How to buy better testing

Using competition to get the most security and robustness for your dollar

Stuart Schechter

Harvard University
stuart@post.harvard.edu

Abstract. Without good testing, systems cannot be made secure or robust. Without metrics for the quality and security of system components, no guarantees can be made about the systems they are used to construct. This paper describes how firms can make the testing process faster and more cost effective while simultaneously providing a reliable metric of quality as one of the outputs of the process. This is accomplished via a market for defect reports, in which testers maximize profits by minimizing the cost of finding defects. The power of competition is harnessed to ensure that testers are paid a fair price for the defects they discover, thereby aligning their incentives with those of the firm developing the system. The price to find, demonstrate, and report a defect that is set by the market serves as the measure of quality.

1 Introduction

A market for lemons is one in which consumers cannot determine products of quality from defective goods. In such a market, consumers assume they will be sold a product of the lowest quality and so become unwilling to pay a price higher than they would for such a 'lemon.' As a result, it becomes economical to produce only products of the lowest quality [1]. Anderson laments that security is a trait that consumers cannot measure and as a result firms have had little incentive to produce more secure systems, so that from the standpoint of security, systems tend to be lemons [2].

This paper proposes a means to make the process of testing systems faster and more cost effective while integrating into the process a reliable metric of the quality of the tested system. This metric of quality, which can be measured throughout the testing process, is the market price to find, demonstrate, and report a previously undetected defect in the system. Previously referred to as the *cost to break* (CTB) of a system [3, 4], this metric is impractical if not impossible to calculate using technical means but can be measured using a market for defect reports. In this market a firm buys defect reports from testers, at a market price governed by the presence of competition among those testers.

Comprehensive testing is an essential ingredient to the process of creating secure and robust systems and components. While testing provides a means of evaluating the progress of the development process, it is also important to be able to evaluate the integrity of the testing process itself. Measuring how well

a system has been tested is essential for determining when a system is to be released and for comparing the system to other systems.

However, the industry has struggled to quantify security, robustness, and other key measures of quality. Existing metrics, such as as Mean Time Between Failures (MBTF), fail to measure how systems respond to extreme conditions and the presence of adversaries. The industry also has yet to discover how to optimize the incentives of developers and testers to ensure that systems are well designed, built, and tested. Firms need a way to ensure that systems have reached a quantifiable level of security and robustness and that they are getting value for their testing dollar. Until this goal can be achieved, firms will overspend on testing while continuing to release bug-ridden systems.

Having a metric of security and robustness as an output of the testing process has further benefits. Consider that a system is only as secure and robust as the weakest of the hardware and software subsystems from which it is built. For example, a system running the world's most secure web server is far from secure if it is running the world's least secure operating system. Before building a system with a desired level of security, it is extremely beneficial to be able to measure the strength of each potential subsystem.

This paper describes desirable properties of a market for purchasing defect reports in Section 2, with rules for the market introduced in Section 3. A number of simplifications are introduced at the beginning of the analysis in Section 4 and are later discussed and revised in Section 5. Applications are discussed in Section 6. Future work and concluding remarks are presented in Section 7 and 8, respectively.

2 What Makes a Good Market?

Given the opportunity, what properties would a firm build into an ideal market for purchasing defect reports from testers?

VALUE *The price paid for each defect found should be minimized.*

If the testing budget is fixed, *value* is obtained by maximizing the number of defects found. If the number of defects found is fixed, *value* is obtained by minimizing the amount of money spent in finding them.

SPEED *Testers should find and report defects to the manufacturer as quickly as possible.*

The sooner a defect is reported, the sooner it can be fixed and the less likely it is to cause damage before it is repaired. Given trade-offs between *value* and *speed*, this paper will introduce solutions that place priority on *value* and then show opportunities for exchanging *value* for *speed*.

ORDER *The easier a defect is to find (or the cheaper the defect report is to produce), the earlier it should be reported.*

The most obvious and most commonly occurring defects are both the most easy to find and the most likely to result in damage. Defects that are difficult to

find are the least likely to occur when the system is operating and thus are the least likely to cause damage [5]. Product quality can be improved more quickly if the most common defects are discovered and fixed first. For any class of security vulnerabilities that cause the same amount of damage when exploited, locating and fixing the cheapest vulnerabilities to find will increase the *cost to break* of the system faster than if harder to find vulnerabilities are given higher priority.

