Applying Security Lessons Learned to Human Computation Solving Systems

Dan Thomsen

This chapter looks at the security issues that arise when using human computation systems to solve problems that no one has solved before. Researchers have spent decades on computer security research and yet surprisingly the biggest factor impacting security issues remains economics. Researchers know how to build secure systems, but cannot develop high assurance software fast enough to keep up with the feature race that shapes modern IT products. Techniques that attempt to crowdsource formal verification may reduce the time it takes for formal assurance, but formal assurance of any kind adds an extra step that slows time to market. The first product to market often has tremendous payoffs in terms of capturing market share. Today the rich feature environment and integration of millions of lines of code into even a simple application have made "security" mean getting hacked less than your competitors. True security means good architecture to control, but more importantly, good architecture to understand the flow of data in a system. You cannot secure what you do not understand. When looking at using a crowd of humans to solve problems, unique security issues arise because the developers must understand how humans impact security.

Money in both the terms of development costs, and reducing the time to market to capture early market share, shaped today's security mechanisms. Human computation systems must build on those flawed mechanisms. However human computation faces unique security challenges, and there is a chance at this early stage to think deeply about these issues and get them right. However, the pessimist in me says follow the money. Money and economics will shape the security philosophies that emerge for human computation unless the groundwork for good security and good architecture gets created early.

D. Thomsen (✉)
SIFT LLC, USA
e-mail: dthomsen@sift.net

P. Michelucci (ed.), *Handbook of Human Computation*,
DOI 10.1007/978-1-4614-8806-4_71,

Driving security choices for human computation systems with economics may seem odd for an emerging science where all the successful examples run on volunteer contributions, but money defines the critical pieces of data targeted for malicious activity. Security analysis identifies these critical pieces of data as assets to be protected. The economics of computer security ensure that you cannot protect all the assets, so when you have limited funds the wisest approach suggests investing security to cover the most valuable assets. This chapter discusses the computer security basics, and then how that reflects on the assets of a human computation system and what protection they need.

Computer security consists of three aspects; confidentiality, integrity, and availability. Confidentiality means ensuring only the right people see they data. Integrity means ensuring only the right people or processes can modify the data. Availability means ensuring that people can access they data they need when they need it. These same aspects apply to human computation, but there are some unique problems arising for human computation. Researchers have postulated many different models of human computation, but by definition all these model involve humans aiding the computation. Humans have very different security properties and behaviors that will influence necessary changes.

For example, consider confidentiality. Often, proprietary information from several different stakeholders may occupy the same computer. A straight-foreword case involves business competitors sharing multi-million dollar high performance computers optimized for running complex models. Competitors can share this expensive resource securely because the computing service can wipe the computer completely blank before receiving the next competitor's model. Unfortunately, if we contemplate an analogous system that incorporates humans as computational elements, each of whom operates on proprietary data, we have no control on what aspects of the data the human participants may remember. Thus to use human computation safely for problem solving, each business would require its own population of human solvers to guarantee their secrets remain secret. This could result in stiff competition to recruit the best human solvers. Of course humans can keep secrets if motivated to by agreements, or laws. Even then, people often disclose little pieces of information they personally deem public or unimportant. If a competitor can aggregate all these little disclosures they might learn something about their competitor. A business does not have to worry about its computers making such self-directed decisions. Human computation systems confidentiality mechanism must address the fact that humans do not always act in predictable ways.

Humans also have different models of data integrity. Integrity ensures that people and processes only modify data in well-defined and understood ways. Computers are very good at following precise rules for modifying data. On the other hand humans almost always put in their own cognitive biases. For example, human solvers tasked with culling data with a specific rubric may systematically develop their own rubric culling more data than desired. Wikipedia serves as an example, where contributors change text descriptions to fit their own biases. Audit logs and history mechanisms can always track these changes, and technology can rollback the changes, but someone must know there was an erroneous change to start with. One can easily envision human computation systems that evolve to contain the cognitive

biases of the dominant sub-population of human solvers. When the system hopes to incorporate multiple viewpoints to cover different possible solutions, bias creep could shrink the number of viewpoints. Standard computer systems only contain the biases of their developers. Human computation systems will contain a myriad of biases. Attackers may change the computers behavior by modifying software, but the computer can be purged and start again with a clean state. Malicious users could attack the integrity of the human solvers directly, consciously and maliciously embedding biases in the solver population to prevent finding a solution.

