any user can take. This is unworkable for PhishTank due to its power-law distribution: some legitimate users participate many thousands of times. In any case, an enforced even distribution is easily defeated by a Sybil attack [7], where users register many identities. Given that many phishing attackers use botnets, even strict enforcement of 'one person, one vote' can probably be overcome.

The next obvious countermeasure is to impose voting requirements. For example, a user must have participated 'correctly' n times before weighing their opinion. This is ineffective for PhishTank, though the developers tell us that they do implement this countermeasure. First, since 97% of all submissions are valid, an attacker can quickly boost her reputation by voting for a phish slightly more than n times. Second, a savvy attacker can even minimize her implication of real phishing websites by only voting for rock-phish domains or duplicate URLs. Indeed the highly stylized format for rock-phish URLs makes it easy to automate correct voting at almost any desired scale.

Let us consider the complementary countermeasure. What about ignoring any user with more than n invalid submissions or incorrect votes? The idea here is that a malicious user is unlikely to force through all of his bad submissions and votes. Hence, a large number of deviating actions is a good proxy of misbehavior. Unfortunately, the power-law distribution of user participation causes another problem. Many heavily participating users who do a lot of good also make a lot of mistakes. For instance, the top submitter, *antiphishing*, is also the user with the highest number of invalid submissions, 578.

An improvement is to ban users who are wrong more than $x\%$ of the time. Nevertheless, attackers can simply pad their statistics by voting randomly, or by voting for duplicates and rock-phish URLs. Furthermore, many well-intentioned users might be excluded. Ignoring all users where more than 5% of their submissions are invalid would exclude 1 343 users, or 44% of all submitters. Ignoring them would also exclude 8 433 valid submissions, or 5% of all phishing URLs.

Moderators already participate in nearly every vote, so it would not be a stretch to insist that they were the submitter or voted with the majority. We do not know how often they vote incorrectly, but as discussed in Section 4.1, we know that even moderators make mistakes. One sign of fallibility is that just over 1% of moderator's submissions were voted down as invalid. Nonetheless, perhaps the best strategy for PhishTank is to use trusted moderators exclusively if they suspect they are under attack. Given that the 25 moderators already cast 74% of PhishTank's votes, silencing the whole crowd to root out the attackers may sometimes be wise, even if it contradicts principles of open participation.

5.2 Lessons for secure crowd-sourcing

We can draw several general lessons about applying the open-participation model to security tools after examining the PhishTank data.

Lesson 1: The distribution of user participation matters. There is a natural tendency for highly skewed distributions, even power laws, in user participation rates. Power law-like distributions have also been observed in the interactions of online communities [27] and blogs [21]. While there may certainly be cases that are not as skewed as PhishTank, security engineers must check the distribution for wide variance when assessing the risk of leveraging user participation.

Skewed distributions can indeed create security problems. First, corruption of a few high-value participants can completely undermine the system. This is not a huge threat for PhishTank since attackers are probably too disorganized to buy off moderators. Nonetheless, the power-law distribution still means that the system could be in trouble if a highly active user stops participating.

Second, because good users can participate extensively, bad users can too. Simple rate-limiting countermeasures do not work here. Bad users may cause significant disruption under cover of a large body of innocuous behavior. Note that we do not take the view that all crowd-based security mechanisms should have balanced user participation. Enthusiastic users should be allowed to participate more, since their enthusiasm drives the success of crowd-based approaches. However, the distribution must be treated as a security consideration.

Lesson 2: Crowd-sourced decisions should be difficult to guess. Any decision that can be reliably guessed can be automated and exploited by an attacker. The underlying accuracy of PhishTank's raw data (97% phish) makes it easy for an attacker to improve her reputation by blindly voting all submissions as phish.

Lesson 3: Do not make users work harder than necessary. Requiring users to vote multiple times for duplicate URLs and rock-phish domains is not only an efficiency issue. It becomes a security liability since it allows an attacker to build up reputation without making a positive contribution.

6 Related work

In earlier work, we have estimated the number and lifetimes of phishing websites using data from PhishTank [15] and demonstrated that timely removal reduced user exposure. Weaver and Collins computed the overlap between another two phishing feeds and applied capture-recapture analysis to estimate the number of overall phishing attacks [26].

