TwoKind Authentication: Protecting Private Information in Untrustworthy Environments (Extended Version)

Katelin Bailey, Apu Kapadia, Linden Vongsathorn, Sean W. Smith

Dartmouth Computer Science Technical Report TR2008-632
August 2008

ABSTRACT

We propose and evaluate *TwoKind Authentication*, a simple and effective technique that allows users to limit access to their private information in untrustworthy environments. Users often log in to Internet sites from insecure computers, and more recently have started divulging their email passwords to social-networking sites, thereby putting their private communications at risk. To mitigate this problem, we explore the use of multiple authenticators for the same account that are associated with specific sets of privileges. In its simplest form, *TwoKind* features two modes of authentication, a `low` and a `high` authenticator. By using a `low` authenticator, users can signal to the server they are in an untrusted environment, following which the server restricts the user's actions, including access to private data.

In this paper, we seek to evaluate the effectiveness of multiple authenticators in promoting safer behavior in users. We demonstrate the effectiveness of this approach through a user experiment — we find that users make a distinction between the two authenticators and generally behave in a security-conscientious way, protecting their `high` authenticator a majority of the time. Our study suggests that *TwoKind* will be beneficial to several Internet applications, particularly if the privileges can be customized to a user's security preferences.

1. INTRODUCTION

Users of online applications today are increasingly placing their private information at risk—a large number of users routinely access Internet sites from untrustworthy computers such as email kiosks and Internet cafes. For example, "Blitz Terminals" (email kiosks) are an integral part of the Dartmouth undergraduate culture even though nearly all students own laptops. Malicious administrators of such computers, and even users who are able to install rogue applications, can easily compromise a user's session and gain unauthorized access to private data. The user's session can be hijacked, or even kept alive by spoofing the logout screen, following which the attacker has unfettered access to all the user's private emails, personal profile information, and so on. In cases where passwords are used as authenticators, the potential damage is even worse because the user's authenticator can be compromised and saved for later use. The risk of session or authenticator compromise is not limited to the use of untrustworthy computers. Social-networking services such as Facebook[1] and LinkedIn[2] ask for users' login information for external email services such as AOL[3] and Google Mail.[4] These social-networking services proceed to download the user's address book and use it to find the user's existing contacts in the social network. In these situations, users provide an online application with unnecessarily, unrestricted access to all of the user's private data on another application. Ideally, the user would authorize the service to download only the user's address book, and disallow access to email. To address these issues, we propose *TwoKind Authentication*,[5] an authentication technique that allows users to *limit the capabilities of an authenticated session*, thereby limiting the amount of damage that can be caused by session or authenticator compromise.

Current authentication mechanisms such as one-time passwords [8] [11] (e.g., RSA SecurID), privileged "trading passwords" (such as those used by eTrade [4] while placing trades), or even PKI tokens do not fully solve this problem. One-time passwords limit the future damage possibly caused by stolen credentials, but allows full-scale compromise in a single hijacked session. PKI tokens do not protect against

[1] http://www.facebook.com
[2] http:// www.linkedin.com
[3] http://www.aol.com
[4] http://mail.google.com
[5] A brief outline of *TwoKind Authentication* and our proposed user experiment was presented as a poster at SOUPS 2007 [1].

.

hijacked sessions either, and can also be susceptible to authenticator hijacking.[6] eTrade-style trading passwords are required by server policy, and users are required to re-enter a trading password while executing privileged actions such as trades. Such systems, however, have usability concerns, since the default mode of access is that of low privilege. Requiring users of an email application to enter a high-privilege password each time they wanted to access archived email, for example, would be a nuisance. *TwoKind* does not prevent a session from being hijacked; rather, it gives users a convenient method to effectively limit the damage caused by hijacking, and allows more usable access modes from trustworthy environments.

In its simplest form, *TwoKind* features two modes of authentication—`high` and `low`. *TwoKind* authenticators could include passwords, PKI-based keys, or hardware tokens. To signal untrustworthy environments to the server, users employ their `low` authenticator to limit the privileges of the session. For example, a user's `low` authenticator for an email service may allow access to only the user's new messages (and not previously viewed messages, messages in folders, and so on); the `low` authenticator for a bank account may disallow any financial transactions or access to financial records other than the account balance. In contrast to eTrade's trading-password approach, which requires users to log in with a low-privileged password by default, *TwoKind* allows users to log in with full-privileged access under normal circumstances (assuming that users normally operate in trustworthy environments, such as on their personal computers), making it less intrusive in general. More generally, *TwoKind* allows users to assign specific permissions to their `low` authenticator, or use any number of `low` authenticators with different, but limited, capabilities.[7]

We believe that *TwoKind* authentication will be an attractive solution to people already sensitive to the security of their account, and studying the effectiveness of *TwoKind* in such a population would probably yield obvious and uninteresting results. Instead, we evaluate its effectiveness in a general population and seek to determine how often users will protect their `high` authenticator in unsafe environments. We focus on the question of whether users would apply such a bimodal cost-benefit tradeoff for authenticated sessions, i.e., "is the *TwoKind model* easy to understand and apply?"