Availability ensures that computer resources and data are available when needed. Providing availability for different parties using a shared resource remains a challenging problem. Denial of service attacks require little sophistication, but have required service providers to greatly expand their processing power to deal with them. In a human computation system with a large enough crowd spanning the globe there could always be a population of humans available. The question remains are they the correct humans for the problem being solved? Do they have the skills and knowledge to contribute quality effort? Also just because a human solver has been assigned a task does not mean they are actually performing it. Incentives may motivate humans to contribute in a timely fashion, but there are no guarantees. If incentives work to motivate contributors, what if a competing system has better incentives? How does one maintain a crowd of solvers if competitors are willing to pay more for solver services?

Human computation may require an additional security consideration beyond confidentiality, integrity and availability. Human computation systems consist of large distributed systems with crowds of human solvers each of whom may have a different world view and agenda. A security rule or mechanism that the system designers have carefully thought about and implemented to ensure proper behavior may seem arbitrary and whimsical to a given human solver that does not understand why the rule exists. Thus, human computation systems may benefit greatly by adding a forth security aspect of "why" to the system, that provides a rationale for system rules.

In high assurance systems, the rationale for every critical security decision exists in formal arguments to prove the software functions correctly. But this "why" never gets passed along to the human users, and in fact may be too complex for the users to understand. Without having a "why" component, security mechanisms could demotivate human solvers by making the system seem burdensome for no purpose.

So far people who understand the system create the security policies. They know why each security decision was made. Human computation distributes the system to many different humans, most of whom know nothing about the goals and reasons for the security policy. Dictating rules to humans volunteering their time to contribute will not be as successful as explaining to humans that the imposed measures protect the critical assets of the system from compromise for stated reasons. If people think a security rule or mechanism is arbitrary they will bypass it when they personally think the benefits outweigh the risks. However, since they do not understand the whole system and all its goals, even well intentioned solvers will make choices that compromise the developer's vision, putting the assets at risk. Human computation requires a security environment that motivates compliance, not defiance.

Human Computation Assets

What are the assets in a human computation environment? For example, solvers might add a file to a specific repository, or they may simply post on an Internet forum. Each of these interactions leaves a trace on the final solution, and becomes part of the critical assets the human computation system must protect.

The organization sponsoring the human computation environment shapes the security solution. Deciding on, implementing, and enforcing security falls squarely on the shoulders of the sponsors. The sponsors benefit from finding a solution, so security compromises that hinder finding a solution directly impacts the sponsors.

A strawman list of human computation assets includes:

- Solutions—solutions emerge from human solver interactions. They cannot be destroyed because the human solvers can recreate them, but they can be stolen.
- Problem specifications—A concise problem specification that motivates human contributors to donate their time is critical to find a solution
- Contributions to a solution—any human solver interaction that moves closer to a solution
- Contributions that do not lead to a solution—these interactions have value because they document parts of the solution space that have been explored and eliminated. Losing them might mean others would invest time exploring the same space again
- The human solvers themselves—they represent the most valuable asset, as no solution will be found without them
- The human computation solution environment—an environment that supports massive collaboration and that can produce solutions to unsolved problems has tremendous value
- Rewards for human solvers—any reward that motivates human solvers will be desired and face security threats by people that want reward, but do not want to do the work.

Intellectual Property

As the asset list for a HC-based problem solving system shows, the intellectual property includes the problem specification, the solution and all the contributions by human solvers. Not all of this IP has the same intrinsic worth. The likelihood of finding a solution increases the more the solvers share information. The more information you share the less control you have over it. Human solvers will need a bare minimum of information simply to get started. Protected IP that no one sees does not help produce a solution, since information has no value unless it allows a human to make a better decision.

Consider, for example, the problem statement for a project that was successful because it found a solution. In this case the problem statement served as an effective

marketing campaign that got the right human solvers interested and engaged in solving a problem. If the sponsors did not share the problem statement, chances are the right solvers never heard of the problem. Sharing information provides critical momentum for the project's success.