For example, a defect in a router that corrupts one of every thousand packets is of more concern than a defect that corrupts one packet per billion. A software company prevents more theft if adversaries must spend millions of dollars to break a cryptographic algorithm than if a vulnerability remains that can be found and exploited for a thousand dollars. That the most frequently occurring problems are often easy to detect is one of the few natural laws that help the system's developer.

Before a product's release, order is the least important of these properties.

3 Designing a Market

A number of papers have already made a great deal of headway into means of classifying defects and vulnerabilities [6–8]. Rather than worry about the different consequences of each class of defect, we start by assuming all defects are equally hazardous. Section 5.6 describes how different defect classes are handled.

We design a market for defects in which there is one buyer, the firm charged with improving the quality of the system, and many sellers, the testers charged with finding and reporting defects. This one-buyer assumption may seem misguided, especially if the defects are security vulnerabilities that, if left unrepaired, would be of value to an adversary. In Section 5.7 we will see why an adversary is not likely to buy information about a vulnerability at any price that the firm is willing to pay, and thus need not be treated as a competing buyer.

With this in mind, the firm may set the rules of the market as follows:

Access *All testers have full and equal access to the system to be tested.*

The average price of a defect report cannot fall below the average cost of finding a defect, otherwise no rational tester will search for defects. To maximize *value* and *speed* it is in the firm's best interest to minimize the testers' costs by giving them full and complete access to the system.

Pricing *A tester reporting a unique and verifiable defect at time t, complete with test case, will receive a reward $r(t)$. A tester reporting a non-unique defect (one that was reported earlier) receives nothing.*

The firm chooses any reward function $r(t)$ such that it increases continuously with time, starting with the lower bound on the cost of finding and reporting a defect and ending with CTB, the *cost to break* of the system that represents the firm's maximum willingness to pay for a defect.

This rule frees the firm from any need to understand the testers' costs in order to set prices and maximize *value*. Rather, the firm will rely upon competition between the testers to minimize its costs.

While not essential to this analysis, I suggest supplementing the pricing rule to require testers to pay the transaction costs of processing defect reports (still receiving their reward if the report is valid). This discourages reports that are erroneous, badly detailed, or duplicates of published existing reports. Testers are expected to integrate these costs into the price they demand for each defect reported.

Selling *A tester may search for defects at any time, and may report a defect immediately or at any time after finding it.*

While forcing immediate reporting would appear to maximize *speed* and improve *order*, it would be impossible to do so. More importantly, doing so would interfere with the testers' ability to make a fair profit and reduce the incentive to test. We will instead rely (yet again) on competition to achieve our goals.

Information *All testers receive the other testers' defect reports immediately after they are received by the firm.*

After we've ensured that testers receive the maximum information available about the system (via the Access rule), testers' efficiency can be further improved by ensuring that they are not wasting effort searching for known defects. The information rule stated above is most sensible before a product's final release, when defects are common and there is a reasonable probability that additional testers would discover a previously reported defect before the firm can fix it. After product release, a firm may want to sacrifice *value* by delaying the release of security defects until after they are repaired in order to ensure that users have a chance to patch their systems before the vulnerability is made publicly available. This will be discussed in more detail in Section 5.8.

4 Simplifying Assumptions

The market for defects has been constructed so that once the the firm has declared its desired *cost to break*, CTB, it has no further strategic choices to make. The strategy lies completely in the hands of the testers.

In order to analyze how the testers will behave, we first simplify the nature and rules of the game until the analysis becomes trivial and then remove the simplifications in order to make the game better reflect reality.

The following story summarizes our simplified game.

> Frank's system has a defect which he is willing to pay $1,000 to locate. Frank invites Ann and Bob to test the system over a thousand hour period and to provide a test case that identifies the defect. The reward for reporting the defect in the first hour is a dollar. As each hour passes, the reward for reporting the defect is increased by a dollar. However, only the first player to describe the defect will receive the reward.

Ann can find the defect with $300 of effort. Bob can find the defect with $500 of effort. No other tester can find the defect for less than $500 of effort. All of these facts are known to all the testers.

The simplifications implied in the above example can be formalized in the following statements:

(1) *There is one and only one defect.*

(2) *The act of finding a defect is atomic and takes no time.*

(3) *Each tester knows her cost of finding the defect.*

(4) *Each tester knows the other testers' cost of finding the defect. Each tester knows that the other testers know each other's cost of finding a defect, and so on.*

If each player knows that a defect exists and knows everyone's cost of finding the defect (perfect knowledge as defined in simplifying assumptions 3 and 4), the market can be modelled as a cooperative game with multiple sellers (the testers) and one buyer (the firm). All testers are offering an information good that is a perfect substitute for the others' good as all are describing the same defect. The firm is only willing to pay for this information from one of the testers.

