Exploring Reactive Access Control

Michelle L. Mazurek[⋆]
mmazurek@andrew.cmu.edu

Peter F. Klemperer[⋆]
klemperer@cmu.edu

Richard Shay[⋆]
rshay@cmu.edu

Hassan Takabi[†]
hatakabi@sis.pitt.edu

Lujo Bauer[⋆]
lbauer@cmu.edu

Lorrie Faith Cranor[⋆]
lorrie@cs.cmu.edu

[⋆]Carnegie Mellon University
Pittsburgh, PA, USA

[†]University of Pittsburgh
Pittsburgh, PA, USA

ABSTRACT

As users store and share more digital content at home, access control becomes increasingly important. One promising approach for helping non-expert users create accurate access policies is *reactive policy creation*, in which users can update their policy dynamically in response to access requests that would not otherwise succeed. An earlier study suggested reactive policy creation might be a good fit for file access control at home. To test this, we conducted an experience-sampling study in which participants used a simulated reactive access-control system for a week. Our results bolster the case for reactive policy creation as one mode by which home users specify access-control policy. We found both quantitative and qualitative evidence of dynamic, situational policies that are hard to implement using traditional models but that reactive policy creation can facilitate. While we found some clear disadvantages to the reactive model, they do not seem insurmountable.

Author Keywords

Access control, Home computing, Human factors, Privacy

ACM Classification Keywords

H.1.2 User/Machine Systems: Human factors; D.4.6 Security and Protection: Access controls; H.5.2 User Interfaces: User-centered design

General Terms

Experimentation, Human Factors, Security

INTRODUCTION

Users without technical expertise are accumulating content on their home devices. Increasing network capacities facilitate sharing this data, increasing the risk of unauthorized information disclosure. Access-control mechanisms are needed to ensure these users maintain control of their data. However, providing this access control effectively is a challenge. Managing access-control policy has historically been time consuming and error-prone, even for experts [9, 16]. Non-expert home users may lack the time, patience, and experience needed to specify correctly a traditional access-control policy covering hundreds of files.

One promising solution is *reactive policy creation*. In a reactive model, resource owners are not required to determine all access-control policies a priori, but may instead do so in response to a request. If a user tries to access a resource but lacks sufficient permission, she can use the access-control system to send a request to the resource owner, who can opt to update his policy and allow the access.

Ad-hoc reactive access control is frequently used in practice. When a person finds herself unable to access a given file, she may contact the file owner to ask permission. However, in most cases reactive policy creation is not supported directly by access-control systems. Instead, users go outside the system and make requests via e-mail or telephone calls. This disconnect between traditional system affordances and user behavior motivates our study; we believe that shifting these requests from external channels to within the system may provide valuable benefits in convenience, security, oversight, and accountability.

Further, we are motivated by a prior study that suggested reactive access control might help non-technical home users more easily manage their access-control policies [10]. Participants in that study responded positively to the idea of reactive policy creation; for many, a system based on requests seemed like it would provide a greater sense of control over their data. Moreover, it fit within the familiar social convention of asking for permission. However, that study asked participants to consider a reactive system only briefly.

In this study, we set out to discover in more depth whether and how a reactive model could contribute to making access control more usable. In particular, we wanted to know whether people have specific policy needs that match better to a reactive model than a traditional model, as well as whether reactive policy creation better matches users' mental models and preferences. We also wanted to know whether

CHI 2011, May 7–12, 2011, Vancouver, BC, Canada.
Copyright 2011 ACM 978-1-4503-0267-8/11/05...$10.00.

responding to requests would prove so tedious or annoying that the reactive model would be impractical. For this study, we chose to focus on how well the reactive model could work for file owners; we leave examining reactive policy creation from a requester's point of view to future work.

To address these questions as realistically as possible without building an actual reactive system, we designed and executed an experience-sampling study intended to simulate the experience of using a reactive policy creation system to manage file access. Our 24 participants provided names of files they have and people they know. For one week, each participant received and responded to simulated access requests drawn at random from these lists of people and files.

We collected a rich set of data that bolsters the case for using reactive policy creation as one of the modes by which home users specify file-access-control policy. We found quantitative and qualitative evidence of dynamic, situational access-control policies that are hard to implement using traditional models but that reactive policy creation can facilitate. Our study showed that the reactive model supports many of our participants' policy creation needs, including the desire for more control and interactivity. While we found some clear disadvantages to the reactive model, they do not seem insurmountable. In fact, we found that some seemingly obvious disadvantages, such as the annoyance of receiving frequent requests, had only a minor impact on the usability of our simulated system and on user satisfaction. In the process, we captured detailed information about the policy decisions users made and how they made them. Our study also served as a very low-fidelity prototype, providing insight into effective interface design for incorporating reactive policy creation into the access-control system we plan to build.

In the remainder of this paper, we describe the methodology of our study, present an overview of our results, describe the trends we identified in more detail, discuss the limitations of our approach, review related work, and present our conclusions.

METHODOLOGY

We recruited 24 adult participants, using craigslist advertisements and fliers posted at our universities. We conducted the study in two rounds, one month apart, with 10 and 14 participants respectively. To focus on non-experts, we limited participation to those without degrees or jobs in computer science or engineering. Table 1 lists demographic information about our participants. Participants were compensated $10 for the initial briefing interview, $15 for the debriefing interview, and 25¢ for each response to a reactive request.

We encountered a large gender disparity during recruitment: the first 10 volunteers who met our requirements were all women. To counter this, we performed a second round of interviews with nine men and five women. While this may affect our results, we believe the effect is small. We discovered no major differences between the participants in the first and second round, apart from expected changes in the use of *maybe* policies, described below.

