INTENTION BASED AUTHORIZATION DIALOG BOXES

By

Marc Santa

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Computer Science

2012

Abstract

INTENTION BASED AUTHORIZATION DIALOG BOXES

By

Marc Santa

Throughout the computing experience, users frequently encounter security authorization dialog boxes in the form of Yes/No questions, which we call *Decision Based Authorization* systems. These systems are confusing to users who may not be technologically adept enough to make a correct "Allow" or "Deny" decision. In this paper, we propose an *Intention Based Authorization* system in which the user is presented with a list of intentions and he/she selects the intention that is closest to his/her actual intention. Only if the users intention matches that of the application, then the application is authorized to perform the action that it requested. We provide a detailed explanation of the process of generating these intentions. We conducted an internet-based survey of various computer users and found that experienced and inexperienced users preferred this new scheme. We also present a detailed analysis of the data collected through the survey and study the behavior of different types of users towards the dialog boxes.

Table of Contents

List of Tables

List of Figures

1 Introduction

Authorization dialog boxes are becoming more and more prevalent throughout our daily computing experience. When the user encounters an authorization dialog box, he or she is generally being prompted by some *moderator* (usually an operating system or website) that an application wishes to perform an action that could possibly be harmful to the user or computer. In the current model, the user is presented with some information about the potential consequences of allowing the action to proceed, and then asked to approve or deny the action in the form of a Yes/No question. In this paper, we call this existing model a decision based authorization (DBA) system. Figure 1 shows a generalized flowchart of the working principle of DBA systems.

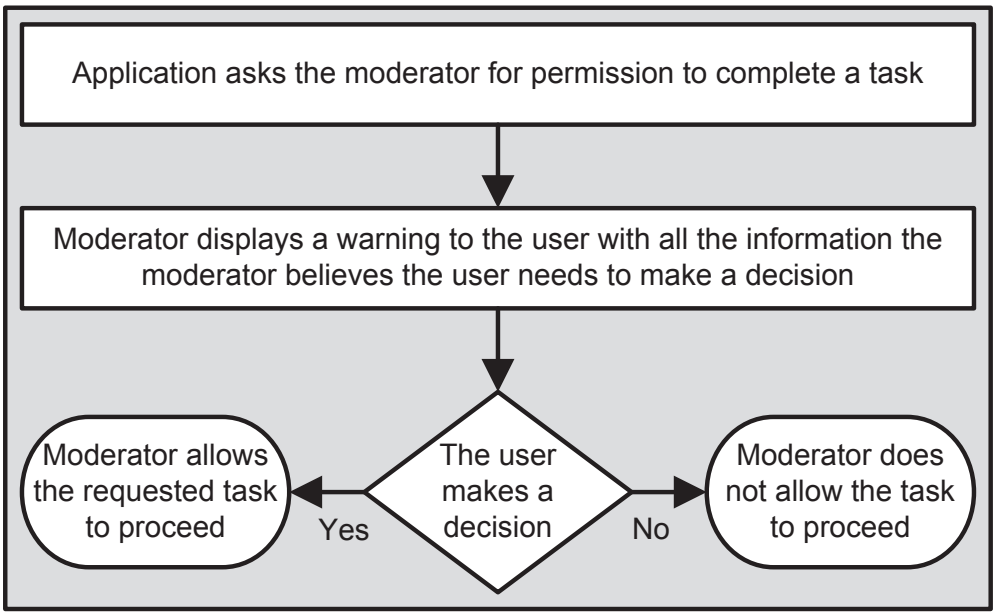


Figure 1: Flow Chart of Decision Based Authorization

1.1 Motivation

Computer users frequently encounter authorization dialog boxes throughout their daily activities. Figures 2-4 show examples of current decision based authorization dialog boxes.

