# Bazaar: Strengthening user reputations in online marketplaces

Ansley Post[†‡] Vijit Shah[*] Alan Mislove[*]

[*]*Northeastern University* [†]*MPI-SWS* [‡]*Rice University*

## Abstract

Online marketplaces are now a popular way for users to buy and sell goods over the Internet. On these sites, user reputations—based on feedback from other users concerning prior transactions—are used to assess the likely trustworthiness of users. However, because accounts are often free to obtain, user reputations are subject to manipulation through white-washing, Sybil attacks, and user collusion. This manipulation leads to wasted time and significant monetary losses for defrauded users, and ultimately undermines the usefulness of the online marketplace.

In this paper, we propose Bazaar, a system that addresses the limitations of existing online marketplace reputation systems. Bazaar calculates user reputations using a max-flow-based technique over the network formed from prior successful transactions, thereby limiting reputation manipulation. Unlike existing approaches, Bazaar provides strict bounds on the amount of fraud that malicious users can conduct, regardless of the number of identities they create. An evaluation based on a trace taken from a real-world online marketplace demonstrates that Bazaar is able to bound the amount of fraud in practice, while only rarely impacting non-malicious users.

## 1 Introduction

Online marketplaces like eBay, Overstock Auctions, and Amazon Marketplace enable buyers and sellers to connect regardless of each other's location, allowing even the most esoteric of products to find a market. These marketplaces have greatly expanded the set of people who can act as a buyer or seller and, thus, can be viewed as democratizing commerce. These sites are extremely popular with users; in 2009, over $60 billion worth of goods was exchanged on eBay alone.

This new freedom, however, does not come without challenges. Online marketplaces are known to suffer from fraud, and often rely on user reputations—formed from the feedback provided by other users—in an effort to mitigate the effects of malicious activities on their sites. For example, on eBay, potential buyers often examine the reputation of the seller to determine the seller's trustworthiness. In fact, it has been observed [13, 15, 19] that sellers with highly positive reputations tend to sell goods at a higher price when compared to sellers with lower reputations, demonstrating the central role that user reputations play in online marketplaces. Malicious buyers (who do not pay for goods purchased) and malicious sellers (who do not deliver the promised goods) quickly gain bad reputations and are avoided [11].

One challenge, however, is that accounts on online marketplaces are often free to create (usually only requiring filling out a form and solving a CAPTCHA [23]), to avoid discouraging potential users. As a result, reputations derived from user feedback are still subject to three types of manipulation:

- Malicious users whose accounts have a bad reputation can effectively *white-wash* their reputation by creating a new account with a blank reputation.
- Malicious users can *collude* by providing positive feedback on each other's transactions, thereby improving both of their reputations.[1]
- Malicious users can create fake identities, known as *Sybils* [7], and use these to provide positive feedback on fictitious transactions between the various identities, thereby inflating their reputations.

Reputation manipulation can lead to significant monetary losses for defrauded users. For example, a single malicious eBay user was recently found to have created 260 different accounts, fabricated positive feedback, and stolen over $717,000 from over 5,000 users [24]. This

[1] In fact, this type of abuse can be plainly viewed on eBay by searching for auctions that are selling "positive feedback." As of this writing, 350 such auctions exist for prices ranging from $0.01 to $0.99.

case is hardly unique: Another malicious eBay user was arrested after defrauding others of over $1 million [20].

In this paper, we propose Bazaar, a system that strengthens user reputations in online marketplaces in the face of collusion, white-washing, and Sybil attacks. Bazaar creates and maintains a *risk network* in order to predict whether potential transactions are likely to be fraudulent. The risk network consists of weighted links between pairs of users who have successfully conducted transactions in the past. When a transaction is about to be completed, Bazaar calculates the max-flow between the buyer and seller; if it is lower than the amount of the transaction, the transaction is flagged as potentially fraudulent. Since Bazaar only needs to determine whether the max-flow is above a given value (instead of calculating the exact max-flow), Bazaar stores the risk network using a novel *multi-graph* representation. We demonstrate that this results in a substantial speed-up of Bazaar's max-flow calculation while imposing only a modest storage overhead.

Bazaar provides a number of useful security properties: First, malicious users in Bazaar cannot conduct more fraud together than they could separately, and as a result, there is no incentive for malicious users to collude. Second, malicious users cannot gain any advantage from conducting Sybil attacks, and thus, there is no incentive to create multiple identities. Third, Bazaar explicitly allows users to create as many identities as they wish; this is sometimes a desired feature in online marketplaces, where sellers may own multiple businesses or wish to maintain separate identities for different types of goods. Fourth, Bazaar provides a strict guarantee that each user can only defraud others by up to the amount of valid transactions the user has participated in, regardless of the number of identities the user possesses, thereby bounding the potential damage.

We evaluate Bazaar using a trace collected from eBay, the largest online marketplace. We collected a 90-day history of five of the most popular categories on the eBay United Kingdom site, encompassing over 3 million users and 8 million auctions. Simulating Bazaar on this data set, we demonstrate that Bazaar successfully bounds the amount of fraudulent transactions that malicious users can conduct, while only rarely impacting the transactions that occur between non-malicious users. We demonstrate that if Bazaar had been deployed on eBay during the 90-day period and in the five categories we study, it would have flagged over £164,000 of auctions that eventually resulted in negative feedback as potentially fraudulent, substantially increasing the reliability of the online marketplace.

The rest of this paper is organized as follows. Section 2 describes the approaches that are currently taken to secure online marketplaces, and Section 3 provides more detail on different types of fraud that are still present today. Section 4 describes the design of Bazaar in detail, and Section 5 details the multi-graph representation of the risk network. Section 6 presents an evaluation of Bazaar. Section 7 details related work and Section 8 concludes.

## 2 Background

Online marketplaces often use site-specific mechanisms for fraud prevention, but many of these can be reduced to a few simple techniques:

**Making joining the market difficult** Certain marketplaces only allow trusted users or organizations to participate as sellers, often requiring upfront fees or accounts backed by difficult-to-forge financial information. An example of such such an approach is Amazon Merchants [3], which requires bank account information, a $40-per-month fee, and pre-approval for listing high-fraud-risk goods. However, by making it more difficult to join, this approach reduces the usefulness of the marketplace and severely restricts the population of sellers.

**Using a trusted broker** In some marketplaces, a middleman participates in the transaction and holds payment until the buyer is satisfied with the transaction. For example, on eBay, there are escrow services that hold money for transactions until the buyer has received the good. However, brokers typically charge a fixed fee and a percentage of the sale,[2] increasing the transaction cost and making escrow practical only for expensive goods (representing a small minority of the goods on typical marketplaces).

**Requiring in-person transactions** Other marketplaces such as Craigslist require buyers and sellers to be within the same geographical area, ensuring that the participants can meet in person to complete a transaction. This approach allows buyers to inspect goods, and sellers to verify payment, before going through with the transaction. However, this approach also severely restricts who is able to buy and sell goods from each other (as the buyer and seller must live close to each other), limiting its usefulness to local marketplaces.

**Providing insurance** Certain marketplaces offer buyer and seller insurance programs, either by default or for a fee. However, coverage is generally limited to certain geographic regions and the cost of the insurance payouts and program administration results in higher fees for marketplace users. Nevertheless, the information that Bazaar provides can be viewed as an estimate of risk be-

[2]For example, eBay's recommended escrow service charges a minimum of $22 and up to 3% of the transaction cost.

tween two parties, and can therefore be used as an input when choosing the appropriate the insurance premium.