The marginal contribution of all testers except for the one with the lowest cost is 0. The marginal contribution of the low cost tester is the difference between her cost of production (finding the defect) and that of the tester with the second lowest cost of production. The tester cannot demand a price higher than the cost of production of the second lowest cost tester. Since the firm does not have the ability to negotiate in this market, we can assume that it will have to pay the highest price that is less than the cost of the second lowest cost tester. In our story, the smallest discrete unit of currency is a dollar.

Ann believes that Bob and the other testers are rational and that they will not sell the defect at a loss. Since no other tester can find the defect for less than $500, Ann can sell the defect at any price below $500. To maximize her profit, Ann finds the defect and reports it when the reward reaches $499.

5 Approaching Reality

In the real world, products have multiple defects, finding a defect is a time consuming operation requiring a large number of subtasks, and each tester's knowledge is far from perfect. We'll now try to replace the above simplifying assumptions with ones that more closely match the realities of testing.

5.1 The presence of multiple defects

This market for defects would be of little value if it could only be used to find a single flaw in the system. This assumption must be removed, which we will represent by writing it again and crossing it out.

~~(1) *There is one and only one defect.*~~

In order for the market described to accommodate reports of multiple defects we must substitute a new assumption for assumption 1.

(1a) *All defects are independent. Finding one defect d in the set of all defects D neither aids nor hinders the search for another defect $d' \in D$, nor does it indicate that a tester is more or less likely to find another defect.*

With this new assumption in place, we can open markets for any number of defects in parallel, even if we don't know how many defects there are. It is easiest to view these parallel markets as one single large market for defects.

> There are now two defects d_1 and d_2, each of which Frank is willing to pay \$1,000 to locate. Ann's costs of finding defects d_1 and d_2 are \$300 and \$400 respectively. Bob's costs of finding defects d_1 and d_2 are \$500 and \$200 respectively. Ann reports d_1 for a reward of \$499 and Bob reports d_2 for \$399.

5.2 Knowledge about others' costs (part one)

~~(4) *Each tester knows the other testers' cost of finding the defect.*~~

It's quite unlikely that a tester can know how hard it is for every other tester to find a defect when she doesn't even know what the defect is yet. We will replace this assumption with one that is somewhat less far fetched. This will allow us to make progress in the analysis before relaxing this assumption further in Section 5.4.

(4a) *Once given knowledge of a defect d, a tester with one of the two lowest costs of finding d will know the other lowest cost tester's cost to find d.*

There are now two sets of moves for each tester. First, a tester must decide if and at what price she will search for a defect. Then, if the tester has found a defect, she must decide at what price she will sell it. She can calculate the appropriate times for these actions from her matching price choices by using the inverse function of $r(t)$.

If Ann knows her cost of finding a defect (Assumption 3), and we define c_a to be this cost, we say that Ann will find a defect when the price covers her cost; when $r(t) \geq c_a$. At this time t she can immediately sell the defect and break even[1], or wait until just before the next lowest cost tester will find the defect and sell it for a profit. She can do this because once she has found the defect she will know the next lowest cost testers's cost of finding the defect. Though the story changes slightly under these new assumptions, we always reach the same end.

> Ann and Bob's costs haven't changed, but neither player starts out with any knowledge about the other's costs. When the reward price

[1] We ignore the unlikely event that Ann's cost is exactly the same as Bob's, and that they both attempt to report the defect at the same unit of time, as this becomes increasingly unlikely as our measurement of time becomes increasingly less discrete.

reaches \$200, Bob spends \$200 to find defect d_2 knowing that he can sell it immediately to break even regardless of what he discovers Ann's cost of finding d_2 to be. After spending his \$200, Bob learns everything he needs to know about d_2, including that it would cost Ann \$400 to find d_2. Bob now knows, just as he did in the previous example, that he can wait to report the defect until the reward is \$399. Using the same logic, Ann will find defect d_1 once the reward reaches \$300 and will once again report it when the reward is \$499.