Code	*Age*	*Gender*	*Occupation*	*Household*	*Condition*	*Files*	*Requesters*
P01	23	F	marketing manager	R	pre	24	10
P02	41	F	magazine editor	A	post	24	9
P03	26	F	law student	R	pre	20	8
P04	32	F	unemployed	C	post	22	10
P05	25	F	law student, legal secretary	A	post	20	11
P06	23	F	student	R	pre	19	10
P07	24	F	graduate student	A	post	20	9
P08	43	F	unemployed	A	pre	9	9
P09	29	F	student	R	pre	26	9
P10	46	F	video producer	C	post	25	9
P11	34	M	legal assistant	F	pre	26	8
P12	23	F	student	F	post	20	8
P13	23	F	student	C	pre	19	8
P14	22	F	student	R	pre	20	8
P15	37	F	business administrator	F	post	24	11
P16	23	F	product developer	C,R	post	25	8
P17	26	M	unemployed	R	pre	21	9
P18	37	M	HR manager	F	post	22	9
P19	34	M	lawyer	C	post	14	8
P20	23	M	marketing coordinator	C,R	pre	20	10
P21	54	M	purchasing manager	F	post	20	8
P22	21	M	student	R	pre	15	8
P23	24	M	bookkeeper	R	post	23	8
P24	26	M	entertainment	F	post	26	8

Table 1. Participant demographics. For household, R indicates roommates, F families, C couples, and A participants who live alone.

Experience-sampling process

We modeled our experience-sampling study on a location-sharing study by Consolvo et al. [5]. Our study included an initial briefing interview, a request phase, and a final debriefing interview. We used two conditions: *pre*-condition participants filled out a grid representing their proactive policy during the briefing interview, while *post*-condition participants filled out this grid during the debriefing interview.

Briefing interview.

We collected lists of eight to 11 people with whom the participant might share files. Participants were required to list anyone they live with, a romantic partner if applicable, at least two family members and two friends not living with the participant, a supervisor, and at least two work colleagues or fellow students.

Participants were also asked to name files they store on digital devices. We prompted participants to think about photos, music, videos, financial files, work or school files, e-mail, and address-book information. In each category, we asked for groups of files, then one or two examples per group. We obtained 14 to 26 file names per participant.

Pre-condition participants only were then asked to define a *yes*, *no*, or *maybe* access policy for each of the file/requester combinations. We call this the *proactive* policy, because it is established before any hypothetical access attempts. The *maybe* policy indicates the participant cannot or does not want to make a decision without more information.

During the first round, participants appeared to have difficulty comprehending the *maybe* policy. To address this, we read a more detailed description of the *maybe* policy to second-round participants and asked them to provide and explain examples of *yes*, *no*, and *maybe* policy choices before completing the grid on their own. As we expected, this change led to increased use of *maybe* policies in the second round, mainly in exchange for decreased use of *no*. No other significant differences were found between the first and second rounds in the Likert questions, proactive grid policies, or reactive policies, except for a slight increase in participants who said they might use a reactive system to request files.[1] As a result, we consider the effect of conducting the study in two rounds to be small.

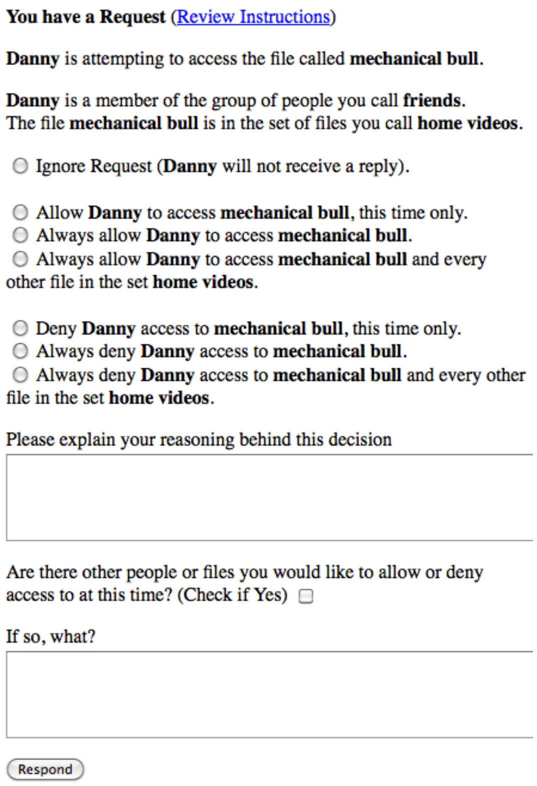
Request phase.

In the second phase, participants received mock file-request e-mails indicating a particular person wanted to access a particular file. The people and files were randomly selected, using a uniform distribution, from the lists provided during the briefing; combinations were not repeated. Each participant received five to 15 requests per day between 7 a.m. and 11 p.m., during a six- to seven-day period, with the exact number and timing also randomly selected with uniform distribution. All requests were simulated; none of the participants' acquaintances were contacted, nor were their files actually shared. Requests were assigned randomly to examine a broad range of requester/file combinations, including those that participants might find strange or uncomfortable.

Each request directed the participant to a website where she could select a response from seven options, as illustrated in Figure 1. The options allowed the participant to *ignore*, *allow*, or *deny* the request. *Allow* and *deny* responses could be set for that request only, all future requests from that person for that file, or all future requests from that person for that file group. The participant was also asked to supply a short explanation of her reasoning, intended specifically for the researchers and not the hypothetical requester. In the second round, participants were also able to provide an optional free-form description of additional policies they would like to create.

Participants' persistent reactive policy decisions did not feed back into requests they received later. As a result, requests could be inconsistent with participants' previous policy decisions (e.g., a participant might receive a request from someone they had previously stated they wished to allow all requests from). A more complete system would have automatically screened out inconsistent requests.