In our user experiment, we do not investigate the usability of a particular *instantiation* of *TwoKind*. For example, in a password-based instantiation, users must memorize additional passwords, and the usability of passwords (in general) would interfere with our measurements on the usability of the *TwoKind* model. In practice, we expect users to pick a few `low` passwords to be used across several accounts. It has already been shown that most users reuse passwords across various accounts [17]. In PKI-based *TwoKind*, users must carry an extra PKI token; not much of a stretch if users can be be convinced to carry a single token.

Our user experiment showed that 70% of subjects understood the motivation behind *TwoKind* and were able to apply *TwoKind* effectively, i.e., they made pragmatic use of *TwoKind* based on their assessment of risk. Additionally, 49% of the time, subjects were able to protect their `high` passwords in unsafe environments and we therefore propose *TwoKind* as a useful authentication method.

1.1 Contributions

We make the following contributions:

1. We propose the use of *TwoKind* (and its generalization), which allows users, or applications on their behalf, to log in with restricted privileges in untrusted environments.

2. We perform a controlled user experiment designed to study the effectiveness of *TwoKind* if employed by everyday users.

3. Based on our results, we find that 70% of users can indeed benefit from the added security provided by *TwoKind*.

1.2 Overview

The remainder of the paper is structured as follows. We describe *TwoKind* Authentication in Section 2. Section 3 describes our methodology for evaluating *TwoKind* Authentication, and in Section 4 we detail the results of our experiment. We present related work in Section 5 and conclude in Section 7.

2. TWOKIND AUTHENTICATION

We now describe *TwoKind* more precisely, followed by its generalization with more than two authenticators.

2.1 Two authenticators for the same account

In *TwoKind*, users are assigned two authenticators, `high` and `low`, for the *same account*,[8] where the `low` authenticator is associated with restricted privileges. These authenticators are used to signal to the server whether the user is in a *safe* or *unsafe* situation, depending on whether they trust the security of the session. For example, a user may determine that using an email kiosk is unsafe, or that giving Facebook his or her Google Mail password is unsafe. A user may carry two PKI tokens, and use the `low` token in unsafe environments, whereas in password-based *TwoKind*, users can use a `low` password.

More formally, let A be the set of all privileges for user u, and $P(x)$ be all the privileges associated with the authenticator x. In our model, we assume that the `high` authenticator is the default authenticator as would be used without multiple authenticators, and has associated with it all the privileges $P(\texttt{high}) = A$. The `low` authenticator has some proper subset of these privileges, i.e., $P(\texttt{low}) \subset A$. The privileges associated with the session are determined by the `low` or `high` authenticator that is used, resulting in privileges $P(\texttt{low})$ or $P(\texttt{high})$ respectively. We note that either the server or the user can define set of privileges $P(\texttt{low})$ associated with the `low` authenticator, although we do not examine the usability of user-defined privileges in this paper.

[6]Authenticators such as private keys can also be "hijacked" [10], where malicious software is able to instruct the PKI token to dutifully sign its requests. In the remainder of the paper we use "authenticator hijacking" to also include authenticator compromise, as in the case of compromised passwords.

[7]We believe, however, that several authenticators with different privileges will lead to confusion, and that *TwoKind* authentication—with exactly two levels of privilege—will be more usable in practice. In this paper, therefore, we evaluate the effectiveness of *TwoKind*.

[8]As opposed to being assigned two *separate accounts*, such as *root* and *user*.

2.2 Multiple authenticators

Although we focus on and evaluate the simpler concept of *TwoKind Authentication*, we expect that some users may desire the ability to create several authenticators for different uses. For example, the user may create an "address-book password" so that social-networking sites may access only the user's address book—nothing else connected with that email account. The user may create a separate "travel password" for use in Internet cafes, allowing the sending and receiving of new mail only (blocking access to folders, and disallowing any deletions or modifications to the account). For n-Kind authentication, users possess the following authenticators: $\texttt{low}_1, \texttt{low}_2, \ldots, \texttt{low}_{n-1}, \texttt{high}$. For each $\texttt{low}_i$, we have that $P(\texttt{low}_i) \subset A$, and these sets are not necessarily disjoint (different `low` passwords may have some privileges in common). Again, either the server or the user could define these sets of privileges. The server may even allow a combination of the two (a static set of server-defined permissions, with the option of adding and/or removing certain permissions to the authenticator).

3. USER-EXPERIMENT DESIGN

We designed our user study to determine how users would employ their *TwoKind* authenticators when presented with safe and unsafe environments. Subjects participated in a game designed to measure their risk-taking habits, given *TwoKind* as a means to mitigate the risks.

3.1 Design rationale

The prime concern in creating the study was to avoid coercing the subjects into behaving in a manner that would yield positive results for the study. To overemphasize neither task completion nor the security concerns, we designed the study as a game. The game incentivized subjects to complete tasks in return for points, but disincentivized subjects from taking risks by subtracting points for risky behavior. The point of abstracting the study into a game, was to make it difficult for subjects to perceive a "correct" behavioral pattern. To maintain relevance, we created an environment in which the subjects would understand the motivation behind having two authenticators, and the possible usefulness of this approach.