In his book ‘The Wisdom of Crowds’, Surowiecki argued that under many circumstances the aggregation of a group’s opinions can be more accurate than even the most expert individual [25]. He noted that web participation is particularly suited to crowd-based aggregation. Surowiecki listed a number of conditions where crowd-based intelligence may run into trouble: from overly homogeneous opinions to imitative users. We have highlighted how crowds may be manipulated if the distribution of participation is highly skewed and the correct decision can be reliably guessed.

Recently, user participation has been incorporated into security mechanisms, primarily as a data source rather than performing assessment as is done by PhishTank. Microsoft Internet Explorer and Mozilla Firefox both ask users to report suspicious websites, which are then aggregated to populate blacklists. ‘StopBadware’ collects reports from users describing malware and disseminates them after administrators have examined the submissions [23]. Herdict is a software tool which collects data from client machines to track malware [8]. Vipul’s Razor, an open-source spam-filtering algorithm used by Cloudmark, solicits user spam emails as input [24]. NetTrust is a software application that shares information about the trustworthiness of websites via social networks [5].

Researchers have observed skewed distribution of user activity on the web in contexts other than security. Shirky argued that the influence of blogs (measured by the number of inbound links) naturally exhibited power-law distributions and discussed concerns about the effects of such inequality [22]. Adar et al. studied the structure of links between blogs to develop a ranking mechanism [1], while Shi et al. found blogs exhibited near-power-law distributions [21] in the number of inbound links. Meanwhile, Zhang et al. found power-law distributions in the participation rates of users in online communities [27].

Concerns over the manipulability of user-contributed web content have been raised before, most notably in the case of Wikipedia [6]. The SETI@Home distributed computational project was reported to have experienced widespread cheating [9]. More generally, Albert et al. found that networks whose connections between nodes are distributed according to a power law are vulnerable to targeted removal [2].

Countermeasures to voting manipulation where some users vote’s are weighed more heavily than others share similarities to research in trust management [4]. Researchers have devised many different metrics which differentiate between good users and bad [19,13], often for use in reputation systems [10]. More sophisticated trust metrics like these might fare better than the simple countermeasures discussed in Section 5.1.

7 Conclusion

End-user participation is an increasingly popular resource for carrying out information security tasks. Having examined one such effort to gather and disseminate phishing information, we conclude that while such open approaches are promising, they are currently less effective overall than the more traditional closed methods. Compared to a data feed collected in a conventional manner, PhishTank is less complete and less timely. On the positive side, PhishTank's decisions appear mostly accurate: we identified only a few incorrect decisions, all of which were later reversed. However, we found that inexperienced users make many mistakes and that users with bad voting records tend to commit the same errors. So the 'wisdom' of crowds sometimes shades into folly.

We also found that user participation varies greatly, raising concerns about the ongoing reliability of PhishTank's decisions due to the risk of manipulation by small numbers of people. We have described how PhishTank can be undermined by a phishing attacker bent on corrupting its classifications, and furthermore how the power-law distribution of user participation simultaneously makes attacks easier to carry out and harder to defend against.

Despite these problems, we do not advocate against leveraging user participation in the design of all security mechanisms. Rather, we believe that the circumstances must be more carefully examined for each application, and furthermore that threat models must address the potential for manipulation.

Acknowledgments

Tyler Moore is supported by the UK Marshall Aid Commemoration Commission and by US National Science Foundation grant DGE-0636782. Richard Clayton is currently working on the spamHINTS project, funded by Intel Research.