**Paying via trusted services** Because certain payment methods (e.g., money orders) are difficult to recover, many marketplaces suggest or require that trusted online payment services (e.g., PayPal) be used. Ideally, such services would link accounts to real-world financial information, making the creation of multiple accounts difficult. However, this is not the case: For example, receiving money with a PayPal account only requires an email address (although financial information is required to withdraw funds). Thus, malicious users can receive money with networks of email-backed accounts, and then send that money to the single, "real" account that is able to withdraw money.

**Leveraging feedback** Finally, many online marketplaces use feedback provided by users who have participated in transactions. For example, eBay's feedback mechanism calculates a score for each user, consisting of the amount of positive feedback minus the amount of negative feedback. Users with highly positive feedback scores are considered to be more trustworthy, and have been observed to sell goods for higher prices [13, 15, 19]. This approach has the advantage of not restricting marketplace membership and allowing any buyer and seller to participate in a transaction. However, as we will observe in the next section, using feedback is often subject to manipulation by malicious users.

Ideally, we would like to prevent fraud without unnecessarily restricting participation in the online marketplace. The first four approaches above artificially restrict the marketplace by making it either harder to join, more expensive to use, segmenting it based on geography, or spreading the cost of fraud to all users. Thus, we focus on the last approach, leveraging feedback, for the design of Bazaar and present a design that is not subject to the manipulation of existing approaches. Focusing on user feedback also has the advantage that is the mechanism used by the largest online marketplaces, such as eBay, meaning Bazaar could be directly applied to such sites.

# 3 Examples of malicious behavior

We motivate the design of Bazaar by examining several types of fraud that have been observed in online marketplaces today. The eBay dataset that we use for illustration is fully described in Section 6, however, our purpose here is simply to provide a few motivating examples. In this section, we focus on *malicious sellers* who attempt to defraud buyers, as sellers are largely protected from *malicious buyers* by being allowed to verify payment before shipping the good. To define the fraud we observe, we look at various sellers' feedback history, consisting of entries recording whether the buyer was satisfied with the transaction.

For clarity, we begin by examining the feedback history of a typical seller, shown in Figure 1 (a). Even though over 99% of the seller's feedback is positive, a few items of negative feedback can be observed. A certain low level of negative feedback is expected even for non-malicious sellers, as some buyers may have been unsatisfied with their purchase (e.g., due to the good being lost or damaged in transit, a miscommunication between the participants, or buyer's remorse). We will use similar timeline diagrams throughout the rest of this section.

## 3.1 Leaving the marketplace

One of the most common types of fraud occurs when a seller participates in the marketplace as a non-malicious user for a period of time, and then turns malicious (often by starting to conduct transactions without ever shipping the goods). As a result, the unsuspecting buyers who have not yet received their goods are defrauded. This type of fraud can be detected once the buyers begin to provide negative feedback, serving as a warning to others. However, malicious users often take advantage of the "window of opportunity" before the negative feedback appears: They can advertise and accept payment for a large number of goods before any user realizes that a fraud has occurred.

An example of such a malicious seller is shown in Figure 1 (b). Towards the end of the seller's timeline, he lists a significant number of goods that are never delivered and eventually result in negative feedback. In fact, this user made significantly more money in aggregate from the fraudulent transactions than from the non-fraudulent transactions. The underlying problem is that *in-progress transactions are not counted against a seller's reputation*, enabling malicious users to establish a reputation, defraud users with the window of opportunity, and then re-join the site with a new account.

## 3.2 Hiding fraud in the noise

As an alternative to leaving the marketplace, malicious users have also been observed to "hide the fraud in the noise" by participating in many non-fraudulent transactions, but conducting fraudulent transactions for (relatively) expensive goods. As a result, their feedback history has only a small amount of negative feedback, and only a close inspection of the transaction values reveals the fraud. An example of a malicious user conducting such fraud is shown in Figure 1 (c), where the user made more money through the two fraudulent transactions than through the hundreds of non-fraudulent

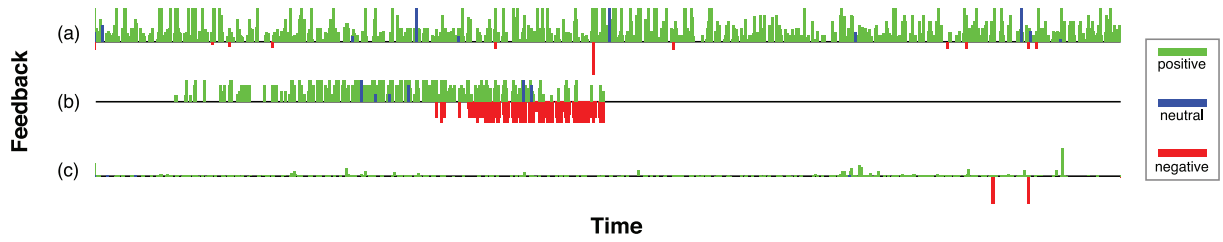


Figure 1: Auction feedback history over time for three eBay sellers: (a) a typical seller, (b) a malicious seller who leaves the marketplace, and (c) a malicious seller who hides the fraud in the noise by conducting a few, large fraudulent transactions. Positive feedback is shown in green, neutral feedback in blue, and negative feedback in red and below the line. The size of each bar correspond to the log of the value of the auction.

transactions. The underlying problem is that *the value of transactions is not considered when determining a seller's reputation*, enabling malicious users to conduct a high-value fraudulent transactions with the same effective penalty (one piece of negative feedback) as a low-value fraudulent transaction.

### 3.3 Conducting fictitious transactions

Malicious users have also been observed to conduct fictitious transactions and provide fictitious positive feedback. The ultimate goal of these transactions is not to sell a good, but rather, to improve the user's feedback score, making the user look more like an non-malicious user. For example, numerous auctions on eBay are labeled with "Positive Feedback Guaranteed." Often, these auctions ostensibly offer a copy of a digital picture or other token item, so as to appear as a legitimate auction.

Thus, it is easy for a malicious user to arbitrarily manipulate his feedback score by adding spurious positive feedback, so as to appear as a legitimate seller. The underlying problem is that *feedback counts the same, regardless of the other user providing the feedback*. This allows malicious users to conspire to inflate each other's feedback score (or, a single malicious user to do the same via a Sybil attack).

### 3.4 Summary

In this section, we described three of the most common types of reputation manipulation that are present in the online marketplaces of today. In the next section, we describe the design of Bazaar, which addresses each type of manipulation by (a) considering outstanding transactions, (b) taking into account the value of transactions with positive and negative feedback, and (c) discriminating between different users' feedback, in order to prevent malicious users from artificially inflating their reputation.

## 4 Bazaar design

We now describe the design of Bazaar.

### 4.1 Overview

Bazaar is intended to augment an online marketplace, run by a marketplace operator, where buyers and sellers may have no previous relationship and accounts are free to obtain. In such systems, buyers must rely on the reputation of the sellers, represented by feedback from other buyers, to distinguish between non-malicious and malicious users. Thus, the goal of Bazaar is to protect buyers from malicious sellers who manipulate their reputation so as to appear non-malicious. Additionally, we aim to keep the existing model and basic user operations, while significantly reducing the vulnerability to fraud. By doing so, Bazaar serves as a drop-in component applicable to numerous marketplaces.

Now, let us introduce a few definitions that we use for the remainder of this section. A *user* corresponds to an actual person in the offline world. An *identity* is an online account with a particular username associated with it. A user can have a potentially arbitrary number of identities. A transaction is an event where two identities agree to a sale, which has some value. Note that both identities in a transaction may correspond to the same user.