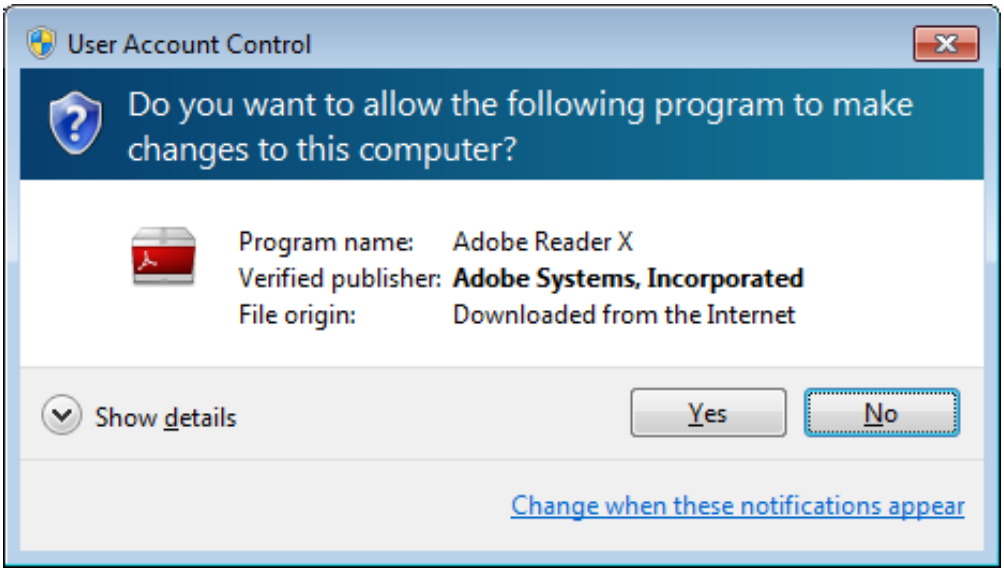


Figure 2: Windows 7 User Account Control. For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this thesis.

Figure 2 shows what a user of a clean installation of Windows 7 sees when he or she attempts to install Adobe Reader, a very common task. The user is informed that the setup application of Adobe Reader wants to make changes to his or her computer and that the application was downloaded from the Internet. The user, then, has to decide whether to allow or to deny the action by selecting Yes or No. Figure 3 shows Microsoft's web browser, Internet Explorer 8, informing the user about an invalid SSL certificate. Internet Explorer informs the user that the certificate that the website provided was for a website located at a different URL. The user must decide to "close this web page" or to "continue to this website". Internet explorer attempts to persuade the user to deny the action by marking the "continue" action as "not recommended". Figure 4 shows an example of a newer authorization dialog box, that of a Facebook application. Here, Facebook informs the user that the application named "@Smiles" is attempting to "access my basic information". The user must read the dialog box and choose to either "Allow" the application to access his/her basic information or "Leave App" to deny it the access.