We asked participants to imagine the requests were real when responding. We asked them not to check their e-mail more often or respond sooner than they might if the requests were real, and to consider any factors that might influence their answer to a real request. During the debriefing interviews, we found that for the most part our participants immersed themselves in the system and demonstrated strong, sometimes emotional reactions to the simulated requests.



You have a Request (Review Instructions)

Danny is attempting to access the file called **mechanical bull**.

Danny is a member of the group of people you call **friends**.
The file **mechanical bull** is in the set of files you call **home videos**.

○ Ignore Request (**Danny** will not receive a reply).

○ Allow **Danny** to access **mechanical bull**, this time only.
○ Always allow **Danny** to access **mechanical bull**.
○ Always allow **Danny** to access **mechanical bull** and every other file in the set **home videos**.

○ Deny **Danny** access to **mechanical bull**, this time only.
○ Always deny **Danny** access to **mechanical bull**.
○ Always deny **Danny** access to **mechanical bull** and every other file in the set **home videos**.

Please explain your reasoning behind this decision

Are there other people or files you would like to allow or deny access to at this time? (Check if Yes) □

If so, what?

Figure 1. Sample response form.

Participants responded to 1360 of 1452 total requests sent to them. The responses represented 30% coverage of 4481 possible requester/file combinations, with minimum coverage of 19% and maximum coverage of 45% per participant.

Debriefing interview.

At the start of the debriefing interview, post-condition participants only were asked to define a proactive *yes*, *no*, or *maybe* access policy for all of the file/requester combinations they had defined in the briefing.

All participants then completed a seven-question survey. Using a seven-point Likert scale from 'strongly agree' (1) to 'strongly disagree' (7), we asked how enjoyable or convenient they found the system, and whether they would consider using such a system in real life. We also asked whether they found the number of messages they received annoying.

We then asked open-ended questions about their experience. We asked for which people and files they particularly did or did not want to use a reactive system, and how well the response options met their needs. We also asked how realistic they found the requests.

Then, we discussed in detail several individual requests, cho-

[1] All differences calculated using unpaired t-tests. Significant p-values (<0.05): Likert *use to request*, p=0.0285; proactive *no*, p=0.0296; proactive *maybe*, p=0.000284. Non-significant p-values: Likert *convenience*, p=0.983; Likert *annoyance*, p=0.741; Likert *enjoyable*, p=0.272; Likert *use to share*, p=0.788; proactive *yes*, p=0.508; reactive *allow*, p=0.0884; reactive *deny*, p=0.125; reactive *ignore*, p=0.803.

sen to provide broad coverage and include particularly interesting cases. For each selected request, participants were asked to explain why they answered as they did and whether they would choose the same answer again. Where applicable, we asked about any social awkwardness that could arise from denying or ignoring a request.

Finally, we asked participants whether they would prefer to create policy all at once, up front; to use a reactive system; or to use a combination of the two, and why.

Grid conditions.

As discussed earlier, participants were divided into two conditions. Participants in the first condition filled in a proactive access policy grid during the initial interview, while participants in the second condition completed the same grid during the debriefing. This grid allowed us to contrast the participants' proactive policies with the reactive policies created by responding to requests, and provided participants with a clear point of comparison between setting policy all at once and setting policy on a per-request basis. We used the conditions to examine priming effects on participants who were required to think through every possible person-file combination by filling out a grid before responding to requests.

In both cases, we emphasized to participants that we were not asking them to match their grid and request responses, and that changing their minds was normal and allowed. We cannot completely account for the fact that participants may have tried to match their answers to appear more consistent, but we attempted to minimize its effects.

Data analysis

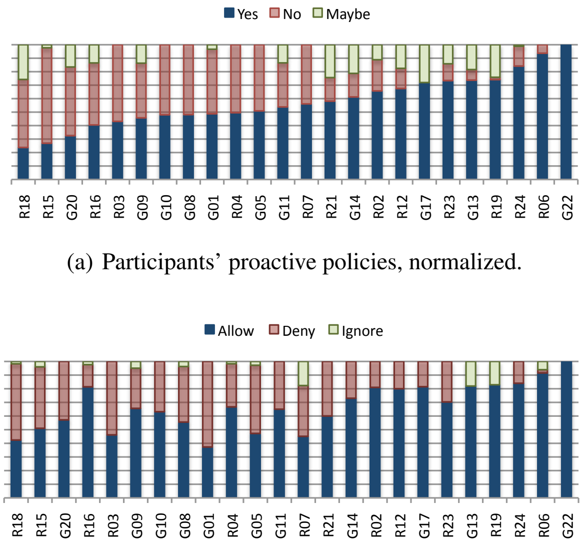
We collected both quantitative and qualitative data. To evaluate participants' free-form responses to debriefing questions, we applied an iterative coding process [14, 15]. An initial fine-grained review of a subset of the debriefing survey responses resulted in an initial set of codes. Then two of the authors independently coded the participant responses. Disagreements were resolved through modification or clarification of the codes until consensus was reached.

Several participants provided no policy choice for some grid cells. To facilitate analysis, only files with complete grid policy definitions are included in this paper. One participant failed to complete an entire page of grid policy, resulting in 97% of the blank grid entries. The dropped entries account for approximately 2% of possible grid policies.

A few participants answered the same request more than once, creating 26 duplicate responses (less than 2% of total responses). In this paper, the first response provided by the participant for any requester/file combination is assumed correct and used for data analysis.

FINDINGS

In this section, we first provide an overview of quantitative results. We then discuss seven specific findings drawn both from these results and from qualitative data that we collected. The first four provide evidence that users' policy needs can be better met by reactive policy creation than by traditional models. The next three describe additional interesting, somewhat unexpected qualitative results that can guide effective design of reactive-policy-creation systems.