3.2 Experimental protocol

Because this study was being performed on college students, we designed a game emulating a Facebook-style application: an application which college students are familiar with and comfortable navigating. While we considered basing the game on a banking application, that would have overemphasized the idea of protecting one's assets—an attitude that would lead to fewer risks taken, but would not be realistic in the context of everyday applications such as email, blogs, photo-sharing sites, and so on. Within the Facebook-style application, which we called *Green Book Online*, subjects were given a fictitious user profile with preset personal information. The subject was then presented with a series of "desired updates" to the profile. To perform these updates, subjects were required to log in to the application using *TwoKind* authenticators. Figure 1 shows a screenshot of *Green Book Online*.

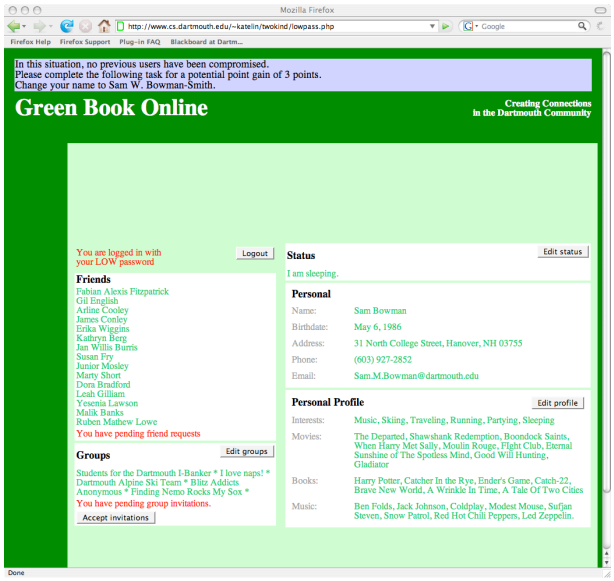

Figure 1: Screenshot of the `low` privilege environment in *Green Book Online*, where buttons for modifying personal information and friends have been removed so that it is absolutely clear that users may not access these functions.

3.3 Environments

To motivate the idea of safe and unsafe situations, we created two different environments in which subjects were asked to complete tasks. The words "safe" and "unsafe" have strong connotations of "good" and "bad," which would overly influence subjects to behave in a cautious manner. We therefore described the safe environment as "On this computer, no previous users have been compromised" and the unsafe environment as "On this computer, some previous users have been compromised." Since users in the real world have no absolute method to determine whether a workstation is safe, this choice of wording allows subjects adjust their risk-taking behavior according to their own perceived risk, thus making the results more relevant to real-world situations. In an actual implementation of *TwoKind*, the users will be creating their own definitions of safe and risky, or adhering to an organization's explanation (such as "Kiosks are to be treated as untrusted computers"). Our aim was to determine whether the subjects, given a distinct line between secure and insecure environments, were able to use *TwoKind* authenticators to make a distinction at that line in such a way that they were protected.

3.4 Authenticators

To access their information in either of these environments, subjects were given the option to log in to an environment by indicating use of a `high` password, a `low` password, or to not log in at all. We chose to use the word "password" instead of "authenticator" because users are more familiar with passwords, and we wanted to avoid describing too much terminology. We chose not to give subjects real passwords because password usability has been tested in previous studies [17] and including a dimension of remembering passwords would only obfuscate the results we were interested in—the

effectiveness of the *TwoKind* model. Additionally, because *TwoKind* would not necessarily be implemented with passwords, the memorization of two passwords is not central to the concept of *TwoKind*, and the results generalize to other authenticators. The `high` password option allowed subjects to view and edit all of their personal information in both safe and unsafe environments. The `low` password option allowed subjects to view all of their personal information, but edit a limited selection of their information. Subjects had five attributes in their profiles: a list of friends; personal information like an email address; a profile which shows favorite movies, books, and so on; a list of groups to which they belong; and a current status. In high-privilege mode, the subjects were able to modify all of these attributes. In low-privilege mode, they only had the ability to modify their groups, status and profile. These three were chosen as low-security actions because they are easily reversible and do not involve modifying the user's social network or private information.

When logged in, the subject was presented with a screen of their information, with certain editing capabilities removed if they were logged in with their `low` password. We also provided the ability to skip a task (users could click on "situation unfavorable" instead of logging in), which represents the real-life analog of a person choosing not to perform an action in an insecure environment. We included this option to emulate the situation in which a user places security concerns above the mandate of completing a task. Since this is an optimal choice in some situations, the option to skip (or otherwise postpone) a task is important to include. Figure 2 shows a screenshot of the login screen.

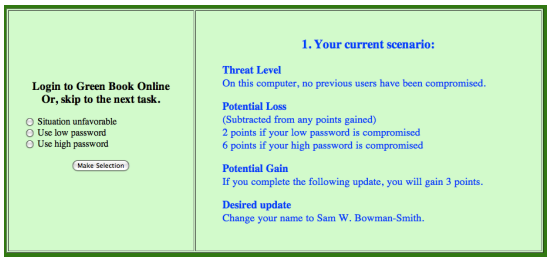

Figure 2: Screenshot of the login screen

3.5 Task completion and points

There were two types of updates a subject could perform: updates that could be completed with either password, and updates that required their `high` password. Subjects were presented with 20 desired updates in a random order, with five updates for each of the four possible situations: high/low-privilege updates in safe/unsafe environments. There were, therefore, some tasks that could be completed only with the high password in an unsafe environment.