5.3 The time and cost of searching

~~(2) *The act of finding a defect is atomic and takes no time.*~~

~~(3) *Each tester knows her cost of finding the defect.*~~

Finding a defect can require both time and expense, and it is hard to determine the cost of finding something until you know what it is you are looking for. The following assumptions are much more realistic.

(2a) *A unit of work w spent searching for a defect is atomic, has a fixed cost c_w, and takes no time.*

(3a) *Each player can estimate the probability p_w that a unit of work she engages in will reveal a previously unreported defect.*

After 500 hours into testing the reward for reporting a defect has reached \$500. Ann has a test that she can design and run for \$5. The test has a 1% chance of revealing an unreported defect. Running the test has the expected value no less than $0.01 \cdot \$500 = \$5.00$. Ann decides to perform the test.

More formally, we say that a tester will search for a defect when her expected minimum reward justifies the cost of searching. The equation below represents this by stating that the expected value of searching, which is the product of the reward and the probability that searching yields a reward, must be greater than or equal to the cost of searching.

$$r(t) \cdot p_w \geq c_w$$

Simplification 2a will not be further refined in this paper, and so our analysis will continue to be predicated on the assumption that a unit of work takes no time. There is no doubt that since even small units of work take time, a trade-off exists between the *speed* and *value* in testing. The choice of testing period is one which needs to be made based on the time and budgetary constraints of firms with knowledge of the state of the art in testing tools and technology. It is thus outside the scope of this paper.

5.4 Knowledge about others' costs (part 2)

~~(4a) *Once given knowledge of a defect d, a tester with one of the two lowest costs of finding d will know the other lowest cost tester's cost to find d.*~~

Removing this assumption about knowledge makes the players' second move strategy less clear. All we can do is analyze the strategies of each player based on his or her beliefs about the costs of the other players.

Once Ann has found a defect, the crux of the problem is that she no longer knows the likelihood that another tester will report that defect at time t for reward $r(t)$.

We will represent Ann's estimate of the probability that no other player will have reported defect d at time t as $p_a(T\backslash a, D, D_r, d, t)$, where $T\backslash a$ is the set of all testers except Ann, and D_r is the set of defects that have been reported.

Ann will report the defect at a time that maximizes her expected payoff function, which is once again the product of the reward times the probability of receiving the reward:

$$r(t) \cdot p_a(T\backslash a, D, D_r, d, t)$$

The better Ann's expected payoff function approximates the knowledge of assumption 4a and the behavior of the other testers who might find the same defect, the more money Ann will be able to make. The uncertainty, however, should encourage Ann to report earlier than she might if given the perfect knowledge she had with assumption 4a.

5.5 Defect dependencies and learning about the skills of others

There is a strong case for removing simplifying assumption 1a. For starters, we wouldn't think Ann was very smart if she discovered a number of defects, watched Bob report each defect at a price below what she believed his costs were, and refused to revise her estimates of Bob's costs to find additional defects.

We also can't ignore the possibility that the discovery of one defect will simplify the discovery similar defects. What's more, it is often difficult to even determine where one defect ends and another begins.

~~(1a) *All defects are independent. Finding one defect d in the set of all defects D neither aids nor hinders the search for another defect $d' \in D$, nor does it indicate that a tester is more or less likely to find another defect.*~~

Once we acknowledge that the discovery of a defect changes every player's cost of finding another defect, a new assumption won't fix the rules of the game. Rather, it is the rules of the market and not the simplifying assumptions that need repair. The following example shows how the firm may fail to obtain *value* under the pricing rules from Section 3.

> Frank's system has three defects which are all closely related. Ann can find the first of three defects for $500, and can find each of the two remaining defects for $10 each given knowledge of the first defect. Bob

can find a defect for $2000 but given knowledge of any of the defects can find the other two for only $5. Charles can find any defect for $600, but isn't aided by information about defects that have already been discovered.

Ann finds the first defect when the reward has reached $500 and then instantly finds the two other defects. She reports all three defects at once when the price reaches $599, collecting $1,797.

The above example demonstrates two problems with the rules as they stand. Since Bob has a lower cost of finding the incremental defects and Ann is the one reporting them, the most efficient tester is not the one testing for the incremental defects. Since the reward function increases monotonically, the firm must grossly overpay for the incremental defects rather than paying no more than the cost of the second lowest cost tester. By the second lowest cost rule, Ann should collect $599 for reporting the first defect, and Bob should collect $9 for each of the additional defects, allowing the firm to pay $617 for the three defects instead of $1,797.