Bazaar relies on two insights. First, successful transactions between different users require significant effort and risk for both parties. Both users are trusting the other to complete the transaction, by providing payment or delivering the good. We refer to this as *shared risk* between two users. Second, once a transaction has been successfully completed, the two users are more likely to enter into a transaction together in the future. Note, however, this risk in not unbounded, and is dependent on the type of transaction that has occurred: The amount of risk that two users are willing to undertake is likely proportional to the amount of risk that has been successfully rewarded.

## 4.2 Risk network

We view a successful transaction as linking two identities in an undirected fashion, where the weight of the link is the aggregate monetary value of all successful transactions—successfully rewarded shared risk—between the two identities. For example, if identities $A$ and $B$ participated in two successful transactions for \$5 and \$10, there would be an $A \leftrightarrow B$ link with weight \$15. Note that link weights must always be non-negative.

The set of all such links forms an undirected network, which we refer to as the *risk network*. An example of such a network is shown in Figure 2 (a). Note that the risk network has a particularly useful property: The weights are automatically generated by user actions, and do not have to be explicitly provided by users. As we demonstrate below, the risk network can be used not only to gauge the risk between two identities who have conducted a transaction in the past, but also between arbitrary identities who may not have directly interacted in the past.

## 4.3 Design

Bazaar is run behind-the-scenes by the online marketplace operator. The basic operation of Bazaar is simple: When a buyer is about to enter into a transaction, the marketplace operator queries Bazaar, which calculates the max-flow in the risk network between the buyer and the seller. If the max-flow is below the amount of the potential transaction, the marketplace operator flags the transaction as potentially fraudulent. We discuss ways in which this output can be used by the marketplace operator in Section 4.5, but for now, we assume that flagged transactions are blocked.

The intuition for this approach lies in the observation above about shared risk. Consider a risk network with only two identities, connected by a link of weight $w$. The identities may be willing to engage in another transaction of value $w$, and if that is successful, then another transaction for a higher amount. Bazaar generalizes this intuition, allowing identities who are not directly connected to engage in a transaction as long as there is a set of paths of sufficient weight connecting them. For example, in the network shown in Figure 2 (a), if $A$ was about to buy a good from $D$, Bazaar would consider the flow on paths $A \leftrightarrow B \leftrightarrow D$ and $A \leftrightarrow C \leftrightarrow D$ in order to determine $D$'s reputation from $A$'s perspective.

In existing online marketplaces, feedback-based reputations are "global," in the sense that everyone has the same view of a given user's reputation. In Bazaar, reputations are a function of both the user who is being asked about as well as the user who is asking. As we demonstrate below, this approach allows Bazaar to mitigate reputation manipulation: Malicious users who conspire to inflate their reputations do not necessarily increase their reputations from the perspective of non-malicious users.

### 4.3.1 Putting credit "on hold"

The design of Bazaar is complicated by the fact that the buyer may not be able to determine whether the transaction was fraudulent immediately after sending payment for the good; generally, there is a delay between when he agrees to the transaction and when the good arrives. In order to prevent malicious sellers from abusing these outstanding transactions in the manner observed in Section 3.1, when the buyer decides to go through with the transaction, Bazaar first determines a path set[3] between the buyer and seller that has a total weight of at least the transaction amount. Such a path set must exist, as, otherwise, the max-flow between the buyer and seller is lower than the transaction amount (meaning Bazaar would have flagged the transaction as potentially fraudulent).

Once the path set is determined, Bazaar temporarily lowers the weights on these paths (in aggregate) by the transaction amount. In essence, this puts the weight on these paths "on hold" until feedback concerning the success or failure of the transaction is received. Since each link weight must always be non-negative, this approach prevents the malicious users from leveraging the weight that is "on hold" in order to conduct additional transactions.

Continuing with our running example in Figure 2, the initial state of the risk network is shown in Figure 2 (a), with each identity having participated in transactions with two other identities. Then, suppose that $A$ conducts a \$10 transaction with $D$. Bazaar determines that the max-flow between $A$ and $D$ is greater than \$10, and therefore allows the transaction to go through without being flagged. In doing so, Bazaar temporarily lowers the links along the path set by a total of \$10 (specifically, \$2 is lowered off of the $A \leftrightarrow B \leftrightarrow D$ path and \$8 is lowered off of the $A \leftrightarrow C \leftrightarrow D$ path). This is shown in Figure 2 (b).

### 4.3.2 Responding to feedback

Finally, once the buyer provides feedback about the transaction, Bazaar makes changes to the risk network. These changes depend on the feedback from the buyer:

- **Positive feedback** If the buyer reports a successful transaction, indicated by positive feedback, Bazaar restores the temporarily lowered weight and additionally creates a new link directly between

[3] If multiple path sets exist that have sufficient weight, Bazaar simply picks one of these sets randomly.

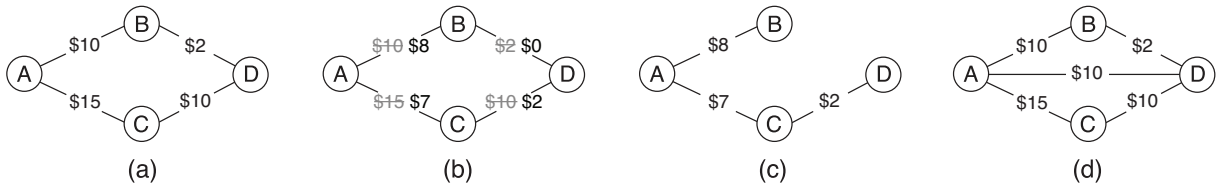


Figure 2: State of the risk network while $A$ conducts a \$10 transaction with $D$. The state is shown (a) before the transaction, (b) while waiting for feedback, (c) if the buyer reports an negative feedback, (d) if the buyer reports a positive feedback, and (a) again, if the buyer reports neutral feedback or the timeout expires.

the buyer and seller weighted by the transaction amount.[4] This has the effect of both restoring the network to its previous state, and creating a new risk link between the buyer and seller. The intuition for this action follows from the discussion above, whereby the buyer and seller are more likely to enter into a future transaction together.

- **Neutral feedback** If the buyer reports a partially successful transaction, indicated by neutral feedback, Bazaar restores the temporarily lowered weight, but does not create a new link. This has the effect of restoring the network to its previous state, with no changes. The intuition for this action is that users who provide neutral feedback are not claiming that the transaction was fraudulent, but are not completely satisfied. Thus, the buyer is not likely to enter into a future transaction with the seller, but does not wish to punish the seller by providing negative feedback.

- **Negative feedback** If the buyer reports an unsuccessful transaction, indicated by negative feedback, Bazaar makes the temporary lowering of the weights permanent and does not create any new links. This has the effect of reducing weight on the seller's links, thereby decreasing the seller's ability to conduct transactions in the future without having them flagged. In particular, if the seller conducts many transactions that end up with negative feedback, eventually, all of his links will be exhausted, and he will be unable to conduct any non-flagged transactions.

- **No feedback** Finally, if the buyer does not report feedback at all, a configurable timeout of $T$ is used, after which Bazaar responds as if the buyer provided neutral feedback (i.e., the temporarily lowered weight is restored, but no new link is created). This is similar to existing sites, which often have a time cutoff for providing feedback.

[4]If a direct link already existed, then Bazaar simply increases that link's weight by the transaction amount.

Returning to our running example in Figure 2, suppose that the feedback is received or the timeout occurs. Bazaar either makes the weight reductions permanent if the buyer reports negative feedback (Figure 2 (c)), restores the previous weights and also forms a new $A \leftrightarrow D$ link if the buyer reports positive feedback (Figure 2 (d)), or restores the previous weights if the buyer reports neutral feedback or the timeout occurs (Figure 2 (a)).