To motivate subjects to complete tasks, each subject was given an initial score of 60 points, and earned additional points for each successfully completed update and sometimes lost points for risky behavior. At the end of the game, the subject was given a certain amount of money directly related to their final score. The subject gained 3 points for completing a high-privilege task and 1 point for completing a low-privilege task. There was an unspecified probability that if a subject logged into an unsafe environment with either password, they would lose some of the points they had accumulated thus far. A compromised `high` password lost 6 points, while a compromised `low` password lost 2 points. Subjects thus focused on maximizing their reward at the end of the study, instead of task completion.

For a fixed probability p of compromise, the expected gain in points for unsafe environments is $1 - 2p$ and $3 - 6p$ for low- and high-privilege tasks respectively. The value of p, however, is unspecified to avoid biasing users towards risking or not risking passwords. We also did not want to bias the use of one authenticator over another. Thus for a fixed p, there is either an expected gain for both types of tasks or likewise an expected loss for both types. The variance, however, differs for the two situations, and thus we can study how many users tend to take higher risks for potentially higher rewards. This game, therefore, allowed us to study the effectiveness of the *TwoKind* model without obviously biasing the subjects' choices.

3.6 Survey

To further explore participants' reaction to the TwoKind method, and to collect information such as whether participants had a background in computer science, we administered a closing survey to each participant. We were interested in measuring how computer-science experience may lead to a change in security-related behavior and accounted for it in the results (Section 4). We did not ask for any more demographic information because of privacy concerns. Participants were given an option to provide thoughts and comments as well as to answer specific questions about using two authenticators for specific applications.

4. RESULTS

We now describe the results of our study. We identify several categories of users depending on their behavior in various situations and present results of our survey questions.

4.1 Patterns of Behavior

While designing our study, we expected subjects to react to both the type of environment (safe or unsafe) and the level of privilege (`high` or `low`) required for the desired update. We found that 26 of 33 subjects (79% of all subjects) fit this pattern, and we discuss the patterns followed by the other 7 subjects (21%). All subjects fit into their categories at least 90% of the time. We attribute minor variations in subjects' behavior to mistakes, an occasional risk, or uncertainty at the beginning of the study. These categories provide information on trends that are useful in further analysis of the effectiveness of *TwoKind*. The patterns are as follows (the results are summarized in Figure 3):

1. *Sensitive to environment and privilege.* All 26 subjects in this category (79%) adhered to the principle of least privilege, i.e., logging in with a `high` password only when it was required to complete a task. Users however, did also respond to the type of environment, which is important for understanding the effectiveness of *TwoKind*. We have further broken this category down into five patterns.

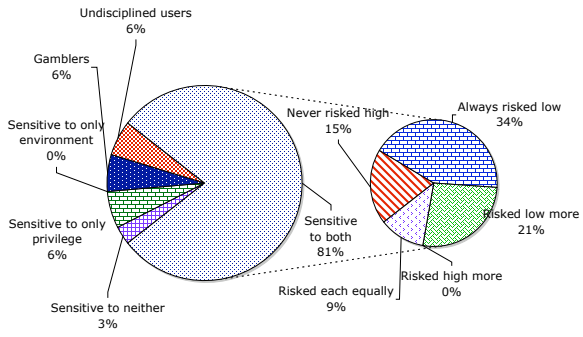

Figure 3: Breakdown of subjects into various groups, with the subsets for group 1, *sensitive to both*.

(a) *Never risked* `high` *password.* These subjects used their `high` password to complete only high-privilege tasks in a safe environment. While some of these subjects were willing to reveal their `low` password, none of them took risks with their `high` password. This is a *safe* outcome where the `high` password is never compromised. 5 of 33 subjects (15%) fell into this group.

(b) *Always risked* `low` *password.* These subjects always took risks for the low-privilege tasks (using their `low` password), but took risks for high-privilege tasks (using their `high` password) in unsafe environments between 40 and 80% of the time. This category shows a clear distinction between the two passwords, with the `high` password being better protected. 11 of 33 subjects (33%) fit into this pattern.

(c) *Risked* `low` *password more.* These subjects took risks for low-privilege tasks some of the time and also took risks for the high-privilege tasks some of the time, but took risks more often in cases where they can use their `low` password. The average disparity between the use of `high` and `low` passwords was 35%. Even though these subjects risked their `high` passwords at times, they were able to make a distinction between safe and unsafe situations and modify their risk-taking behavior, taking less risks in an unsafe environment. 7 of 33 subjects (21%) fell into this category.

(d) *Risked* `high` *password more.* These subjects took risks for both types of tasks and passwords, but took more risks for the `high` password. No subjects fell into this category, which is a good indication that they were making distinctions between the passwords in a way which favored protecting their high password.

(e) *Risked both equally.* Subjects who took risks with both their `high` password and their `low` password, at the same rate. Subjects in this category understood the difference in privilege, using their low password whenever possible, and understood the difference in environment, as demonstrated by a change in behavior. However, they were equally likely to risk their `high` or `low` password, indicating that they did not appreciate the fact that risking a `high` password was a more costly risk than risking the `low` password. 3 of 33 subjects (9%) fell into this category.