To fix this flaw I propose that the market be reset each time a single defect is reported, with the reward price being reset down to the transaction cost of reporting a defect ($0 in our story). If the reward is reset to $0 when Ann reports the first defect, Bob will report the next defect when the reward price reaches $9. The reward will once again be set to $0 and after it climbs to $9 again he will report the third defect. While this approach sacrifices a modest amount of *speed*, without it *value* is all but nonexistent.

The firm may want to adjust the reward function so that it grows very slowly (or remains at $0) while an outstanding defect is being fixed. This prevents a slew of related defects and test cases, which are trivial modifications of the first reported defect, to be reported at high frequency causing strain on the reporting system. Alternatively, allowing the reward to increase and a flood of related test cases to be reported may be just what is needed to build a good internal regression testing suite.

There is also a rather practical benefit to setting the reward back to $0 each time a defect is reported in that it allows for more predictable budgeting. If a dollar is added to the reward 'pot' every hour, and reporting a defect allows the tester to take the money in the pot, then the rate at which defects are reported does not affect the rate at which money flows out of the firm's coffers. The disadvantage is that fixing the budget removes any guarantees about the cost to find and report a defect (cost to break) after any given testing period. A compromise solution exists at which the firm may decide to increase the flow of money into the pot in order to cause defects to be reported more quickly.

With the new rule for the reward function in place, Ann may now wish to recalculate $p_a(T \backslash a, D, D_r, d, t)$ for her unreported defects each time a defect is reported and the reward is reset. In doing so, she may actually determine that she may want to take a loss if she has underestimated the other testers in her past calculations of p_a. However, the guiding equation behind Ann's strategy

need not change as she is still working to maximize the expected reward for each defect she has found.

5.6 Some defects are more important than others

Different classes of defects have different consequences and so the firm may be willing to pay more for some defects than others. Rather than create a market for each class of defect, it is easier to pick one class of defect as the baseline class and measure the other classes relative to the baseline class. If the firm is willing to pay only half as much for a defect of a new class as it is for the baseline class, the reward for finding a defect from that class at time t can be set to $\frac{1}{2}r(t)$. Generally, if a defect is from a class that the firm is willing to pay x times more for than it would for a defect from the baseline class, the reward for reporting such a defect should be set to $x \cdot r(t)$.

This approach is equivalent to running a parallel market for each class of defect, except that all the rewards are reset each time a defect is reported. Since defects may not be independent of defects from other classes, and finding one in one class may make it easier to find one in another class, resetting all rewards is indeed necessary.

5.7 The presence of additional buyers

We began our analysis by assuming that the firm is the only buyer of defects during the testing process. One might be concerned that adversaries may purchase security defects. Assume that, following the testing process, the firm continues to offer a reward equal to its claimed cost to break (CTB) to anyone reporting a new security vulnerability. Under what conditions will an adversary buy a vulnerability during the testing process?

A security defect has no value to an adversary if the flaw is discovered and fixed by the firm before the product is released. An adversary can buy a security vulnerability by publicly offering to purchase a vulnerability at the end of the testing period at a price just higher than the the reward offered by the firm, or $\text{CTB} + \epsilon$. Neglecting ϵ and assigning to p the probability that the defect will not be discovered by the end of the testing process, we note an adversary would not pay more than $\text{CTB} \cdot p$ for a security vulnerability during the testing process.

However, the tester selling the defect also has the option of waiting until testing is over before making the transaction, and knows the firm will always willing to pay CTB for a previously unreported vulnerability even if the adversary is not. The tester will demand a price that is greater than $\text{CTB} \cdot p$. This means that the sale is a zero sum game, between the tester selling the vulnerability and the adversary buying it, in which the player who can best estimate the true probability p wins. Since the adversary does not know anything about the vulnerability until he buys it, he will never be able to estimate p as well as the tester who discovered the vulnerability and knows how hard it was to find.

What's worse, if the tester has found multiple vulnerabilities she can maximize her profits by selling the adversary the one that is most likely to be discovered by another tester (and thus is the least valuable.) As a result we can conclude that adversaries buying security vulnerabilities will be faced with a market for lemons for the duration of the testing process, and will find it in their best interest to wait until testing is complete to make a purchase. When they finally do purchase a vulnerability, they will have to pay at least as much as CTB, the price offered by the firm.