From a computer security point of view, sharing with another human represents letting the cat out of the bag. Technology cannot put the cat back in the bag, or even allow a peek at the cat in the bag without letting the cat out. Once information transfers to a human mind, that human can duplicate the information and bypass any technology. In the world of government security they have created a procedure to address this fact based on clearances. A security clearance represents a contract between two parties to share the information in controlled and predefined ways. A security clearance represents trust between two parties. For the most part trust in clearances works, but it can also fail spectacularly in cases such as wikileaks (Keller 2011).

Human computation environments may need to create an agreement that parallels government clearances. It might be as simple as a non-disclosure agreement, or it could be a complex set of clearances that allows different solvers to see different pieces of IP. Whatever the agreement winds up being, it must have some teeth, some penalty for the human solvers that break the trust. Breaking a non-disclosure agreement could be resolved in a court of law, but proving the amount and value of information disclosed may make such a court case hard to win, or simply too cost prohibitive to ever enforce. Other penalties might include ostracizing violators from the site or other reputation degradation penalties for violators. Reputation penalties require associating the solver's real world identity to the human computation environment to ensure the penalty actually penalizes the violator.

In the case of altruistic goals, sharing the IP maybe considered a good thing, so no agreement is needed. But IP that solves a technical problem may be repurposed to solve other related problems. For example, a solver could learn something from the computation system and use that to start a competing product. The sponsors would harvest no benefit from that product, but maybe they would accept that risk to allow them to make progress on their core problem. The sponsors must decide if they would rather reap rewards from partial or tangential solutions, or if they want a free exchange to increase the chance of finding a solution.

The Crowd

The human solver crowd represents the most valuable commodity for the sponsors. Without human solvers the solution stagnates. Jane McGonigal has postulated an engagement economy that competes for the eyes and brains of humans to join into human computation environments (McGonigal 2011). Many collaboration sites exist that never got the minimum number of people involved to make progress toward the goal. Solvers typically can change their minds and switch alliances to other sites, or simply decide to no longer invest their time in a specific site.

For example, suppose a person did not like the altruistic goal of a human computation environment. How can they prevent the sponsors from reaching that goal? They can push all the human solvers away from the site. They could try doing it with a negative advertisement campaign, but a more subtle and effective approach would be to pose as a legitimate solver and cause the system to crash or become unstable. If people think a web site is unstable, or the site appears to drop their work they will choose to stop investing their time in that site, even if they still believe in the altruistic goal.

Another attack on the crowd of human solvers would be a well-placed trolling attack. Comments like, "You call that a logical argument?" or, "That will never work!" can derail a collaborative effort. If human solvers feel no one appreciates their contributions, or if associating with the site makes them feel bad they will stop interacting with the site. If the site allowed anonymous posting, you could envision a malicious bot that randomly posts negative comments on forum threads, which we will call a robo-troll attack. With no human interaction by the attacker, the overall sentiment of the site becomes negative and the attrition rate of legitimate solvers will climb.

A more subtle robo-troll attack involves posting legitimate, but dumb answers. The software could use even poor natural language processing software to create posts that sound like they are related, but that make no sense. These poor posts will waste legitimate human solvers time reading and responding to them. Eventually they may feel they contributions are falling on deaf ears and drop out of the project.

Tying human solvers to their real world identity will curb robo-troll attacks because at the very least a single human must register the account. Unfortunately, once the account is created the malicious attacker could install a robo-troll that continues to post around the clock greatly magnifying the amount of damage a single attacker can do. This implies human computation environments will need a reputation system that eventually silences people that continuously make unconstructive contributions. Such censorship must be clearly explained to the other solvers so they do not feel the site has become draconian.

Reward

Many human computation environments build motivation by rewarding the human solvers for their participation. The reward may be as simple as an in game reputation score, or it may be a large monetary rewards for winning a contest. Any reward that motivates human solvers will be coveted by people who want the reward without doing the work.

What types of rewards are there?

- Money
- Reputation
- Altruistic
- In environment rewards
- Education/knowledge

Money

A human computation environment may use money in a variety of ways. The amount can range from micro-payments to large cash payouts for winning a contest. The sponsors determine the rewards based on the behavior they need to motivate, and the available budget.