2. *Other categories.* One subject (3%) did not appear to understand the purpose of having two passwords, and used the same password regardless of which environment he/she was given. This subject was *sensitive to neither environment nor privilege.* Two subjects (6%) ignored the type of environment, and followed the principle of least privilege for completing tasks. These subjects were *sensitive to privilege but not environment* and would therefore risk compromise in unsafe environments. Two subjects (6%) made the decision to always risk their `high` password, while only risking their `low` password some of the time. We call these subjects *gamblers.* Lastly, two subjects (6%) did not seem to have a discernible pattern. We call them *undisciplined users.*

Students with computer-science background.

Table 1 shows how experience in computer science (CS) correlated with subjects' behavior. 13 subjects (39%) had a background in some sort of computer science. There were few differences between the CS and non-CS subjects, but we discuss the major points of disparity below.

CS students who took risks with both passwords made, on average, a greater distinction between privilege levels. On average, the disparity between `high` and `low` was 50% for CS students, as opposed to 28% for non-CS students. For example, if a CS student risked their `low` password 80% of the time, they would risk their `high` password 30% of the time, while a non-CS student would risk their `high` password 52% of the time, a noticeable difference. In short, non-CS students were more likely to risk their `high` passwords.

Consistent with the previous finding, CS students also had a lower risk-taking tendency in the category *Sensitive to both: always take low-privilege risks* (1b). CS students who always risked their `low` password risked their `high` password 48% of the time, as opposed to non-CS students, who risked their `high` password 67% of the time. It is also interesting to note that neither of the *gamblers* were CS students.

In cases where users are weighing the risks and potential benefits, CS students tend to be more risk-conscious and cautious about revealing their passwords and appear to better understand the difference between privilege levels and the need to protect the `high` authenticator in unsafe environments.

4.2 General Trends

It is important to note that there were several overarching trends. All but four users followed the principle of least privilege, using their `high` password only when it was necessary to complete the task, and using the `low` password whenever they could. The prominence of this behavior is a positive sign towards individuals using the lowest privilege possible, and by proxy leaving themselves open to as little risk as they can. Again, it is interesting to see how these users behave in unsafe environments, since the principle of

Table 1: Number of computer science students in each behavior group

Category	*CS Students*	*Non-CS Students*	*Total*
Sensitive to both: never take high-privilege risks	15%	15%	15%
Sensitive to both: always take low-privilege risks	38%	30%	33%
Sensitive to both: take low-privilege risks more often	8%	30%	21%
Sensitive to both: take high-privilege risks more often	0%	0%	0%
Sensitive to both: take equal risks for both types of tasks	15%	5%	9%
(Sensitive to both: totals)	(76%)	(80%)	(79%)
Sensitive to neither environment nor privilege	8%	0%	3%
Sensitive to privilege but not environment	8%	5%	6%
Sensitive to environment but not privilege	0%	0%	0%
Gamblers	0%	10%	6%
Undisciplined Users	8%	5%	6%

least privilege by itself will result in the compromise of the `high` authenticator when used in unsafe environments.

While few users were willing to skip tasks on a consistent basis, they were able to show good decision-making skills in weighing the risks of using their `high` password as opposed to the `low` password, often protecting their `high` password more than their `low` password. This behavior is consistent with the goals of *TwoKind*, which aims not to create a method of absolute rules for when to use `high` versus `low`, but to give the users the choice of protecting some capabilities over others, and allowing low-privilege tasks to be accomplished in unsafe environments.

Overall, 70% of subjects were sensitive to both (group 1) and made a distinction between the passwords in an *unsafe* environment (risking `high` less). We conclude that *TwoKind* provides a multiple authentication method that is useful to this 70% of users, who make a conscious decision based on the perceived risks and are more protective of their `high` authenticator.

4.3 Task Groups

It is also interesting to see how subjects reacted to groups of tasks. Recall, there were four main groups that tasks fall into; permutations of the safe or unsafe environment combined with a high- or low-privilege tasks. Figure 4 shows how subjects reacted to these four situations.

1. *High-privilege task, safe environment.* Subjects overwhelmingly chose to log in with their `high` password, (90% of the time) since there was no reason to do otherwise.

2. *High-privilege task, unsafe environment.* We observed a fairly even split between users who attempted to log in with their `high` password and those who skipped the task, with a small percentage trying multiple authenticators. These users were probably trying to accomplish the task with the `low` password, and most of them skipped when this was not possible. About half of the time (49%), subjects would have protected their `high` password from compromise.

3. *Low-privilege task, safe environment.* Although in this environment it would have been safe to use the `high` password, we see an overwhelming majority (86%) choosing to use their `low` password, which shows that subjects followed the principle of least privilege.

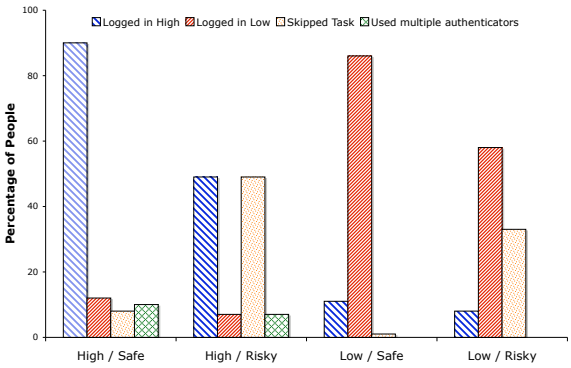

Figure 4: Chart of task groupings. This figure shows the overall reaction of subjects to various permutations of safe/risky environment and high/low-privilege tasks.