Can the tester sell at a price below CTB and make up for the difference in volume by selling to multiple buyers? In such a case, a buyer is able to engage in arbitrage by purchasing the defect and selling it to the firm. This will result in the defect being fixed, and any exploits of the defect becoming worthless. What's more, once the first buyer has purchased the exploit he would also be able to sell the exploit to other buyers. As there would now be multiple sellers of the same information good, the price of the good would drop to zero. For both these reasons, a tester cannot expect to be able to sell a security defect more than once.

In fact, the tester's ability to resell security defect information adds an additional barrier to the sale of security defects to adversaries at *any* price. Any adversary must worry that after he has paid to learn the details of the vulnerability, the seller will resell that information to the firm (or other buyers). If the transaction protects the anonymity of the seller, the tester can claim that another tester must have discovered and reported the defect. Since the only way to ensure the testers won't resell is to take them out of the game, testers who wish to avoid sleeping with the fishes will be wary of selling security defects to anyone other than the firm.

5.8 Delaying release of discovered defects

It may not always be advantageous to release defect reports immediately to all testers. This is especially true when the test is open to the public, the product is in use by customers, and the defect affects the security of the product.

If defects are not immediately released, the firm should continue to pay only the first reporter the full reward or it should split the reward among each tester. This will cause testers to demand more for each defect, as they must bear the risk of not receiving the full reward for their efforts in searching for defects. Testers will also be wary that the firm might be tempted to create a phantom tester which it could claim reported defects before the true reporter. If the testers do not trust the firm, the firm may need to employ a trusted third party to manage the entire defect reporting and reward process. If only the first reporter receives the reward, the trusted third party may only be needed to sign and time stamp a message authentication code of each defect report submitted, thereby providing proof of who found the defect first.

The firm should never give a full reward $r(t)$ to every tester who reports a defect before the public is made aware of the defect's existence. This may seem like a good way to put *speed* before *value* and encourage as much testing as

possible, but it will more likely cause the testers to collude, sharing defects to bleed cash from the firm. This is particularly serious if adversaries pose as testers and learn about vulnerabilities by taking part in such collusion.

6 Applications

Testing processes that result in metric of quality, such as we've seen above, have applications that expand beyond improving testing. These include improving product security, better management of release processes, and improving the effectiveness of incentives for both in-house developers and outside contractors.

6.1 Differentiating system security

In previous work I proposed that security should be measured economically by the cost to break into a system and that, by introducing this metric into the marketplace, the providers of more secure software can drive out the lemons [3]. The metric fixes the information asymmetry and addresses the lemons uncertainty problem.

My previous work also shows how to measure the *cost to break*, or CTB, of the firm's system F against that of a competing system C [3]. To place an upper bound of u dollars on C's cost to break, the firm offers a reward of u dollars to the first tester to report a vulnerability in C. To place a lower bound of l dollars on F's cost to break, a reward of l dollars is offered for each and every unique vulnerability reported in F. By showing that $l > u$, the firm demonstrates that its system F is more secure than the competitor's system C.

The bounding approach is effective if the true cost to break is higher than the lower bound, but exceedingly expensive otherwise, as the firm must pay a large reward for a large number of vulnerabilities. Ideally, the firm would like to measure the cost to break at lower expense during testing and throughout the development and release process. The techniques described in this paper, when applied to a set of testers that includes the public, allow the firm to use competition to avoid overpaying for the vulnerabilities that must be repaired to reach a desired cost to break.

If the firm can assume that an adversary who wants to exploit its system has access to the same public that was invited to test, it is reasonable to expect that the adversary will have to spend at least as much as the firm did in order to find an exploit.

The *cost to break* metric also aids the construction of more secure large-scale systems built from component subsystems. Recall the example of the highly secure web server running on an insecure operating system. System builders, faced with the choice of how much to invest in the security of a component, often err by paying too much for one component and too little for another. If the security of each can be measured, then the firm can be sure to invest in the security of the weakest link in the system. For systems that are only as strong as their weakest component, such a strategy will result in the maximum increase in the security of the composite system.

6.2 Compensating white box testers and software engineers

Defect markets also can be used directly for compensating all types of test engineers. Public testing does not eliminate the need for professional internal testers, as they will be able to perform white box testing using information not available to the public to find defects at lower cost. Internal testers are also best used early on as the defect reports they produce are of higher quality and are thus cheaper for the firm to verify and process. Such costs may be a significant component of the transaction when defects are inexpensive to find.