(a) Participants' proactive policies, normalized.

(b) Participants' reactive policies, normalized.

Figure 2. Participants' proactive and reactive policies varied widely.

Overview of quantitative results

Participants filled in a total of 4481 policy grid entries. Of these, 56% (2518) were *yes*, 34% (1518) were *no*, and 10% (445) were *maybe*. Individual choices ranged from 100% *yes* (P22) to 72% *no* (P15). Details of each participant's grid policies can be found in Figure 2(a).

Request responses showed a similar distribution, with 67% (913) *allow*, 30% (406) *deny*, and 3% (41) *ignore*. Individuals ranged from 100% *allow* (P22) to 62% *deny* (P01). Participants used *allow once* 251 times, *always allow file* 281 times, and *always allow group* 381 times. *Deny* responses were divided into 108 *deny once*, 81 *always deny file*, and 217 *always deny group*. Figure 2(b) summarizes individual participants' responses.

We also measured conflicts between participants' grid entries and their responses to requests. We consider instances where the participant marked *yes* but responded with any type of *deny*, or marked *no* but responded with any type of *allow*, as conflicts. Overall, 12% of responses resulted in conflicts, ranging from several participants with no conflicts to P08, for whom 49% of responses resulted in conflicts.

We found no significant difference in grid or response patterns between pre- and post-condition participants.[2]

[2] Differences calculated using unpaired t-tests, with p-values as follows: proactive *yes*, p=0.952; proactive *no*, p=0.972; proactive *maybe*, p=0.967; reactive *allow*, p=0.808; reactive *deny*, p=0.704; reactive *ignore*, p=0.619.

During the debriefing interviews, participants answered several Likert questions about their experiences. When asked whether they enjoyed using the system, the median response was a moderately positive 5 out of 7 (66.7% agree, 16.7% disagree, 16.7% neutral). Participants also agreed (median 6 out of 7; 66.7% agree, 12.5% neutral, 20.8% disagree) that this type of system would be convenient for them. Asked whether they found the e-mail requests annoying, participants disagreed slightly (median 3 out of 7; 25% agree, 16.7% neutral, 58.3% disagree).

We also asked participants whether they would use a similar system in real life, both to share their own files and to request files from others. Eleven participants said they probably would use such a system, nine said they might or might not, and only four said they probably would not.

In addition, we asked participants whether they would prefer reactive policy specification (represented by the request system), proactive policy specification (represented by the grid they filled out), or a combination of the two. Seven participants preferred the reactive model and 14 preferred the combination; only one preferred proactive policy alone.

During the second round, we asked participants if they would rather receive requests via e-mail, text message, phone call or other means. They overwhelmingly preferred e-mail, often in combination with text messages. A few participants asked for requests to be aggregated, either in a periodic digest e-mail or via a web-based service the participant could check at her convenience.

Policies change over time

Our results indicate that participants' file-sharing policies change relatively often, in response to a variety of factors. Reactive policy creation is better suited to express these kinds of dynamic policies than traditional access-control models. Participants demonstrated this dynamism through their proactive and reactive policy choices, including use of *maybe*, one-time decisions, and policy conflicts.

One indicator of dynamic policy was the frequent use of *maybe* in participants' policy grids, both before and after using the request system. As mentioned previously, in the second half of the study, when we clarified the *maybe* option, the use of *maybe* rose from 3% of grid entries in round one to 15% of grid entries in round two. Second-round participants used *maybe* more often at the expense of *no*; the rate of using *yes* stayed roughly unchanged. Especially in round two, participants often used *maybe* in thoughtful, deliberate ways in cases where context was important and access policy could not be entirely specified in advance. For example, P21 used *maybe* in reference to sharing financial data from his son's business with some of his co-workers. He never expected to share that file with those people, but said he might make an exception if an accountant wanted to provide financial advice. He chose *maybe* rather than *no*, planning to make an informed decision in context. P17 never used *no* in his grid, preferring *maybe* for the same reason.

Proactive *yes*, reactive *deny*	
Number of yes-deny conflicts	64
As percent of all conflicts	38.1
As percent of all responses	4.7
Proactive *no*, reactive *allow*	
Number of no-allow conflicts	104
As percent of all conflicts	61.9
As percent of all responses	7.7
Total conflicts	
Number of conflicts	168
As percent of all responses	12.4

Table 2. Conflicts between proactive and reactive policy. Most conflicts occurred when a proactive *no* decision was overturned by a reactive *allow* decision.

As with *maybe*, participants used the one-time *allow* and *deny* options to express policies that were expected to change. P10 selected *deny once* rather than *always deny* for one request because "it is within the realm of possibility that something would alter" and she would change her mind. P23 denied a work colleague access to a photo of him with friends once, after marking that combination maybe in the grid. He explained that he might grant access "maybe if we were just having casual talk at work and I mentioned something we did over the weekend."

In many cases, differences between proactive and reactive policy also indicated changing policy preferences. We found that 12% of total request responses conflicted with the participants' grid entries. Although a few of these conflicts were generated by participant misclicks in the response form, most reflect actual policy changes. Table 2 details the conflicts we observed. Interestingly, 62% of all conflicts involved answering *no* in the grid but allowing a request, regardless of whether the participant filled out the grid first or answered the request first. This provides some evidence that people will share more reactively than proactively.

P05 approved an access on her grid, but refused the same access as a request because "today he's on my blacklist." P12 said, "Some files might be consistent all the time," while others "depend on [the requester's] reasoning or might depend on my mood." P02 refused a request for a work document in progress, but later marked yes in her grid because the document had since been completed. (This is consistent with Razavi's findings about sharing patterns over document life cycles [13].)