The intuition for why Bazaar is able to prevent fraud is demonstrated by the network shown in Figure 3, where a malicious user $X$ has created a number of identities ($X_1$ ... $X_5$) and has conducted fictitious transactions between them (in essence, the weight on these links can be arbitrarily set by $X$). Without Bazaar, potential victim $Z$ would only see $X_1$'s fictitious feedback consisting of a number of positive entries. Not knowing that all of this positive feedback was from other identities owned by the same underlying user, $Z$ would likely be defrauded. With Bazaar, however, the fictitious transactions do not contribute to the max-flow between $Z$ and $X_1$, and Bazaar is likely to flag the transaction as potentially fraudulent (even though Bazaar had no a priori knowledge that all $X_i$ identities belong to the same user). Moreover, should

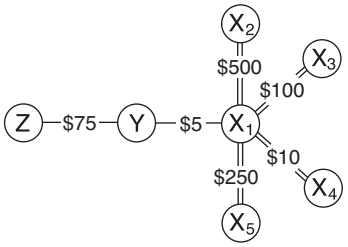


Figure 3: Example risk network, showing why Bazaar secures reputations (links represent previous real transactions, and double links represent fictitious transactions). Honest identity $Z$ is considering entering into a transaction with malicious identity $X_1$ (owned by the same user as $X_2$ ... $X_5$). Without Bazaar, $X_1$ appears to be a reputable seller. With Bazaar, the fictitious transactions do not increase the max-flow (\$5) between $Z$ and $X_1$, thereby preventing the reputation manipulation.

$X$ use one of these identities to conduct a fraud—of no more than $5, since anything greater would be automatically flagged as potentially fraudulent—the $Y \leftrightarrow X_1$ link will have credit put "on hold" and eventually reduced (once the buyer provides negative feedback), regardless of which identity $X$ selects as the seller. This is the case regardless of the number of identities $X$ creates or how he creates fictitious transactions between them. In effect, Bazaar forces $X$ to participate in successful transactions with other non-malicious users in order to increase his max-flow, and penalizes these links whenever $X$ conducts fraud.

#### 4.3.3 Bootstrapping

New users, by definition, have no transaction history and therefore have a max-flow of 0 to all other users. To allow new users to participate without having all of their transactions flagged as potentially fraudulent, Bazaar uses two techniques. First, Bazaar allows users to create virtual links to their real-world friends (in the same manner as malicious users can create links in the risk network between their identities by conducting fictitious transactions). This mechanism allows users to obtain a few "starter" links from the friends, without opening a new security vulnerability: Since the user's friends are, in effect, vouching for the new user, the friends are putting their existing links on-the-line. If the new user defrauds others, not only would his links be penalized, but the links of his friends would be as well.

Second, if the new user does not have any real-world friends in the marketplace, Bazaar allows him to optionally provide the marketplace operator with an amount of money to hold in escrow. In return, the marketplace operator creates links between the new user's identity and other, random identities with a total value of the amount in escrow. These newly created links allow the new user to participate in the marketplace. At some later time, the new user can request that the escrowed money be returned (and the marketplace operator will remove the created links). However, if the created links represent weight on hold, or if the they have been lost (due to a fraudulent transaction), the marketplace operator would refuse to return the escrowed money. This approach does not open up a new vector for attack, as (a) the most the new user could defraud is the amount of escrowed money, and (b) if the user does commit such a fraud, he would lose his escrowed money. In essence, such an attack would not allow a malicious user to gain any money.

### 4.4 Guarantees

We now discuss the guarantees that Bazaar provides. In brief, Bazaar ensures that malicious users can only defraud others up to the total amount of successful transactions that they have participated in with non-malicious users. To see this, let us imagine a malicious user $X$, whose identity has outgoing links with weight totaling $a_X$. Each time $X$ conducts a fraudulent transaction, some of his links are reduced, in aggregate, by the amount that he defrauds. Thus, once $X$ has defrauded a total of $a_X$, all of his links have been removed and he is prevented from participating in transactions in the future. Moreover, $X$ cannot use the "window of opportunity" (discussed in Section 3) to conduct fraud before feedback is provided, as Bazaar puts link weights on hold until the feedback is received.

Moreover, the same analysis holds for any subgraph or any cut in the network. Thus, collusion between malicious users does not help; the users can only defraud together for the total of what they could defraud separately. This argument also explains why creating fake identities also does not help, as it is the cut in the network between the user's identities and the rest of the network that bounds the amount that the user can defraud, instead of the number of identities the user has or the amount of fictitious feedback. The upshot is that Bazaar does not explicitly detect Sybil nodes or malicious users in the network, rather, it provides a strict guarantee on the amount of fraud that they are able to conduct.

The implication of this analysis is that we can characterize the amount of fraud the malicious users are able to conduct, in aggregate. Let us partition the network in two groups: $G$, containing non-malicious identities who do not conduct fraudulent transactions, and $M$, containing malicious identities whose goal is to defraud others. Let us consider the cut in the network between these two sets, with total value $c_{MG}$. We make two observations: First, any links that lie along this cut must represent non-fraudulent transactions between non-malicious users and malicious users; in essence, these represent instances where the malicious users were non-malicious. Second, any time one of the malicious users defrauds a non-malicious user, this cut is reduced by the amount of the fraud. Thus, malicious users can only defraud non-malicious users of up to $c_{MG}$ before the two groups are partitioned and all of the malicious users' transactions are flagged as potentially fraudulent to the non-malicious users.

It is worth noting that this is a much stronger guarantee than what can be provided today. For example, today, a user can potentially purchase a large amount of fictitious positive feedback with a low monetary investment, use that feedback to appear as an non-malicious seller, and then defraud users of a significant amount of money. This problem is exacerbated by the fact that the defrauded users have to realize that they have been defrauded before they can provide negative feedback and

warn others, leaving a significant window of vulnerability. Moreover, the malicious user can simply repeat this process with a new identity. By putting this bound in place, we are able to force the malicious user to participate in valid transactions with non-malicious users, thereby significantly reducing the attractiveness of committing such a fraud.

## 4.5 Discussion

We now discuss a few deployment issues with Bazaar.

**User interaction** The marketplace operator can use the output of Bazaar in multiple ways. For example, the marketplace operator can provide strong fraud guarantees by not allowing flagged transactions to go through. Alternatively, the marketplace operator can require that flagged transactions use an escrow service or insurance service, or can more closely scrutinize the transaction. The latter options represent an additional incentive for the marketplace operator to deploy Bazaar, as selling additional services such as escrow or insurance may increase their revenue while at the same time attracting customers due to a decrease in fraud.

**Providing honest feedback** An additional concern is whether buyers are incentivized to provide honest feedback on transactions in Bazaar. First, rational buyers have no incentive to provide incorrect negative feedback: By doing so, they penalize their own links and they prevent the creation of a new link between themselves and the seller. Since having more links is desirable (as it allows a user to participate in more and higher-valued transactions), buyers are disincentivized from providing incorrect negative feedback. Second, rational buyers also have no incentive to provide incorrect positive feedback. In particular, if they were unhappy with the transaction, providing positive feedback creates a new direct link to the seller; this is likely to be highly undesirable if the buyer felt defrauded, as it risks the buyer's existing links.