4. *Low-privilege task, unsafe environment.* A small percentage (8%) of subjects risked their `high` password here, but the majority (58%) chose to log in with their `low` password. It is interesting to see that a significant number of subjects (33%) chose to also protect their `low` password from compromise as well, and skipped these tasks.

It is important to remember that each of these groups contains five tasks each from 33 subjects, meaning that individual users could have been inconsistent over these groups, as we saw with the patterns discussed above. Subjects followed the patterns they were placed into 90% of the time, and while those patterns varied widely in several circumstances (i.e., the subjects who *never* risked their `high` passwords versus those who *always* risked their `high` passwords) analyzing the overall trends gives us an idea of how the method would be used by a larger population, regardless of the 10% inconsistency from subjects. These overall trends from groups of tasks, when combined with the patterns we discussed above, gives a full picture of users' behavior. 49% of the time, subjects protect their `high` password from compromise, and an even higher percentage (70%) meet the goals of the *TwoKind* project of making pragmatic decisions depending on the type of environment. We posit that this is a large enough portion

of the population to establish *TwoKind* as a viable future authentication mechanism.

4.4 Survey Results

We administered a closing survey to users from the study, asking them a few questions which have helped us to identify where *TwoKind* may be most useful. They were asked whether they had taken any computer science classes, and their answers were taken into account when analyzing the data and creating patterns. The following questions on *TwoKind* for email authentication, and Friend-finder passwords (explicitly stated below) interest us the most. Table 2 shows the responses received from the first two questions.

TwoKind *for email authentication.*

The first question concerns an email application, BlitzMail, which is pervasive on the Dartmouth College campus. Undergraduate students use the application for email, but also treat it as a means of instantaneous communication, often in place of cellphones or more popular instant-messaging programs. Public BlitzMail terminals are located around campus, and students regularly log into these computers, which could easily be compromised.

The question asks: *Suppose you had an additional password for BlitzMail that would allow you to read any mail in your inbox and send messages, but not to read mail in other folders or the trash. Given that the low-privilege password would limit the risk of other people reading your mail, would you use the additional (low-privilege) password in insecure situations, such as Blitz terminals around campus?*

Responses to this question varied, and 45% of the subjects expressing interest in this option. The reasons given for negative responses show that students were generally receptive to the idea, but wanted more flexibility in the passwords, so that they could specify their own preferences for `low` password capabilities. Several students, for instance, indicated that they don't use folders for their mail, so such an option would be useless to them. Others wanted an option that would not allow deleting or moving mail from the inbox, while others wanted to go even further and not allow new mail to be sent with the `low` password. Considering how ingrained the current BlitzMail system is in the undergraduate community, that 45% of subjects were willing to consider changing their current authentication method is a significant statement. If one further includes those that would consider it with added features or increased flexibility, the percentage rises to 60%.

Friend-finder passwords.

The second question concerns a larger issue: *Many sites, such as Facebook, offer a "friend finder" service that asks users to give their email address and password, which they will use to download your address book and find your friends for you. However, this gives them unlimited access to your email account. If there were a low-privilege password that was designed so that it would only allow access to your address book and NO abilities to read or send mail, would you find it useful in these situations?*

The responses to this question were a fairly even split of yes (48%) and no (45%), but more positive than the answers to the last one. While many of the comments indicated that users were unwilling to use "friend finder" as-is, several indicated that if a password existed that allowed programs to have access only to their address book, they would take this option. One subject even expressed surprise that the social-networking applications as-is have access to her full account. He/she had no idea that by giving her email address and password he/she was giving them implicit access to everything from that account. While this reaction is more extreme than we found among most subjects, the desire for more restricted access is clearly a common one, as indicated by the answers to the question. We argue that this need can be filled by the *TwoKind* authentication method, as supported by our survey responses.

Free-form feedback.

The third question merely asked for other comments on the user study or the application. Five subjects mentioned that having multiple passwords is a cumbersome method, an issue that we discuss in Section 2. One person specifically said "Unless you make people aware of the risks, the inconvenience of having two passwords is higher." This is an excellent point that brings to light the importance that education would have to play in implementing the *TwoKind* authentication method. Without some minimal education, users would be unsure of the point of two authenticators and where they should use them.

Another comment indicated that for people to make the best use of *TwoKind*, the users would have to be able to make choices about what capabilities they want for the `low` [authenticator]. Since each user is likely to have different privacy or security preferences, flexible settings would allow users to personalize the passwords in a way which is most convenient for them. For consistent results, this study did not address this concern, but it could be addressed in further work.

4.5 Scope of the user study

We performed our study on 33 Dartmouth College undergraduate students, and our results are therefore representative of a younger college-going demographic. Further work could seek to explore the differences between college students' use of *TwoKind*, and use among the general public.

Our study presented users with a single task to be completed in a particular situation. It is possible that users' behavior will be altered when they log into their account for long-lived sessions to perform multiple tasks. Our results are more relevant to occasional and small bursts of activity such as checking email messages or reviewing bank balances at a public kiosk, or the occasional use of "friend finder" services on social-networking websites.