Policies are situational

Our results indicate that participants' policies are dynamic in part because their sharing decisions depend heavily on the details of the situation at the time the access-control decision is made. Again, this context-awareness is a natural fit for a reactive model, which allows users to make decisions at the relevant time rather than a priori.

Participants frequently explained that the reason why someone wanted to access a file mattered in making policy. P04 denied a request from a friend for video of a family wedding because "there's no reason she would want to see it. ... If it had come with some kind of explanation for a reason why," then she might have permitted the access. P10 said, "Almost every answer I have is based on context." She chose to allow her husband to see an invoice from her business only once, saying she would want a reason to give him the file because he is "less careful than I am about sharing digital information." P12 used *deny once* for several requests for videos of herself practicing public speaking because "I don't really like to be recorded and on camera, but in the future if there were some really good reason, I'd allow it." She also said she might consider a request from a professor for work from a different class, but she would want a valid academic reason to share it. P08 added that if she were sending requests, she would want to supply a reason, as it seems "presumptuous" to ask without explaining.

Some participants even invented reasons for our randomly generated requests. P23 allowed a request from a friend for his sister's contact information once only, suggesting he might accept that request in an emergency. P12 guessed a friend might want to see her resume "as an example." P13 allowed a work colleague's request for a term paper she'd written on 'Feminism and Film' because "maybe he's interested in the topic and if I can help him get a broader understanding of it, then that would be good." She said an explanation of the request would help her make a better decision about a file like this one, which is "kind of personal but then kind of not."

Other examples demonstrate that this situational dependence can make it difficult to accurately specify policy ahead of time. Seeing a request helped remind P06 of ethical considerations. In reference to a request from another student for a term paper, she said, "Sharing your work with another student has potential to get you in trouble. ... At the time [when filling out the grid], I didn't think it would be an issue with me, but actually seeing, 'Matt is requesting your term paper,' the light went off, saying 'bad idea.'" In her grid, P01 allowed her work photographs to be seen by many requesters. However, when the requests were sent, she reconsidered her professional responsibilities and decided against sharing those files. P02 had the opposite reaction; she allowed a request for a sensitive file, but then said no in the grid because seeing all the people at once on the grid reminded her that information shared with one person will often be disseminated further.

Policies are also complex in other ways

Our findings indicate that many users' policy preferences are also complex in less dynamic ways.

Some participants considered factors beyond the sensitivity of the information in question when making decisions. For example, P09 denied a request from a friend for a Christmas photograph because "she doesn't celebrate Christmas and might be offended." P07 denied several requests for music when she thought the requester might not enjoy the song in question.

Two participants said they might like to grant fine-grained permissions to sections within files, not just to files themselves. P02 keeps all her passwords written down in one file; discussing a request from her teenage daughter for that file, she said that she might want to share some of those passwords with her daughter, but not others, "because someone who's younger doesn't know how to disseminate it or not." Similarly, P21 said he might want to share inventory and planning spreadsheets from his construction business with clients or vendors but redact some cost and pricing details.

Reactive policy creation fits users' interest in control

Several of our participants found a request-based system appealing because they felt it provided added control over the dissemination of their files. This finding confirms that reactive policy creation continues to fit well into users' mental models after a week of simulated exposure to it.

Six participants said they might use reactive policy creation to help them track who was accessing their files and when. P11 said he would use a system like this one to "see who is actually accessing my files" and create considered responses. P21 liked that a system like this would provide a record of "who had access to what." P02 said a reactive policy creation model would make it easier than a proactive model to start saying no if someone is abusing access privileges. P02 also never used the group options for a response, saying that she wanted to know which individual files people were accessing, even if that would mean receiving more requests. P13 added that "it makes me feel comfortable knowing what people are trying to access."

Other participants liked that the reactive policy model incorporates the idea of requesting permission. P01 said, "I enjoyed people asking for permission to see the files." P16 said she used *maybe* for certain grid entries because "sometimes I would be willing to share ... but I'd like him to ask me."

Participants also said the reactive model helped them make better decisions. P15 said answering requests "made you think"; with current systems, she often sends files or forwards e-mails "automatically, without thinking." P14 said the reactive model provides "more of an opportunity to really think about it."

Social norms influence policy choices

As a low-fidelity prototype, our study provided insight into the ways people might use a reactive-policy-creation system. Social factors played a large role in participants' reactions, as well as in the specific policy decisions they made. Understanding these influences can help designers structure a reactive system to maximize user comfort.

Many participants expressed discomfort at receiving requests they considered inappropriate, such as from friends for confidential work documents or from co-workers for financial files. Several said they were confident they would never receive such requests from real people, who would "know better."

Participants had interesting reactions to the *ignore* option.

The 11 participants who used this option applied it only 41 times, totaling only 3% of responses from all users. Seven participants told us they expected never to use this option, mainly because they found it rude not to send a reply. P01 said she wouldn't use *ignore* because she would like a reply if she sent requests. P05 said she wouldn't use *ignore* because "I don't like to live in the gray area." Others suggested that using *ignore* would only postpone the problem, as the requester would simply ask again until she received a response.

Some did see value in using *ignore*. Most commonly, *ignore* was used when the participant felt uncomfortable sharing the file but also uncomfortable denying the request outright. In many cases, this happened when a supervisor or authority figure asked for personal materials. For example, P13 felt "uncomfortable sharing personal pictures with a professor I am not close with, but I don't want to deny him access outright and make him feel uncomfortable. I figure ignoring his request will get the message across." She added that "For me, an ignore is like, 'I never want you to see it, but I don't want to talk about why.' It's just a more passive rejection for me." P06 used *ignore* to avoid saying no to her father, who had asked to see a video from her 21st birthday party that she preferred not to share with him. A few participants suggested that if they chose to *ignore* a requester, that requester might forget about the request entirely, neatly solving the problem.