References

1. Adar, E., Zhang, L., Adamic, L., Lukose, R.: Implicit structure and the dynamics of blogspace. In: Workshop on the Weblogging Ecosystem, 13th International World Wide Web Conference (WWW) (2004)
2. Albert, R., Jeong, H., Barabási, A.: Error and attack tolerance of complex networks. In: Nature 406, 378–382 (2000)
3. Anderson, L., and Holt, C. Information cascades in the laboratory. In: American Economic Review 87(5), 847–862 (1995)
4. Blaze, M., Feigenbaum, J., Lacy, J: Decentralized trust management. In: IEEE Symposium on Security and Privacy (S&P), pp. 164–173. IEEE Computer Society, Los Alamitos (1996)
5. Camp, L. J.: Reliable, usable signaling to defeat masquerade attacks. In: Fifth Workshop on the Economics of Information Security (WEIS) (2006)
6. Denning, P., Horning, J., Parnas, D., and Weinstein, L. 2005. Wikipedia risks. In: Communications of the ACM 48(12), 152–152 (2005)

7. Douceur, J.R.: The Sybil attack. In: Peer-to-Peer Systems, 1st International Workshop, Revised Papers. Lecture Notes in Computer Science (LNCS), vol. 2429. Springer (2002) 251–260
8. Hwang, T.: Herdict: a distributed model for threats online. In Bradbury, D. (ed.): Network Security, pp. 15–18. Elsevier, Oxford (August 2007)
9. Kahney, L.: Cheaters bow to peer pressure. Wired, 15 Feb 2001. http://www.wired.com/news/technology/0,1282,41838,00.html
10. Kamvar, S., Schlosser, M., Garcia-Molina, H.: The EigenTrust algorithm for reputation management in P2P networks. In: 12th WWW, pp. 640–651. ACM Press, New York (2003)
11. Lamport, L., Shostak, R., Pease, M.: The Byzantine Generals Problem. In: ACM Transactions on Programming Languages and Systems 4(3), 382–401 (1982)
12. Larkin, E.: Online thugs assault sites that specialize in security help. PC World, 11 Sep 2007. http://www.pcworld.com/businesscenter/article/137084/online_thugs_assault_sites_that_specialize_in_security_help_.html
13. Levien, R.: Attack resistant trust metrics. PhD thesis (draft), University of California at Berkeley (2004)
14. McMillan, R.: 'Rock Phish' blamed for surge in phishing. InfoWorld, 12 Dec 2006. http://www.infoworld.com/article/06/12/12/HNrockphish_1.html
15. Moore, T., Clayton, R.: Examining the impact of website take-down on phishing. In: Anti-Phishing Working Group eCrime Researcher's Summit (APWG eCrime), pp. 1–13. ACM Press, New York (2007)
16. Newman, M.: Power laws, Pareto distributions and Zipf's law. In: Contemporary Physics 46(5), 323–351 (2005)
17. Pan, Y., Ding, X.: Anomaly based web phishing page detection. In: 22nd Annual Computer Security Applications Conference (ACSAC'06), pp. 381–392. IEEE Computer Society, Los Alamitos (2006)
18. PhishTank: http://www.phishtank.com/
19. Reiter, M., Stubblebine, S.: Toward acceptable metrics of authentication. In: IEEE S&P, pp. 10–20. IEEE Computer Society, Los Alamitos (1997)
20. Ross, B., Jackson, C., Miyake, N., Boneh, D., Mitchell, J.: Stronger password authentication using browser extensions. In: 14th USENIX Security Symposium, p. 2. USENIX Association, Berkeley (2005)
21. Shi, X., Tseng, B., Adamic, L.: Looking at the blogosphere topology through different lenses. In: International Conference on Weblogs and Social Media (2007)
22. Shirky, C.: Power laws, weblogs, and inequality. http://www.shirky.com/writings/powerlaw_weblog.html (2003)
23. Stop Badware: http://www.stopbadware.org
24. Vipul's Razor: http://razor.sourceforge.net
25. Surowiecki, J.: The wisdom of crowds: why the many are smarter than the few. Doubleday, New York (2004)
26. Weaver, R., Collins, M.: Fishing for phishes: applying capture-recapture to phishing. In: APWG eCrime, pp. 14–25. ACM Press, New York (2007)
27. Zhang, J., Ackerman, M., Adamic, L.: Expertise networks in online communities: structure and algorithms. In: 16th WWW, pp. 221–230. ACM Press, New York (2007)
28. Zhang, Y., Egelman, S., Cranor, L., Hong, J.: Phinding phish: evaluating anti-phishing tools. In: 14th Annual Network & Distributed System Security Symposium (NDSS 2007) (2007)