Testing often suffers because the job is considered unrewarding. Test engineers are often paid less than developers, and their positions are less esteemed, since building systems is more highly regarded than finding problems with them. All too often, good testers are promoted out of testing. This would be less likely to happen if testers' compensation more directly matched the contribution they make to the product. Markets for defects create a meritocracy in which the best testers are rewarded in a way that makes their jobs more desirable while ineffective testers are encouraged to find more suitable positions. One might even wonder whether great testers lost to development teams will start testing systems again, in their free time, to pick up extra money.[2] Some might even learn that their true calling is to return to testing.

Testers are not the only members of the team who can be evaluated based on the cost of finding defects in systems. It has often been difficult to provide engineers with incentives for developing high quality work (rather than fast work) without an objective measure of the level of defects. The cost for a tester to find a defect when the engineer releases his work for testing provides the measure of quality that can be integrated into the engineer's incentives.

6.3 Release Management

The cost of finding a defect may be used as a measure for determining when systems are ready to ship, especially when other metrics, such as Mean Time Between Failures (MTBF), are hard to measure or not applicable. Many other metrics fail to account for extreme conditions such as extreme load or attack by an adversary with extensive knowledge of the system, that would not otherwise occur in the laboratory and aren't detectable in typical real world use.

6.4 Contracts

The question of when a software development contractor has fulfilled its obligation to provide a system of a reasonable quality is one that has been the subject of innumerable disputes. Placing a lower bound on the cost of finding a defect in the completed system may be just the metric needed to allow parties to reach agreement on quality before work begins. A wise client will pay more to ensure

[2] Developers should not be allowed to collect rewards for testing their own systems as doing so compromises their incentive to create as few defects as possible.

that the testing budget comes out of the contractor's pocket, thus ensuring that the contractor will have the maximum incentive to provide initial quality. The client should also insist that the testing itself be open to third parties that cannot be influenced by the contractor.

7 Future Work

There are many ripe areas of research to explore as the ideas I have outlined are put into practice. A starting point is to relax the assumption that work spent searching takes no time (part of Assumption 3a). One approach would be to quantify the trade-offs between the value and speed of testing, perhaps as a function of the number of available testers and size of the system to be tested. Much can also be gained by formalizing rules for determining whether a defect report is indeed valid, and to explore the role played by third parties. More light needs to be shed on how testers behave given their limited knowledge of the abilities of others. This may arise through empirical study, or by adding additional assumptions so that a Bayesian-Nash Equilibrium analysis may be performed.

8 Conclusion

The security and robustness of systems has traditionally suffered because testers are not properly rewarded. I have described an economic approach to rewarding testers that aligns their goals with those of the firm developing the system. This approach allows firms to cost-effectively test systems for individual flaws and simultaneously measure the quality of the overall system.

Using testing based on defect markets as described in this paper, subsystems can be built with measurable security and can be used to construct more complex systems which can then also then be measured.

9 Acknowledgements

This paper could not have been completed without the advice, comments, and suggestions from Adam Brandenburger, L. Jean Camp, David Molnar, David Parkes, Michael D. Smith, and Omri Traub.

This research was supported in part by grants from Compaq, HP, IBM, Intel, and Microsoft.

References

1. Akerlof, G.A.: The market for 'lemons': Quality uncertainty and the market mechanism. The Quarterly Journal of Economics **84** (1970) 488–500
2. Anderson, R.: Why information security is hard, an economic perspective. In: 17th Annual Computer Security Applications Conference. (2001)

3. Schechter, S.E.: Quantitatively differentiating system security. In: The First Workshop on Economics and Information Security. (2002)
4. Silverman, R.D.: A cost-based security analysis of symmetric and asymmetric key lengths. http://wwww.rsasecurity.com/rsalabs/bulletins/bulletin13.html (2001)
5. Brady, R.M., Anderson, R.J., Ball, R.C.: Murphy's law, the fitness of evolving species, and the limits of software reliability. http://www.ftp.cl.cam.ac.uk/ftp/users/rja14/babtr.pdf (1999)
6. Camp, L.J., Wolfram, C.: Pricing security. In: Proceedings of the CERT Information Survivability Workshop. (2000) 31–39
7. Aslam, T., Krsul, I., Spafford, E.: Use of a taxonomy of security faults. In: 19th National Information Systems Security Conference. (1996)
8. Landwehr, C.E., Bull, A.R., McDermott, J.P., Choi, W.S.: A taxonomy of computer program security flaws. ACM Computing Surveys **26** (1994) 211–254