Other explanations for using *ignore* included adding a delay while deciding how to answer, handling requests the participant considered too outlandish or inappropriate to deserve an answer, and handling requests for files the participant could not immediately identify. One participant used *ignore* to register her displeasure with a friend who had "made me upset that day."

We also asked participants directly if it bothered them to refuse or ignore requests, and if they worried that the requester would be upset. Most participants said they weren't bothered, saying that people who asked for inappropriate files should expect to receive a negative result. This is one area where we think the gap between experience sampling and real life has a strong impact; we expect that people might be more worried about the social consequences of denying or ignoring requests from real people. On the other hand, P06 pointed out that ignoring or denying a request is no worse within a reactive policy creation system than in any other sharing model, "just because I think eventually you have to do that anyway."

Social factors also played a role in several participants' desire to manage file access "manually," in person or otherwise outside a file-sharing system like the one we proposed. Several participants said they might not use a system like ours to request files, preferring to ask directly, over the phone or even by e-mail, rather than allowing the system to send an automated request. P07 said she would rather share in person: "I can show [this friend] this collection of music some other time when I see her on a daily basis." P10, a video producer, said she might use a request for one of her films to generate a personal interaction. She wanted to send a message with her response, saying "I'd love to share that with you. ... Do you want to watch together?" and hoping to "develop a conversation." P04 was also looking for more personal interaction, saying, "If my stepmother wants my friend's contact information, she needs to personally talk to me."

As discussed above, our participants were far more likely to say *yes* than *no* in their grids, and even more likely to accept a request than to say *yes* in the grid. Some of this may be attributable to the files being self-selected (discussed in more detail later), but it may also reflect a general social pressure to say yes when asked for something and to avoid the appearance of having secrets. In addition, as mentioned above, most conflicts between proactive and reactive policy were more permissive reactively. Taken together, these results suggest that perhaps forcing users to respond to direct requests increases social pressure to share.

People have difficulty trusting systems

Several participants were concerned about the security of our proposed system. Some worried that a system that exposed files for access via requests could be attacked, allowing unauthorized access. P08 said she would be "sort of paranoid" about exposing her files this way, in case an attacker "somehow [got] in to take other stuff" or "plant a worm on your drive. ... I would feel more comfortable if I would send it versus they go in and get it off my computer." Another participant worried that showing the existence of financial files could provide incentive for attackers to attempt to break into her system. Others were concerned that the source of a request could not be properly authenticated, and worried they might grant permission to a malicious user masquerading as a friend or co-worker. P08 suggested a spammer would "sooner or later ... hit on a name that was someone I know," and P18 asked "can someone request a file [with my friends name] and I'm giving access that I shouldn't be?"

These concerns, expressed by one quarter of participants, arose organically; in our script, we were careful to specify that "no one can access [your files] without your permission," but participants were unwilling to take our word for it, even with no real data at risk. This suggests a strong level of skepticism toward new systems claiming to provide secure data sharing. Designers of new reactive systems must take this into account and find ways to convincingly demonstrate security in order to get user buy-in.

Reactive policy specification raises specific concerns

Although we found significant evidence of the need for and popularity of reactive policy creation, we also identified specific concerns that must be carefully managed for a system based on this model to be successful. These potential problem areas include annoyance with too many messages, the possibility that users may not recognize a requested file, the need for well-chosen response options, and the potential for disclosing information via lists of requestable files.

Annoyance with requests.

We feared that even a small set of daily requests would cause annoyance and delay in answering, outweighing the possible benefits of reactive policy specification. We discovered this was not at all the case—participants expressed little annoyance and for the most part answered requests promptly. Most participants reported answering requests immediately upon receiving the e-mail. Measured request-response intervals accord with this, showing no increase in delay from the beginning to the end of the study. On a Likert scale, participants disagreed slightly with the statement "I found the e-mail requests annoying." We also asked them whether the number of requests they received was "way too many, very annoying"; "more than I would have preferred but not so many that it really bothered me"; or "a reasonable number of messages." Only one participant chose "way too many"; the rest split evenly between the other two options. Several participants indicated that although they received a lot of messages during the study, they would expect to receive many fewer in real life, limiting the annoyance. A few participants said the requests were no worse than the normal volume of e-mail they receive.

Although our ability to realistically evaluate the annoyance potential of a request-based system was somewhat limited, our impression after talking with participants is that while some frequency of requests would be intolerably annoying, our study did not reach that level. We are optimistic that a real system could maintain a tolerable message frequency. We expect that requests in a real system would decrease over time, as users grant persistent permissions or apply policy to groups of people or files and the system handles more and more situations automatically. This depends, of course, on how often users select the *always* and *always group* responses. In our study, participants selected these options more than two-thirds of the time, despite the fact that only a few coarse-grained file groups were available. P13 said she and a friend "share photos all the time," so she would allow all that friend to see all photos because it's "less of a hassle" than handling them individually. P23 also mentioned that he wished "the system after a while would have recognized" his previous choices and then sent fewer requests.

Trouble recognizing files.

The reactive policy model assumes users can make effective decisions when presented with a request for file access. This assumption breaks down if the user does not recognize the file in question. At least three participants in our study could not identify a requested file, despite the fact that we used only a small subset of their files, which we asked them to describe memorably less than a week prior to the request. We expect this problem to get worse when dealing with a user's complete set of files, some of which may not be named descriptively, over an indeterminate period of time. This problem could be mitigated by offering the user a chance to view the file in question before making a decision.

Additional response options.