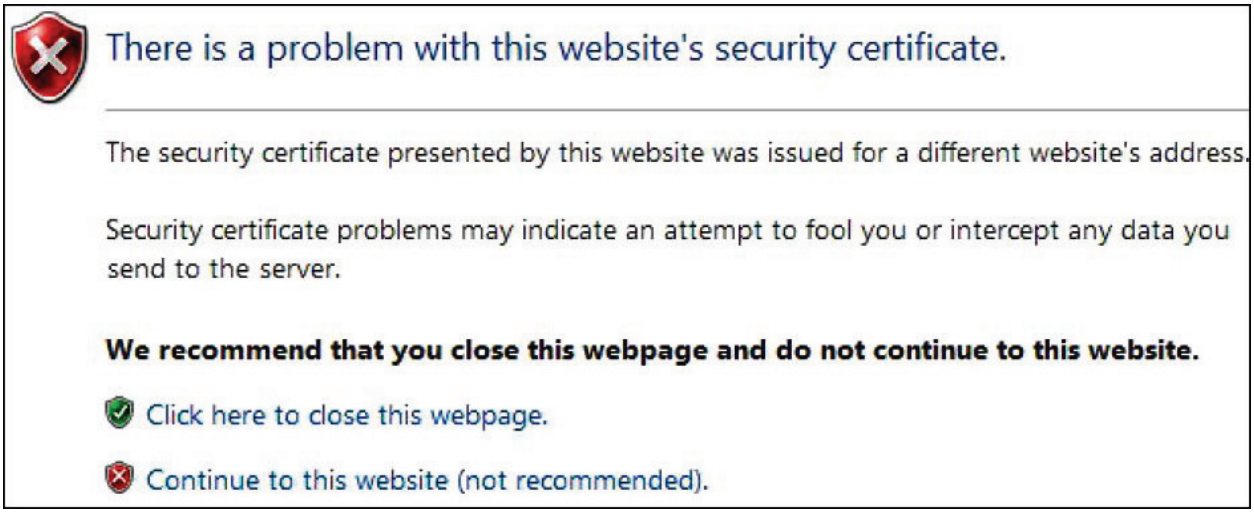


Figure 3: Internet Explorer 8 SSL Certificate Warning

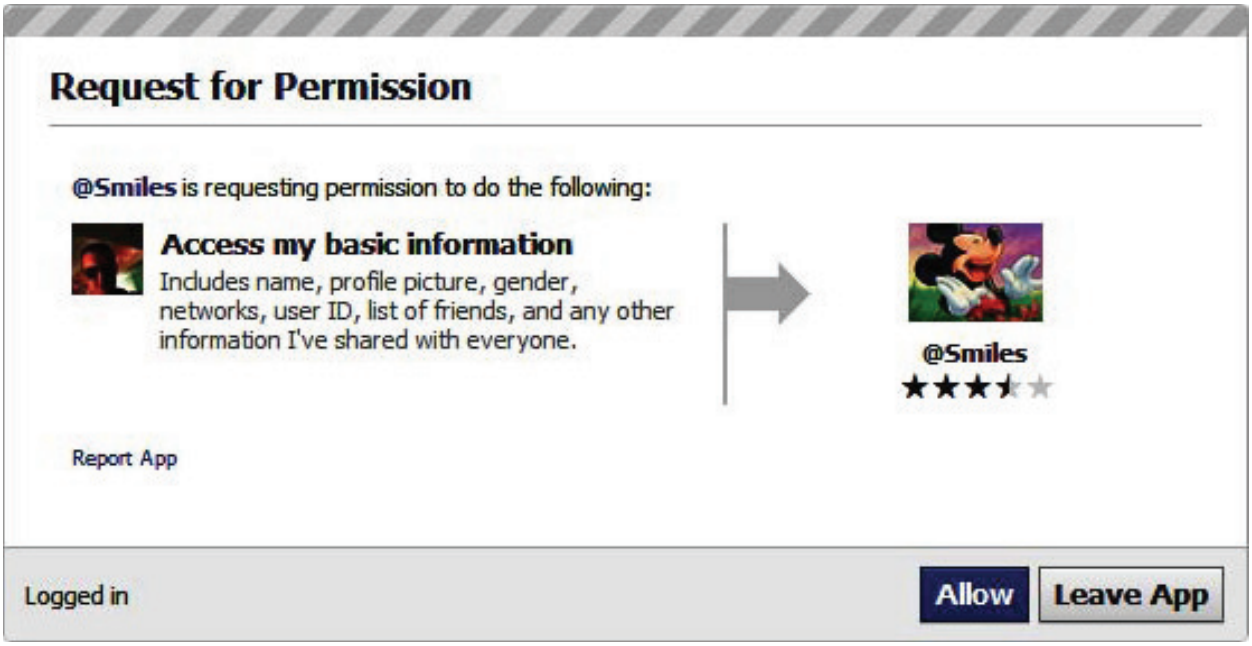


Figure 4: Facebook Application Installation

In the three examples given above and in many other scenarios, a typical computer user is not informed well enough to answer these questions correctly. For example, it is not unreasonable to state that most computer users do not understand what a SSL security certificate is and what it means if there is a problem with one. Even with a brief explanation, the user may not understand the consequences of their actions. Similarly, in the example given in Figure 4, it is possible that a user may not understand what accessing an application means and how his/her personal information could be manipulated. Users may not even understand what a Facebook application (or "app") even is.

Zurko *et al.*performed an analysis of one scenario in which Lotus Notes users were presented with a dialog box asking them if they would allow unsigned code from an e-mail message to run [7]. They found that 25% of the study participants would run unsigned code from an e-mail message without even being provoked to. Their conclusion best describes our motivation of the work presented in this paper.