**Targeted attacks** Another possible concern is whether Bazaar introduces a new attack vector by allowing a malicious user to conduct a targeted attack on a seller by purchasing their goods and then always providing negative feedback (thereby damaging the seller's reputation). First, such an attack is possible in existing marketplaces, as malicious users can conduct this attack by creating numerous free identities and then purchasing the victim's goods. Thus, Bazaar does not open up a new avenue for attack. Second, we note that Bazaar raises the bar on this attack, making it more difficult to conduct: With today's marketplaces, the malicious users can purchase the victim's goods immediately after creating another identity. With Bazaar, the malicious users must first conduct non-fraudulent transactions in order to obtain enough links to be able to conduct the attack, making such an attack significantly more difficult and less attractive.

**Compromised accounts** If a user's account password is compromised, an attacker can conduct fraudulent transactions on the user's behalf, eventually causing the user to run out of links. However, this attack is not unique to Bazaar, since attackers could conduct the same attack with the reputation systems in-use today. Moreover, with Bazaar, the amount of fraud that can be conducted is still subject to the Bazaar bounds, whereas without Bazaar, it is potentially unbounded.

**Protecting sellers** Bazaar, as described so far, focuses on protecting buyers from being defrauded by malicious sellers who manipulate their reputation. However, in certain marketplaces, it may be necessary to protect sellers as well (e.g., from buyers who use fraudulent payment mechanisms like stolen credit cards). We leave protecting sellers to future work, with one comment: The need to protect sellers is somewhat mitigated by the fact that marketplace operators generally allow sellers to verify payment before shipping the good.

**Maintaining full network knowledge** The design of Bazaar proposed so far requires knowledge of the complete risk network. This is not an unreasonable assumption, as online marketplaces are generally run by a single operator that has full knowledge of all transactions. Given this information, the marketplace operator can create and update the risk network as necessary. It may be possible to decentralize knowledge of the risk network, but this remains an open research question and is a subject of future work. A decentralized system has several advantages with regards to privacy and scalability, but as we do not know of any decentralized online marketplaces, the path to deploy a decentralized solution is unclear.

# 5 Calculating max-flow using multi-graphs

The Bazaar design described so far relies on finding the max-flow path between two nodes in order to calculate the amount of risk embedded in a potential transaction. Since the risk network may have large number nodes and links, finding the max-flow between nodes using traditional approaches like Ford-Fulkerson [8] and Goldberg-Rao [9] may prove to be expensive. Similarly, pre-computing max-flow values through techniques like Gomory-Hu Trees [12] may also prove too costly, and are complicated by the fact that the risk network is changing over time. Instead, Bazaar uses a novel approach called *multi-graphs* in order to reduce the computation required. In this section, we first describe useful observations on risk networks and of our desired max-flow al-

gorithm, detail the multi-graph data structure, and finally demonstrate how multi-graphs reduce the complexity of finding max-flow values.

## 5.1 Observations

We begin by making two observations concerning the risk networks in online marketplaces and the properties of the max-flow calculation in Bazaar.

1. **Dense core** First, like social networks [16], the risk networks we observe in real-world online marketplaces tend to have a dense core, meaning a small minority of users possess the majority of the links. Moreover, the higher-valued links (representing risk relationships with higher values) also tend to fall in this "core." As a result, the risk network tends to shrink rapidly if links with less than a specified weight are discarded. We demonstrate this with real-world data in the following section.

2. **Actual max-flow not needed** Second, and most important, Bazaar does not need to actually calculate the value of the max-flow between a potential buyer and seller. Instead, Bazaar simply needs to verify whether the max-flow is above a certain value (i.e., the value of the potential transaction). This implies that the complexity of calculating the max-flow in Bazaar may not be as high as a general max-flow calculation.

The multi-graph optimization, described next, leverages both of these observations in order to reduce the complexity of the max-flow calculation in Bazaar.

## 5.2 Multi-graphs

Formally, we define a multi-graph $M$ to be a set of graphs

$$M = \{G_0, G_1, ..., G_n\}$$

where each graph $G_i = (V_i, E_i)$. These graphs are related: First, $G_0$ is defined to be the entire risk network. Second, $G_i$ is defined to be the subgraph of $G_{i-1}$ with

$$\begin{aligned} E_i &= \{e \in E_{i-1} : w(e) \geq k^i\} \\ V_i &= \{v : (v, \cdot) \in E_i\} \end{aligned}$$

where $w(e)$ represents the weight of edge $e$ and $k$ is a configurable system parameter with a suggested value of 2. Thus, the multi-graph contains a series of risk networks, where each subsequent network is a subgraph of the previous containing only those links with an exponentially higher weight. An example of converting a risk network into a multi-graph is shown in Figure 4.

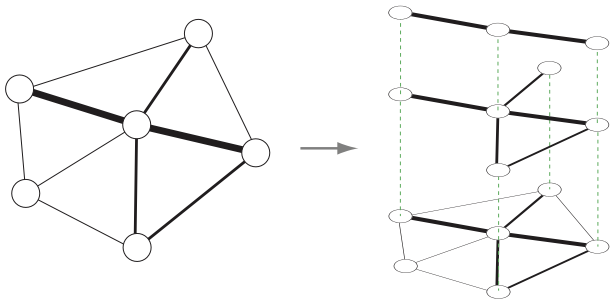

Figure 4: Conversion of a risk network (left) to a risk multi-graph (right). Links with higher weights are shown with thicker lines. Graphs at higher levels in the multi-graph only include links with exponentially increasing weights (e.g., with $k = 2$, the three levels of the multi-graph would represent all links, links with weight \$2 and higher, and links with weight \$4 and higher).

Note that a multi-graph contains multiple copies of a given link, the weights of which need to be kept consistent. There are three operations on the risk network under which Bazaar must maintain consistency:

- **Link addition** When a new link is added, it is simply added to all of the graphs to which it belongs (e.g., if the link weight is $w$, the link is added to $\{G_i : w \geq k^i\}$).
- **Link weight change** When the weight of a link is changed, it is simply added to or removed from the appropriate graphs. Conceptually, this can be viewed as removing the link from all graphs, followed by adding it back at its new value.
- **Link weight temporary adjustment** Recall that Bazaar may temporarily lower the weight of a link when a transaction is in progress. Conceptually, this can be viewed as changing the weight of the link. Later, if the adjustment is undone, this can again be viewed as a weight change.

## 5.3 Max-flow on multi-graphs

Now, let us consider what happens when Bazaar calculates whether a path set of total weight $w$ exists between a source and destination. With a normal risk network, Bazaar must use an algorithm like Goldberg-Rao, which runs over the entire risk network and is optimized to determine the actual max-flow between the source and destination.In contrast, with a multi-graph, Bazaar proceeds by first finding the highest-weight network $G_m$ where both the source and the destination are present. Then, Bazaar runs any existing max-flow algorithm on $G_m$, looking for a set of paths of collective weight $w$. If such a

set is found, then the algorithm returns that set and is finished. If no such set is found, Bazaar repeats the process with the next-lowest graph $G_{m-1}$. This process continues until either a set of paths of weight $w$ is found, or Bazaar cannot find such a set of paths in the lowest graph $G_0$. The latter case indicates that the max-flow in the original risk network was lower than $w$, demonstrating that finding the max-flow in a multi-graph is guaranteed to have the same outcome as finding the max-flow in the original risk network.

It is worth noting that multi-graphs require an increase in storage costs, since multiple copies of many links must be stored. However, as we demonstrate in the evaluation, the storage requirements of the multi-graphs are modest and are easily met by today's computing hardware.