We asked participants if they had ever wanted to provide a response not available on the form. The most popular suggestion was to reply asking why the file was wanted, further demonstrating the context-dependent nature of some policies. Other interesting suggestions included a "request pending" response to provide assurance the request was not being ignored, as well as an option to forward the request directly to a third party to make a decision. The forwarding option was mentioned in the context of work-related requests requiring approval from a supervisor as well as requests for address book information that should be approved by the person whose contact information was being requested.

Information disclosure via lists of files.

For a request-based system to work, requesters will likely need some information about what files or directories are available. This will by necessity leak information about what files exist, possibly including sensitive information if filenames are specific and descriptive. In a real system, we would expect to provide users with some control over file visibility; for simplicity, in this study we assigned requests randomly rather than allowing users discretion.

We asked participants how this issue would affect their interest in using this kind of system. Some participants weren't worried at all, because the files they wanted to restrict included things like tax files or calendar entries; these participants reasoned that "everyone" has files like those and so revealing their existence wouldn't be damaging.

As expected, however, many participants did express concerns. These concerns generally took two forms: worry that listing files would provide temptation for attackers to attempt to break the system security, and worry that sensitive information would be leaked as part of file and directory names. For example, P21 mentioned keeping the existence of a new resume file from a boss when searching for a new job; another participant was concerned that revealing e-mail subject lines might lead to a friend discovering a social event she hadn't been invited to. We believe these concerns can be effectively addressed by allowing users to decide which files should be visible and requestable to which people.

LIMITATIONS

It can be difficult to evaluate how well a proposed system feature will work without actually implementing the system to test it. Because of this, we put significant thought into our study design, and evaluating its success was one of our major goals. Despite the inherent limitations of experience-sampling simulation and the specific limitations of our methodology, overall we found evidence to suggest that our results can apply to reactive policy-creation systems in general.

First, although we asked participants to imagine receiving requests and sharing their files, they were aware no data was actually at risk. As a result, a participant might have refused a simulated request that in reality she might have accepted in order to avoid an awkward social situation. On the other hand, participants might have more casually accepted simulated requests than they would real ones. This problem is compounded by randomly generated requests that sometimes appeared bizarre or inappropriate, combined with our inability to tell participants why a given request was made. Several participants said they might accept requests they considered unusual or inappropriate if the requester had a good reason, which we were unable to supply. This may have reduced participants' ability to imagine the system to be real. However, based on our debriefing interviews we contend that our participants suspended their disbelief, took the requests they received seriously, and answered carefully.

In addition, our reactive response form allowed participants to explain their decisions. Participants provided reasons for 87% of responses, indicating thoughtful decision making. We also asked each participant for details about several individual responses; their detailed, reflective answers demonstrate they took the requests seriously. Participants who made policy decisions because they were angry at the requester or concerned she would not like the content, or who created justification scenarios for unlikely requests, clearly engaged with the system as though it were real.

Our decision to pay participants 25¢ per response created another potential limitation. We paid participants this way partially in order to replicate social incentives to respond to requests received from friends, family, and colleagues. The payment, however, might have induced participants to respond to more requests than they would have otherwise, or else reduced their annoyance at receiving requests. Because the payment per request was so small, we don't believe it introduced very much skew into our results. In addition, during the debriefing we asked participants directly about the annoyance of receiving requests; their frank and thoughtful replies provide at least some evidence that annoyance was not suppressed by the payment. Participants' tendency to respond relatively quickly to requests (as discussed earlier) also suggests they were motivated more by interest than payment, as they were paid regardless of when they answered.

Another possible limitation is that the requestors and files used in our study were selected by the participants. Because participants selected only a small subset of their files, it is likely they chose not to mention some of the most sensitive or private items. They also selected only a small subset of the people they know, so it is likely that some people with whom they have unique or complex relationships were left out. We tried to mitigate this by asking about a diverse variety of people and files, and by requiring each participant to supply at least one requester and file in each of several standard categories, including potentially sensitive categories like financial files and supervisors.

We asked participants to use our simulated reactive-policy-creation system for only a week. Longer use might cause behaviors and opinions to change. We believe, however, that the one-week period was enough to gain valuable insights. Observing most participants, it quickly became clear which requesters or files would result in strong policy preferences and which would be complex, dynamic, or borderline cases. Even dynamic decisions often followed similar lines of reasoning for each participant. As a result, we believe that for most participants we reached a saturation point where we had explored the majority of their policy decision space.

Our methodology only addressed reactive policy creation from the point of view of the resource owner; in future work, we hope to examine it from the requester's perspective. We also do not consider how users could view or modify existing policies created via *always* or *never* responses, which would be an important piece of a real reactive policy-creation system. A practical system would also need to consider how users could verify the source of a request.

RELATED WORK

Several prior studies investigate when and how users are willing to share personal data. Brush and Inkpen found users want individualized data and settings within shared devices [4]. Ahern et al. found that users' online photo sharing decisions are driven by concerns about security, social disclosure and convenience [2]. Participants interviewed by Little et al. expressed concern that information disclosure related to ubiquitous computing would disrupt family dynamics. Our work, also focused on personal sharing, examines how well reactive access control meets these consumer needs.

Olson et al. investigated sharing preferences related to a wide variety of personal and professional data [11]. They found that preferences can be categorized into broad groups based on trust relationships as well as the information sensitivity, but that more granular specification may also be required to handle exceptions. In this study, we explored trust relationship and file groupings in the context of a reactive model; we found that while broad groupings do often apply, exceptions remain important and meaningful.