Sites like Amazon Mechanical Turk provide micro-payments for doing tasks requiring human insights. Micro-payments can motivate people with low earning potential, or provide additional incentive for doing a worthwhile task, such as participating in a scientific experiment. Usually the sponsors have large numbers of simple tasks like categorizing images. At any one time the sponsor can check the work of the person and decide if they are doing the job correctly. Often these solvers must first perform a qualifying task that establishes the trust relationship between the worker and the sponsor. During the qualifying task the sponsor can check the work against expected responses to ensure the workers simply isn't picking random answers. Often sponsors then have multiple people perform the task and compare the tasks to check for people who pass qualification and then revert to random guessing.

Some micropayments are used to elicit opinions from humans. Sponsors cannot check a person's opinion for correctness. People trying to earn more micro-payments faster could create programs to answer opinion questions randomly. This attack, which I will call robo-pundits, generalizes to all opinions systems. In the case of Amazon mechanical Turk the on-line pseudonym is tied to the persons real world identity to allow the robo-pundit to collect the micro-payments. However, robo-pundits could make many shell web accounts that funnel to the same bank account. In this case you would have to follow the money to ensure you have only one human associated with each web account.

We haven't seen a large impact of robo-pundits, because to earn a lot of money still requires a lot of human intervention and opinion systems pay only small rewards. It is clear that people will exploit this avenue of attack when suitable rewards exist. Already many product review sites have been tainted by people willing to use their real world reputation to extol the virtues of a product for a micro-payment. By applying robo-pundit technology these people could greatly increase their reward.

In the case where sponsors offer real money for completing some task, such as a contest to find a specific solution, the security posture changes. For large prizes the sponsors must scrutinize submitted solutions to ensure they satisfy the win criteria. Stealing solutions in contests has already been seen in the U.S. State Department "Tag Challenge" by attacking the reputation of other teams to steal their crowd of followers (Rahwan et al. 2013). Interestingly, combining ideas to solve a problem provides a valid way to solve problems, but the originating human solvers will find other people benefiting from their efforts detrimental to motivation. In these situations people or teams will keep their research and work secret to prevent theft. Hoarding insightful information will hinder finding solutions. So the sponsor must carefully set the reward criteria to reward the behavior they want, and provide the necessary security to protect solvers efforts.

Depending on the problem domain several solutions might provide viable protection to the sponsors. First the sponsors could strip off the domain-specific jargon and try to have the solvers address the generic problem. The problem remains if the solver can solve the general problem, they probably possess enough intelligence to apply the solution to different domains. The second approach protects the solution by breaking the problem into many smaller problems. When the problems assigned have little context, the chances of a solver seeing how the pieces fit into the solution shrink. Sometimes, this may remove some cognitive biases that will allow the solver to see a solution, but often it will handicap the solver because they won't have enough context to find a solution. Fold-it provides an example where the shrinking context could protect the final solution without handicapping the solvers. Fold-it provides a game where the crowd manipulates protein structures in three dimensions to determine how they will fold (Cooper et al. 2010). Here the crowd uses its understanding of manipulating objects in three dimensions, but that does not translate to how the shape of the resulting protein interacts with other proteins. Solvers may recognize the protein being folded which may reveal some information the sponsors would like to keep secret. Striking a balance between disclose and protecting assets will be a constant balancing act for problem solving environments.

Human solvers will attempt to maximize their monetary reward, and unwary sponsors may be surprised at how clever the solvers are at circumventing the intent of the rules simply to win the reward. The sponsors should adopt proven reward functions or even run small contests with smaller rewards to see if anyone can find loopholes in the reward criteria. One safety net clause to put in the contest rules may simply state that a valid solution must meet the intent of the contest as defined by the sponsors. Legitimate solvers will probably not be concerned about such a statement.

Reputation

If a human builds up a reputation for a pseudonym on a human computation site, an attacker can potentially steal the identity and reap the benefits. However, when tied to the real world identity the victim can prove they are who they say they are through conventional means, like passports and fingerprints.

People will attempt to steal reputation as well. In the 1990 one of the early firewalls, Sidewinder, hosted a contest to break the security of its firewall (Thomsen 1995). The prize was a custom leather jacket with the Sidewinder logo on back. As the contest moderator I was surprised how many people claimed to have hacked the firewall and received the jacket even though no one ever met the victory criteria. Claiming victory without producing the jacket costs the reputation stealer nothing, but the cost of a custom leather jacket would be a small cost for someone hoping to establish his reputation in the hacker community. This reinforces the idea that contest sponsors must protect solver reputations to avoid de-motivating them.