### 5.4 Benefit of multi-graphs

We now describe how the use of multi-graphs speeds up the max-flow calculation in Bazaar. Consider the case of a transaction of value $w$. First, because of observation 1 above, the sizes of the graphs $G_i$ decrease extremely rapidly as $i$ increases. Thus, running a max-flow algorithm over $G_i$ is significantly faster than running it over $G_{i-1}$. Second, because of observation 2, it is possible to modify the max-flow algorithm to terminate as soon as it finds a path set of weight $w$, instead of continuing to find the actual max-flow. For example, if we are using Ford-Fulkerson, only a few rounds may be are needed in order to find a set of paths of weight $w$. Third, the increasing link weights in higher $G_i$ further reduce the running time of the max-flow algorithm, as the path set in higher $G_i$ is likely to consist of only a few paths. As we demonstrate in the evaluation, these effects allow multi-graphs to significantly speed up the calculation in practice.

## 6 Evaluation

In this section, we present an evaluation of Bazaar. In particular, we use data collected from a real-world online marketplace to determine if the max-flow technique employed by Bazaar is able to detect and prevent fraudulent transactions. We describe the data collected, verify our observations in the previous section, demonstrate the performance gains of using multi-graphs, and present an evaluation of Bazaar on real-world data.

### 6.1 Auction data

In order to evaluate Bazaar, we collect data from eBay, the largest online marketplace. We focus on collecting data from the `ebay.co.uk` site, containing United Kingdom auctions.

| Category | Purchases | Users | Avg. Price |
|---|---|---|---|
| Clothes | 3,311,878 | 1,436,059 | £9.73 |
| Collectibles | 940,815 | 454,773 | 8.90 |
| Computing | 964,925 | 661,285 | 21.31 |
| Electronics | 861,108 | 652,350 | 20.67 |
| Home/Garden | 2,795,795 | 1,426,785 | 16.57 |
| Total | 8,874,521 | 3,168,455 | £14.12 |

Table 1: Distribution and monetary values of feedback seen in our trace.

eBay makes the feedback for all users public. Each piece of feedback consists of the feedback value (positive, negative, or neutral), the auction the feedback was for, the identity of the user providing feedback, and a short message from that user explaining the feedback. Feedback can be provided by both the buyer and seller, so each auction can result in two pieces of feedback. eBay only makes detailed feedback available for 90 days, after which time, information about the auction the feedback is for is removed, and only the feedback value, message, and providing user remain. Thus, we are only able to collect detailed feedback for the previous 90 days.

eBay provides an API to collect data, but rate limits the requests to a very low rate. Instead, we use web scraping to collect data. We start from one user and crawl their feedback profile. From this profile, we learn about other users and proceed to crawl them. We continue this process until we exhaust all known users, effectively performing a breadth-first-search of the feedback graph.

In order to make our data collection process tractable, we only consider auctions and feedback that occur in five of the largest auction categories, shown in Table 1. Thus, we do not crawl other users that appear in the feedback history if the auction is not in one of these five categories. Since eBay allows users to participate in international transactions, not all users we discover are located in the United Kingdom. We restrict our crawl to only consider users located in United Kingdom, leaving us with a total of 3,168,455 distinct users (note that users may participate in multiple categories). Finally, because Bazaar focuses on protecting buyers from malicious sellers, we only collect feedback from buyers to sellers (and ignore feedback from sellers to buyers). In total, our dataset contains information on 8,874,521 items of feedback.

### 6.2 Dense core of risk networks

We now turn to validate our observation in Section 5 that motived our multi-graph design. Specifically, we examine whether there tends to be a dense "core" of users in the risk network, which was necessary for the multi-graph representation to have acceptable overhead. To do so, we use a similar approach to prior studies [16] and ex-

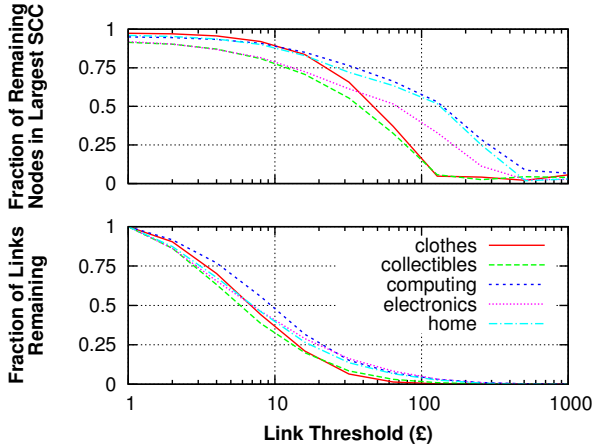


Figure 5: Fraction of links remaining (bottom) and fraction of the remaining nodes in the largest SCC (top) as only higher-weighted links are considered. Even as the majority of links are discarded, the largest SCC still contains most nodes, indicating the presence of a core.

amine the subgraph consisting of highly weighted links. We are interested in both the size and the connectedness of these subgraphs. Figure 5 shows how these two attributes vary as only higher-weighted links are considered. As the threshold rises from £1 to £20, almost 80% of the links are discarded. However, the vast majority of the remaining nodes are still in the largest strongly connected component (SCC), indicating the presence of a strong core. For some of the categories, the largest SCC does not disintegrate until only links of over £100 are considered. This validates our observation from the previous section, and indicates that multi-graphs are likely to speed up Bazaar's max-flow calculations in practice.

## 6.3 Multi-graph performance

We now turn to evaluate the benefits of using the multi-graph representation on the performance of finding max-flow paths. Specifically, we examine the tradeoff between memory and speed; since multi-graphs store multiple copies of certain links, they naturally have higher memory requirements than only using a risk network. First, we show the number of multi-graph levels and the resulting memory overhead, relative to the single graph, of storing a multi-graph in Bazaar in Table 2. As can be seen from the table, while the relative storage overhead is a 3- to 4-fold, the absolute overhead is small.

Next, we turn to evaluate the speedup of verifying whether a max-flow exists using a multi-graph in Bazaar. To do so, we create separate risk networks from each of the five categories by aggregating our feedback trace, creating links between users who participated in transactions with positive feedback. We then randomly select

| Category | Size (MB) | Levels | Overhead Rel. | Overhead Abs. (MB) |
|---|---|---|---|---|
| Clothes | 7.38 | 12 | 234.6% | 17.3 |
| Collectibles | 2.01 | 14 | 221.0% | 4.44 |
| Computing | 3.47 | 13 | 282.9% | 9.83 |
| Electronics | 3.23 | 13 | 255.9% | 8.25 |
| Home/Garden | 7.31 | 13 | 251.8% | 18.4 |

Table 2: Memory requirements of a single graph representation of the risk network, and number of levels and overhead (both relative and absolute) of a multi-graph representation, with $k = 2$.

1,000 pairs of nodes from each category and an amount from the prices in the observed auction trace. We calculate the time required to verify whether a set of paths exist with at least the selected auction amount between the pair of users. For this experiment, we used a machine with a 2.83 GHz Intel Xeon processor.

Table 3 presents the results of this experiment. Using the multi-graph representation shows a significant performance gain, with speed-ups ranging between 1.92× and 2.86×. In fact, with the multi-graph, most of the max-flow calculations take less than 6 seconds to complete. However, most of the calculations that are successful (e.g., a set of paths is found with at least the specified weight) finish quickly, while the calculations that eventually fail (e.g., no such set is found) take much longer to finish, thereby inflating the average. This trend is expected since a failure must traverse every graph in the multigraph, whereas a success has the potential to end early. This observation suggests a further avenue for speeding up the max-flow calculation in practice, by considering calculations that run longer than a specified amount of time to have failed. For example, in the Computing category, if all calculations that take longer than two seconds are considered to have failed, this would only misclassify 5.5% of the eventually to-succeed calculations, and would lower the average running time from 1.66 to 0.70 seconds.