Ackerman's study of the gap between social norms and technological possibility for information sharing in computer-supported cooperative work suggests that human information sharing relationships are fluid, nuanced, and time and context dependent [1]. In a similar result, Razavi et al. found that file sharing preferences in a personal learning space vary with document lifecycle and cannot be sufficiently expressed by static access polices [13]. The same study also demonstrated that implementing this kind of dynamic policy using traditional access controls was too labor intensive for users. Reactive policy creation supports dynamism by allowing information owners to consider requests in context, without the need to predetermine all possible policy in advance. We do not address other issues Ackerman raises, including user buy-in and the ability to gracefully change existing policy.

Early work by Lampson hints at problems with revealing the existence of files in order to validate access control, an unresolved issue in reactive policy creation, but asserts that content owners will shield files with directories that limit access to those who can know the existence of the files [8].

The closest analog to reactive policy creation may be Povey's optimistic access control scheme, which assumes that most requests are legitimate and should be granted [12]. This model relies on external administration to detect and respond to unauthorized access. Neither the reactive nor the optimistic models require content owners to predetermine complete policies; the optimistic model supports exigent circumstances more neatly, but the reactive model avoids the potential for unauthorized accesses that succeed and must be dealt with post facto.

Other prior efforts also concern non-traditional access control models. Egelman et al. designed a new model for home computer accounts, leveraging the fact that home users may not need controls as strict as those required in corporate environments [7]. Bauer et al. implemented the Grey system, which allows mobile phone users to reactively delegate authority to open locked doors [3]. We consider the reactive model in the context of security for personal and home data.

Our use of experience-sampling methodology was inspired by Consolvo et al.'s work examining location-sharing preferences [5]. The experience-sampling method was developed by Csikszentmihalyi and Larson [6].

CONCLUSION

In this study, we set out to determine if users' expressed interest in a reactive policy creation tool would hold up under simulated use of such a tool. Despite some limitations, the experience-sampling methodology we used yielded rich quantitative and qualitative data about users' access-control decisions and the factors that influence them.

Reaction to our simulated system for reactive policy creation was encouraging, if not definitive. We found evidence of access-control policies that are hard to express using existing static mechanisms, but that reactive policy creation can facilitate. Of the policies we collected from participants in our study, 21% used *maybe* or involved conflicts, and hence could not be easily expressed without a reactive policy-creation mechanism or other extension to traditional policy-creation practices. An additional 16% of policies were one-time policies, meant to be changed after a single access. These too may be difficult to define using traditional methods, and we conjecture that many of them would benefit from reactive policy creation.

We also found that when making policy decisions, people want more control and interactivity and rely on social norms, all areas where reactive policy creation can contribute to an access-control system. We found that while there are some clear disadvantages to the reactive model, they don't seem insurmountable. Overall, our results demonstrate that reactive policy creation is a strong candidate for further research and for potential inclusion in future access-control systems.

ACKNOWLEDGMENTS

This research was supported by the National Science Foundation via grants CNS-0831407 and DGE-0903659, by CyLab at Carnegie Mellon under grants DAAD19-02-1-0389 and W911NF-09-1-0273 from the Army Research Office, and by a gift from Microsoft Research. The authors thank Meghana Koushik for her help with the interviews. In memory of Niel Mazurek.

REFERENCES

1. M. S. Ackerman. The intellectual challenge of CSCW: the gap between social requirements and technical feasibility. *Hum.-Comput. Inter.*, September 2000.
2. S. Ahern, D. Eckles, N. S. Good, S. King, M. Naaman, and R. Nair. Over-exposed? privacy patterns and considerations in online and mobile photo sharing. In *Proc. CHI*, 2007.
3. L. Bauer, L. F. Cranor, M. K. Reiter, and K. Vaniea. Lessons learned from the deployment of a smartphone-based access-control system. In *Proc. SOUPS*, July 2007.
4. A. Brush and K. Inkpen. Yours, mine and ours? Sharing and use of technology in domestic environments. In *Proc. UbiComp*. 2007.
5. S. Consolvo, I. E. Smith, T. Matthews, A. LaMarca, J. Tabert, and P. Powledge. Location disclosure to social relations: why, when, & what people want to share. In *Proc. CHI*, 2005.
6. M. Csikszentmihalyi and R. Larson. Validity and reliability of the experience-sampling method. *J Nerv Ment Dis*, September 1987.
7. S. Egelman, A. Brush, and K. Inkpen. Family accounts: A new paradigm for user accounts within the home environment. In *Proc. CSCW*, 2008.
8. B. W. Lampson. Dynamic protection structures. In *Proc. AFIPS*, 1969.
9. R. A. Maxion and R. W. Reeder. Improving user-interface dependability through mitigation of human error. *Int. J. Hum.-Comput. Stud.*, 63, 2005.
10. M. L. Mazurek, J. P. Arsenault, J. Bresee, N. Gupta, I. Ion, C. Johns, D. Lee, Y. Liang, J. Olsen, B. Salmon, R. Shay, K. Vaniea, L. Bauer, L. F. Cranor, G. R. Ganger, and M. K. Reiter. Access control for home data sharing: Attitudes, needs and practices. In *Proc. CHI*, 2010.
11. J. S. Olson, J. Grudin, and E. Horvitz. A study of preferences for sharing and privacy. In *Proc. CHI*, 2005.
12. D. Povey. Optimistic security: A new access control paradigm. In *Proc. NSPW*, 2000.
13. M. N. Razavi and L. Iverson. A grounded theory of information sharing behavior in a personal learning space. In *Proc. CSCW*, 2006.
14. L. Richards. *Handling Qualitative Data: A Practical Guide*. Sage Publications, 2007.
15. L. Richards and J. M. Morse. *Readme First for a User's Guide to Qualitative Methods*. Sage Publications, 2007.
16. S. Voida, W. K. Edwards, M. W. Newman, R. E. Grinter, and N. Ducheneaut. Share and share alike: exploring the user interface affordances of file sharing. In *Proc. CHI*, 2006.