Altruistic

Contests that motivate solvers to achieve altruistic goals like curing cancer provide the solver an intrinsic reward that cannot be stolen. Or can it? What if a malicious person published a false account of the same cancer curing technology being used to create a bio-weapon? The solver may loose the feeling of accomplishment for aiding the effort to find a cure.

Humans can delude themselves into thinking they have contributed. Maybe the person simply created an account on the collaboration site, or simply talked to someone else who had, and because of that they felt good when a solution was found. In the end, the sponsors will still have gotten a solution and some person will have felt better about himself for no reason. While this may seem benign in the case when the system found a solution, consider the case when the site never motivates enough solvers to actually perform work and thus no solution is found. Sponsors must clearly define what constitutes a solid contribution and advertise it to potential solvers. Altruistic rewards require proactive protection by the sponsors just as much as monetary rewards.

In Environment Rewards

"In environment" rewards represent unique, often digital, goods used in human computation environments as rewards. Suppose for example that the Farmville game (a game about planting crops) was actually a serious human computation system with a purpose of finding optimal crop rotations. The special edition seed planter given out to those that participated in March represents a badge of accomplishment intended to reward participation over a specified period of time, but also it represents something that people value because it helps them play the game. Possessing such an item reinforces putting in the work to get the reward.

In game digital rewards like this cost the sponsors very little and can provide significant motivation to solvers to engage in behaviors the sponsors believe will result in a solution. When the rewards only live in the collaboration environment the sponsors can create sufficient security to ensure that the no one can counterfeit the digital goods. In the very least proper auditing of solver behavior could reveal whether the solver earned the reward or not.

Some organizations allow the digital goods to go beyond the collaboration environment. Mozilla has a project at openbadges.org that allow organizations to create badges people can display, and which interested parties can authenticate to ensure the badges' legitimacy. Such portable digital goods provide more motivation for solvers because they can use their digital goods in more places. Such portability requires a long-lived infrastructure to provide authentication. Future rewards might consider code fragments that provide some utility. Consider for example instead of an animated gif, an interactive gif as a reward that a solver can put on her own site.

For either reward, the sponsor must protect the solver's value by preventing others from copying the image or code fragment for display on their own web site when they did not earn the reward.

Education

One reward comes as a natural by-product of solving hard problems: education. If the problem requires human insight to solve, there is a good chance the human will learn something in the process. This includes insights into the target problem and unique problem solving skills. Educational rewards cannot be stolen, and they cannot be earned without the solver doing the work.

Summary

The goal of freely sharing information to solve problems directly conflicts with the system's ability to protect intellectual property without creating some sort of user agreement to mitigate the risk. Robo-troll attacks present a new kind of attack specific to human computation environments designed to erode the number of human solvers on a project; the project's most valuable asset.

Many of these attacks point to solutions that do not allow anonymous human solvers, but tie the solvers to their real world identities. This allows for real world punishment for rule breakers, but also reduces the number of robo-troll attacks. When tied to a real world identity, reputation remains a reward that costs the sponsors little and cannot be stolen or sold.

Overall the magic of the human computation comes from bringing many people together, at their convenience to solve a problem. While it would be nice to create a cyber-utopia where people cooperate freely to solve the problem, unfortunate aspects of the real world will assert themselves to interfere with this goal. Fortunately, many of the solutions to these real world problems can also be applied in cyber space. For example, if the sponsors know who is contributing and in what ways, they will be able to execute both punishments and rewards that have an impact. Follow the money may seem cynical, but it provides the best insight on where the problems will emerge in the new area of human computation solution environments.

References

Cooper S et al (2010) Predicting protein structures with a multiplayer online game. Nature 466(7307):756–760

Keller B (2011) Dealing with Assange and the WikiLeaks secrets. New York Times

McGonigal J (2011) Reality is broken: why games make us better and how they can change the world. Penguin Pr, New York

Rahwan I et al (2013) Global manhunt pushes the limits of social mobilization. Computer 64:68–75

Thomsen DJ (1995) The sidewinder challenge—results so far. Electron Cipher IEEE Secur Priv Newslett