Regardless, even without this further optimization, the average max-flow calculations in the largest category we examine (Clothes) required 6.29 seconds, meaning that 13,736 calculations could be completed per server per day. Using our trace, we determined that the highest number of auctions closing on a single day in this category was 80,846, meaning that Bazaar could be deployed in this category by purchasing a server with at least 6 cores. Of course, synchronization would need to be maintained to ensure that two cores were not using a single link at once. We observed, though, that such conflicts occur rarely (0.0165% of the time in this category), implying that parallelism of the max-flow algorithm [1] is likely to provide significant performance gains.

| Category | Time (s) Single | Multi-graph | Speedup |
|---|---|---|---|
| Clothes | 18.0 | 6.29 | 2.86× |
| Collectibles | 2.53 | 1.18 | 2.14× |
| Computing | 3.78 | 1.66 | 2.27× |
| Electronics | 2.71 | 1.41 | 1.92× |
| Home/Garden | 11.6 | 5.34 | 2.15× |

Table 3: Average max-flow calculation times, and relative speedup when using multi-graphs with $k = 2$.

| Category | Fraction of transactions incorrectly flagged |
|---|---|
| Clothes | 1.11% |
| Collectibles | 1.12% |
| Computing | 3.23% |
| Electronics | 4.68% |
| Home/Garden | 2.43% |

Table 4: Fraction of non-fraudulent transactions that are incorrectly flagged as fraudulent by Bazaar. The fraction flagged incorrect is never higher than 5%, indicating that non-malicious users are largely unaffected.

## 6.4 Detecting fraud with Bazaar

We now turn to examine how well Bazaar is able to detect fraudulent transactions. In particular, we are interested in three aspects of Bazaar's performance: First, what is the impact on non-malicious users? In other words, how often are non-malicious users' transactions incorrectly flagged as potentially fraudulent? Second, is Bazaar able to bound the amount of fraud that malicious users are able to conduct? Third, what impact, in terms of the amount of fraud prevented, could we expect from Bazaar if it were deployed on a online marketplace?

To conduct the evaluation, we use a random subset of 80% of the feedback data to create a risk network for each of the five categories, and then use the remaining 20% of the feedback data to simulate the operation of Bazaar. Because our data only represents a 90-day period, many of the users participate only in a single transaction (and therefore have a max-flow of 0 to all other users). In order to reduce the bias caused by our short time-window of data, we only simulate users who we observe to participate in at least five transactions during the time range. Finally, for each data point, we repeat the experiment 10 times using different random seeds.

To simulate Bazaar, we need a few pieces of information from each auction transaction: the identity of the buyer and seller, the price of the auction, the purchase and feedback time, and the feedback itself. Our crawled data unfortunately only contains the purchase time for 54.6% of the data.[5] So, for the auctions where the purchase time is not available, we artificially select a purchase time by subtracting a random "delay" from the feedback time. This delay is randomly drawn from the observed purchase-time-to-feedback-time delay distribution of the other auctions.

### 6.4.1 Impact on non-malicious users

Our first evaluation examines the potential negative impact that Bazaar has on non-malicious buyers and sellers. The primary form that such impact takes is incorrectly flagging transactions as potentially fraudulent. To determine the frequency with which this happens, we simulate Bazaar without any malicious users and calculate the fraction of transactions that had positive feedback but that would have been flagged by Bazaar due to insufficient max-flow. The results of this experiment are shown in Table 4, listing the fraction of non-fraudulent transactions which are flagged as potentially fraudulent by Bazaar. The results show that no more than 5% of all non-fraudulent transactions are flagged, indicating that non-malicious users in Bazaar are largely unaffected.

### 6.4.2 Blocking malicious users

We now evaluate whether Bazaar is able to bound the amount of fraud that malicious users can conduct in practice. Recall that Bazaar guarantees that each user is only able to conduct fraudulent transactions up to the amount of non-fraudulent transactions that he has participated in. Thus, we are interested in comparing how much fraud malicious users can conduct, relative to the amount of non-fraudulent transactions they participated in.

To simulate the behavior of malicious users, consistent with prior studies [22], we randomly select 1% of the users to be malicious. For each user, we simulate Bazaar running with other, randomly selected users purchasing items from the malicious user. We then calculate the total amount of fraudulent transactions that each user can conduct, until the point at which Bazaar flags all transactions with the malicious user as potentially fraudulent.

Figure 6 presents the results from conducting this experiment, by plotting the amount of fraudulent transactions a malicious user can conduct versus the sum of the malicious user's initial links. As can clearly be seen in the figure, Bazaar's bound on the amount of fraudulent transactions holds: the amount of possible fraud is strictly bounded by the sum of the non-fraudulent transactions that the malicious user has participated in so far.[6]

[5] In more detail, the purchase time of fixed-price auctions—where a user sells multiple, identical items at a fixed price—is not available, as these auctions have multiple buyers purchasing the items.

[6] A careful reader will note that malicious users are sometimes bounded to less than the actual total of their previous successful transactions. This occurs when, for example, a malicious user is the only

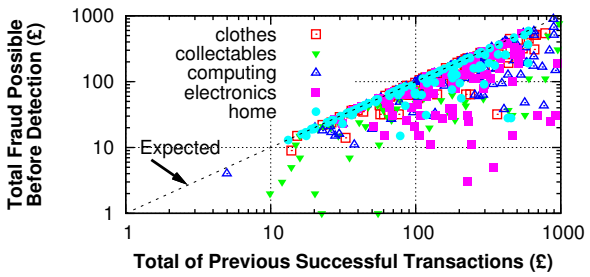


Figure 6: Aggregate amount of fraudulent transactions that malicious users can conduct versus the aggregate value of previous successful transactions. Also included is the expected bound ($y = x$). As expected, Bazaar ensures that malicious users can only commit fraud up the amount of successful transactions that they have participated in previously.

Even if the malicious user whitewashes his account (by creating a new identity), or conducts a Sybil attack (by creating multiple identities and linking them by fictitious transactions), he is unable to conduct any more transactions that are not flagged as potentially fraudulent.

#### 6.4.3 Preventing fraud

As a final point of evaluation, we examine the amount of fraud that Bazaar would prevent, were it to be deployed on a real-world online marketplace. In other words, what impact could we expect from Bazaar?

To evaluate this, we use the same 90-day trace from the five eBay categories. Then, for each seller, we calculate the total amount of goods sold with positive feedback, and the total with negative feedback. Recall that Bazaar prevents any user from having more (price weighted) negative feedback than positive feedback, so the auctions that represent the excess negative feedback would have been flagged as potentially fraudulent. We therefore calculate the total of this excess, and determine what fraction of the overall negative feedback it represents.

Table 5 presents the results. Bazaar would have flagged between 29% and 42% of all auctions that resulted in negative feedback as being potentially fraudulent, thereby possibly preventing these auctions from occurring. While we cannot say that all of these transactions represent fraud (e.g., the negative feedback could simply represent buyer's remorse), the fact that these all come from sellers whose weighted negative feedback is greater than their weighted positive feedback strongly suggests so. In total, the auctions that Bazaar would have prevented represent £164,791.55 worth of goods, significantly bolstering the reliability of the online marketplace. Moreover, this amount is only for a 90-day period in the five categories we study; the amount is likely to be significantly higher if Bazaar were deployed on the entire marketplace and over a longer period of time.

user that another user is linked to: Even though the malicious user's total is increased, this link does not increase the max-flow to any other users (much in the manner of the $X_2 ... X_5$ identities in Figure 3).

| Category | Total flagged | Fraction of all negative feedback |
|---|---|---|
| Clothes | £28,291.34 | 29.9% |
| Collectibles | 4,995.04 | 38.2% |
| Computing | 48,742.66 | 39.7% |
| Electronics | 34,476.87 | 42.6% |
| Home/Garden | 47,285.64 | 32.4% |
| Total | £164,791.55 | 36.0% |

Table 5: Total number of auctions with negative feedback that would be flagged as potentially fraudulent, and the fraction of all auctions with negative feedback that this represents. Overall, Bazaar would have flagged £164,791.55 worth of auctions that eventually resulted in negative feedback, representing 36% of all such auctions.