Our study does not address the issue of how users judge whether they are in an untrustworthy environment. Instead, our study seeks only to see how users' behavior changes in what they perceive as a safe or unsafe environment. In practice, users would need to be educated that, for example, computer science labs were safe, but kiosks were not. Moreover, it is possible that users might consider their personal computers to be in a different class ("absolutely safe"?) and use their high passwords exclusively instead of following the principle of least privilege for email and bank accounts. In the future it might be interesting to study more about the two types of environments.

5. RELATED WORK

The concept of principle of least privilege has been known

Table 2: Responses to the survey questions

Question	*No*	*Yes*	*Uncertain*
Would use TwoKind *with BlitzMail*	55%	36%	9%
Would use TwoKind *with "friend finder"*	45%	48%	6%

for several decades [12]. This principle states that users should log in with only enough privileges necessary to accomplish the desired task. Our work provides users with multiple authenticators associated with different levels of privilege, but does not emphasize this principle. Instead, we expect users to employ authenticators based on the trustworthiness of the environment. For example, users may trust their personal laptop machines, and log into their email account with full-privileged access, but use a low-privileged password when using an untrusted machine.

TwoKind allows users to *signal* untrustworthy environments to servers. Some websites include radio buttons on the login page that ask users to indicate whether they are on a public computer.[9] If so, the user's credentials or personal information are not stored within the browser as a cookie. The problem with this approach is that it is not designed to thwart malicious administrators, who can easily invert the user's choice. Signaling using TwoKind, on the other hand, cannot be subverted by a malicious administrator, since the authentication token itself encodes the trustworthiness of the environment.

Instead or relying on the user to signal the type of environment, the server may be able to determine the trustworthiness of the client configuration by itself. For example, Seshadri et al. [13] have developed a remote software attestation technique using which a server can ensure that the remote machine is in a valid configuration. In the context of our problem, the server could degrade the privileges associated with the session if it is unable to attest the remote platform. Along these lines, computers now shipping with the Trusted Platform Module [16] can use the Direct Anonymous Attestation (DAA) protocol [3] to attest to its trustworthy configuration anonymously (earlier versions of the TPM specification allowed for attestation without the privacy guarantees of DAA). Garriss et al. [5] leverage users' mobile devices to assess the integrity of a kiosk's software before using it. The problem with these techniques is that they have not seen widespread deployment. Since most installations (e.g., at Internet cafes) do not run such attestation services, our approach is still the most practical since it does not require any modification of the client configuration. Furthermore, even if the server has determined that the remote platform is untrustworthy, it has no way of warning the user, thereby limiting the utility of these approaches. In contrast, TwoKind allows users to determine the trustworthiness of the environment for themselves and act upon that determination.

Other work has explored "proxy certificates" as a means to delegate limited privileges to other users or platforms. Users rely on certificate repositories such as MyProxy [2] to issue temporary proxy certificates when needed. Followup work such as SHEMP: Secure Hardware Enhanced MyProxy [9] extends this approach to make the credential repository more trustworthy. The limitation of these approaches is that users have no way to access these repositories from an untrusted machine. To address this shortcoming, Sinclair and Smith [15] create a "Portable PKI" where users can leverage trusted mobile devices to generate proxy certificates for untrusted machines. A proxy certificate is downloaded to the workstation, with attached privileges for the session. Sharp et al. [14] allow users to make use of untrusted displays using a trusted mobile device—the trusted device displays the sensitive information, while the untrusted display shows only general information. While these techniques are certainly viable options for untrustworthy environments, they require modifications to the client machines. Again, TwoKind does not require any such modifications, and is a practical solution in the near term.

One concern with our approach is that it requires users to memorize additional passwords. On the issue of memorability, Yan et al. [17] have found that passwords based on mnemonic phrases tend to be as memorable as naively-selected passwords, and as secure as randomly-selected passwords. Unfortunately, about 10% of users did not comply with the password-selection guidelines and were therefore easily compromised. We emphasize again that in our system the `low` password is expected to be compromised, and hope that users can create simple mnemonic phases for their `low` passwords.

Gaw and Felten [6] found that as users get older, they accumulate more accounts online. Yet, users tend to have only about three distinct passwords that they tend to reuse, and the number of distinct passwords does not seem to grow with the number of unique accounts. We expect that users may create a `low` password that is reused across several accounts, and will have a low-level of security associated with that password (further justifying its reuse across several accounts).

Halderman et al. [7] develop a technique that requires users to memorize only one password. Using a secure hash function, this password can be transformed into distinct passwords for each website. Related approaches include those by Yee and Sitaker [18]. This technique reduces the burden on a user's memorization, although the user must still rely on the client computer to compute these hashes. This technique, therefore, is not meant to be used in untrustworthy environments, and protects against passwords compromised *by the server* and not against malicious client machines.

6. FUTURE WORK

The results from our study suggest several possible areas for further investigation. It would be interesting to study how multiple passwords affect the security of each individual password, any relationships between passwords for a particular account (does a user's `low` password provide a hint about the `high` password?), and how users would react to having

[9]More recently, services such as Hotmail and Google Mail have explicitly asked if they should "Remember [the user] on this computer?"

to remember and maintain additional passwords. Further exploration of the usability of other authentication methods like PKI tokens would provide a more complete survey of security solutions.

Additionally, it would be interesting to study how privilege levels could be set and maintained, and the behavior of users when given the opportunity to set their own privilege levels. In our study, we told users whether a situation was safe or unsafe. Research into users' ability to judge the security of real-world situations would provide insight into the effectiveness of solutions like TwoKind.

7. CONCLUSIONS

We proposed a method called *TwoKind Authentication*, which protects users from malicious administrators or third party services. Using the `low` authenticator, users can signal untrustworthy environments to the server and reduce the privileges associated with that session. A compromised authenticator, therefore, allows attackers only limited access to private information. We performed a user experiment with thirty-three subjects, in which 70% of users employed the two authenticators in a way which was consistent with the goals of *TwoKind*, including making distinctions between environments, recognizing privilege levels, and protecting the `high` authenticator by means of the `low` authenticator. Furthermore, 49% of the time, subjects did not risk their `high` authenticator in unsafe environments. Our study suggests that for the majority of users, *TwoKind* would enable better security practices in the real world.

We believe that the *TwoKind* method is a feasible and useful authentication method, which is an improvement to the current practice of using a single high-privilege authenticator, or repeatedly requiring high-privilege authenticators for certain actions. Although users may not always use the *TwoKind* method ideally, allowing their `high` authenticator to be compromised on occasion, it appears that the majority of users would employ *TwoKind* to their benefit. Since the study on the whole demonstrated that users are willing to use the `low` authenticator to protect the `high` authenticator, *TwoKind* seems to generally increase the security of high-privilege actions and reduce the risk of compromise in unsafe environments.

8. ACKNOWLEDGMENTS

We would like to thank Denise Anthony, Sergey Bratus, Chris Masone, Sara Sinclair, and the anonymous reviewers for their helpful comments. This research was supported in part by the NSF, under Grant CNS-0448499, the Bureau of Justice Assistance, under grant 2005-DD-BX-1091, and the Women in Science Project at Dartmouth College. The views and conclusions do not necessarily reflect the views of the sponsors.

9. REFERENCES

[1] K. Bailey, L. Vongsathorn, A. Kapadia, C. Masone, and S. W. Smith. TwoKind authentication: Usable authenticators for untrustworthy environments (poster abstract). In *Symposium on Usable Privacy and Security (SOUPS 2007)*, pages 169–170, July 2007.

[2] J. Basney, M. Humphrey, and V. Welch. The MyProxy online credential repository: Research articles. *Softw. Pract. Exper.*, 35(9):801–816, 2005.

[3] E. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. In *CCS '04: Proceedings of the 11th ACM conference on Computer and communications security*, pages 132–145, New York, NY, USA, 2004. ACM.

[4] eTrade Trading Passwords. https://www.etradeaustralia.com.au/EStation/hep_aec_connecting.asp.

[5] S. Garriss, R. Caceres, S. Berger, R. Sailer, L. van Doorn, and X. Zhang. Towards trustworthy kiosk computing. *Mobile Computing Systems and Applications, 2007. HotMobile 2007. Eighth IEEE Workshop on*, pages 41–45, 8-9 March 2007.

[6] S. Gaw and E. W. Felten. Password management strategies for online accounts. In *SOUPS '06: Proceedings of the second symposium on Usable privacy and security*, pages 44–55, New York, NY, USA, 2006. ACM.

[7] J. A. Halderman, B. Waters, and E. W. Felten. A convenient method for securely managing passwords. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 471–479, New York, NY, USA, 2005. ACM.

[8] L. Lamport. Password authentication with insecure communication. *Communications of the ACM*, 24(11):770–772, Nov. 1981.

[9] J. Marchesini and S. W. Smith. SHEMP: Secure Hardware Enhanced MyProxy. In *PST*, 2005.

[10] J. Marchesini, S. W. Smith, and M. Zhao. Keyjacking: the surprising insecurity of client-side SSL. *Computers and Security*, 24(2):109–123, 2005.

[11] RSA SecurID. http://www.rsa.com/node.aspx?id=1156.

[12] J. H. Saltzer and M. D. Schroeder. The protection of information in computer systems. *Communications of the ACM*, 17(7), July 1974.

[13] A. Seshadri, A. Perrig, L. van Doorn, and P. Khosla. SWAtt: Software-based attestation for embedded devices. In *Proceedings of the IEEE Symposium on Security and Privacy*, May 2004.

[14] R. Sharp, J. Scott, and A. Beresford. Secure mobile computing via public terminals. In *Proceedings of the 4th International Conference on Pervasive Computing (Pervasive)*, pages 238–253, May 2006.

[15] S. Sinclair and S. W. Smith. PorKI: Making user PKI safe on machines of heterogeneous trustworthiness. In *21st Annual Computer Security Applications Conference*, pages 419–430, Los Alamitos, CA, USA, 2005.

[16] Trusted computing group, May 2005. https://www.trustedcomputinggroup.org/home.

[17] J. Yan, A. Blackwell, R. Anderson, and A. Grant. Password memorability and security: Empirical results. *IEEE Security and Privacy*, 2(5):25–31, 2004.

[18] K.-P. Yee and K. Sitaker. Passpet: convenient password management and phishing protection. In *SOUPS '06: Proceedings of the second symposium on Usable privacy and security*, pages 32–43, New York, NY, USA, 2006. ACM.