## 7 Related work

Researchers have previously studied approaches to detecting auction fraud, usually relying on machine-learning techniques [4, 18] based on bidding behavior. While these techniques succeed at detecting some fraudulent users, they rely on characteristics of malicious behavior. As a result, unlike Bazaar, these approaches do not provide a bound on the amount of fraud any user can conduct. Additionally, researchers have developed techniques [14, 21] to detect shill bidding, where users conspire with others to artificially inflate the selling price of their auctions. Bazaar is complementary to this work, as it is not concerned with shill bidding, but rather, fraud caused by reputation manipulation.

Other work [5, 10] has examined building reputations based on social relationships between users. While some of the techniques used are similar to Bazaar, Bazaar must determine pairs of trusting users itself (instead of assuming pairwise trust is externally provided). This introduces significant challenges, but enables Bazaar to be deployed on existing sites.

There is also significant work that studies the network formed by users who trust each other, and a number of research systems have already been proposed to leverage this trust. Perhaps the most well known of these are the PGP web of trust [27] and the Advagato trust metric [2]. However, these systems are generally concerned with providing a stronger notion of identity, instead of bounding the amount of malicious activity.

More generally, recent work has focused on detecting Sybil accounts using social networks [6, 25, 26]. These

approaches are not directly applicable to online marketplaces for two reasons: First, they assume the existence of a social network that is not necessarily present, and second, they only bound the number of Sybil accounts that are admitted, not on the amount of fraud that malicious users can conduct. Thus, even with Sybil detection algorithms, malicious users are still able to conspire to arbitrarily inflate each others' reputations.

Like other work [22], Bazaar uses a mechanism that is loosely based on the one used in Ostra [17], a system that uses a social network to block senders of unwanted communication. However, Bazaar differs from Ostra in three important ways. First, while Ostra is based on a relatively stable, unweighted social network, Bazaar uses a weighted risk network that is changing with every transaction (e.g., links are added and removed, and the links weights can grow and shrink over time). Second, Ostra assumes the trust network is given from an external source, while Bazaar constructs the risk network during the operation of the system. This requires Bazaar to face additional challenges, as malicious users are able to create links by participating in transactions (this is not possible in Ostra, as Ostra's assumption is simply that links to non-malicious users take effort to form and maintain). Third, Bazaar works by calculating the max-flow in the risk network, instead of simply finding a single path (as in Ostra). This induces significant engineering challenges and results in a system with a different set of guarantees.

## 8 Conclusion

In this paper, we presented Bazaar, a system that strengthens user reputations in online marketplaces. Bazaar is based on max-flow calculations over a risk network, a data structure that encodes the amount of rewarded shared risk between participants. Using data on over 8 million purchases from a real-world online marketplace, we demonstrated that Bazaar is able to effectively bound the fraud that malicious users are able to conduct, while only rarely impacting the transactions conducted between non-malicious users.

Given the popularity of online marketplaces and the large amount of fraud that such marketplaces currently experience, our hope is that Bazaar can be used as a drop-in component on real-world sites. Bazaar is designed to be readily applied to such marketplaces.

## Acknowledgements

We thank the anonymous reviewers, Peter Druschel, Lakshmi Subramanian, Bimal Viswanath, and our shepherd, Dina Katabi, for their helpful comments. This research was supported in part by NSF grant IIS-0964465 and an Amazon Web Services in Education Grant.

## References

[1] R. J. Anderson and J. Setubal. On the parallel implementation of Goldberg's maximum flow algorithm. *SPAA*, 1992.

[2] Advagato Trust Metric. `http://www.advogato.org/trust-metric.html`.

[3] Amazon Merchants and Marketplace. `http://www.amazonservices.com/content/sell-on-amazon`.

[4] D. H. Chau, S. P, and C. Faloutsos. Detecting fraudulent personalities in networks of online auctioneers. *PKDD*, 2006.

[5] D. DeFigueiredo and E. T. Barr. TrustDavis: A Non-Exploitable Online Reputation System. *CEC*, 2005.

[6] G. Danezis and P. Mittal. SybilInfer: Detecting Sybil Nodes using Social Networks. *NDSS*, 2009.

[7] J. Douceur. The Sybil Attack. *IPTPS*, 2002.

[8] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Can. J. Math.*, 8, 1956.

[9] A. Goldberg and S. Rao. Flows in Undirected Unit Capacity Networks. *FOCS*, 1997.

[10] A. Ghosh, M. Mahdian, D. Reeves, D. Pennock, and R. Fugger. Mechanism Design on Trust Networks. *WINE*, 2007.

[11] D. G. Gregg and J. E. Scott. The Role of Reputation Systems in Reducing on-Line Auction Fraud. *Int. J. Elec. Comm.*, 10(3), 2006.

[12] R. E. Gomory and T.C. Hu. Multi-Terminal Network Flows. *SIAM*, 9(4), 1961.

[13] D. Houser and J. Wooders. Reputation in Auctions: Theory, and Evidence from eBay. *Econ. Strat.*, 15, 2006.

[14] R. J. Kauffman and C. A. Wood. Running up the bid: Detecting, predicting, and preventing reserve price shilling in online auctions. *ICEC*, 2003.

[15] D. Lucking-Reiley, D. Bryan, N. Prasad, and D. Reeves. Pennies from eBay: The determinants of price in online auctions. *Indus. Econ.*, 55(2), 2007.

[16] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and Analysis of Online Social Networks. *IMC*, 2007.

[17] A. Mislove, A. Post, K. P. Gummadi, and P. Druschel. Ostra: Leverging trust to thwart unwanted communication. *NSDI*, 2008.

[18] S. Pandit, D. H. Chau, S. Wang, and C. Faloutsos. Netprobe: a fast and scalable system for fraud detection in online auction networks. *WWW*, 2007.

[19] P. Resnick and R. Zeckhauser. Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System. *The Economics of the Internet and E-Commerce*, volume 11, Elsevier Science, 2002.

[20] B. Sullivan. Man arrested in huge eBay fraud. 2003. `http://www.msnbc.msn.com/id/3078461/`.

[21] H. S. Shah, N. R. Joshi, A. Sureka, and P. R. Wurman. Mining eBay: Bidding strategies and shill detection. *WebKDD*, 2002.

[22] N. Tran, B. Min, J. Li, and L. Subramanian. Sybil-Resilient Online Content Voting. *NSDI*, 2009.

[23] L. von Ahn, M. Blum, N. Hopper, and J. Langford. CAPTCHA: Using Hard AI Problems for Security. *EuroCrypt*, 2003.

[24] J. Weaver. How a bold eBay scam was tracked to South Florida. *The Miami Herald*, 2010.

[25] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao. SybilLimit: A Near-Optimal Social Network Defense Against Sybil Attacks. *IEEE S&P*, 2008.

[26] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. SybilGuard: Defending Against Sybil Attacks via Social Networks. *SIGCOMM*, 2006.

[27] P. R. Zimmermann. *The Official PGP User's Guide*. MIT Press, 1994.