"More research is needed in shielding users from having to make security-related decisions while still providing them with rich and flexible features . . . The study makes it clear that the common software practice of warning users of danger but letting them click on something to proceed anyway is not going to provide adequate security".

In this paper, we propose a novel idea of Intention Based Authorization (IBA) where, instead of asking the users a Yes/No question in a dialog box, the moderator asks the user to provide it with his/her intentions of what he/she is doing. The moderator allows the application to proceed if the intention of the user matches that of the application. This makes it much easier for the user since he/she only has to tell the moderator his/her intention instead of understanding some complex security message in a DBA dialog box.

1.2 Limitations of Prior Art

There has been little work in the area of improving authorization dialog boxes, and none of it is as comprehensive as the solution we propose. Web Wallet, proposed by Wu *et al.*, is a web browser plug-in that requires the users to enter sensitive data into itself instead of directly onto a web page [6]. Web Wallet is limited to web pages only, and can be easily subverted by malicious users or applications. Brustoloni and Salomn tackled authorization dialog boxes in general and proposed a solution in which the user is presented with series of dialog boxes in order for to make a Context-sensitive guidance (CSG) decision.[1]. Their system is limited because intentions must be hard-coded; they cannot be dynamically generated. Their system also relies on system administrators reviewing audit logs, which our system does not require.

1.3 Our Approach

In the existing system, the moderator presents the user with some amount of information about application's action and then asks a Yes/No question. In IBA, we ask the user to tell us what he/she is trying to do and then see if it matches with what the application is trying to do. Because speech recognition is not up to par, and recognition and processing of typed text has accuracy problems, we must come up with a way for the user to tell us his or her intention. We already know what the application's intention is (either inherently or by querying it), so we can present the user with a multiple-choice list of intentions of what he/she is trying to do. Only one out of these intentions will match the intention of the application and the rest will be generated by the moderator. The order of the multiple-choice list will be randomized each time to prevent training. If the user selects his/her intention to be the one that matches that of the application, the moderator will allow the application to perform it's requested task. If the user selects any other option on the multiple choice list, or exits out of the dialog box in any way, the moderator will not allow the application to

execute it's requested task.

1.4 Key Contributions

In this paper, we make following three contributions: (1) design and implementation of Intention Based Authorization (IBA) dialog boxes, (2) Internet based survey taken by 93 participants (including experienced and inexperienced users) to evaluate the prototype IBA dialog boxes in comparison with DBA dialog boxes, and (3) detailed analysis of the data collected from the survey to develop insights into user perception towards both types of dialog boxes.

2 Related Work

In the past ten years, there has been little work attempting to understand and improve authorization dialog boxes. Wu *et al.*introduced web browser security software named Web Wallet [6]. Web Wallet disables HTML forms in the browser and requires the users to enter data directly into the Web Wallet software. Web Wallet then investigates the security of the website and attempts to assist the user in determining if it is safe to submit the information or not. Instead of presenting a Yes/No dialog, Web Wallet presents a list of many websites including the current website and asks the user, "Which website are you trying to submit this information to?". If the user selects the current website, then the submission is allowed to proceed. This solution is limited in the sense that it relies on a web browser plug-in which can be disabled by the user, other users, or malicious software.

Brustoloni and Salomn introduced several improvements to standard Yes/No security dialog boxes [1]. The first improvement is polymorphic dialog boxes in which the layout of the dialog box changes each time the dialog is opened, forcing the user to thoroughly examine the dialog before selecting an option. Next, they present the user with a series of dialog boxes in order to gain enough information to make a context-sensitive guidance (CSG) decision. Based on the users selections, the action may or may not proceed. These CSGs must be hard coded for each application. Their scheme is different from ours in that the user must respond to a series of questions, not just one. Also, every application must be extensively modified in order to support this design. Furthermore, their system always provides an obvious option for the user to allow the action. In contrast, IBA asks the user to provide his/her intention to the moderator and the moderator decides whether or not to let the application proceed.

Finally, their system warns the user that their action may be audited. The answers to security questions are logged and an administrator can review individual user's past responses. This encourages the user to make informed decisions. IBA can generate lists

of intentions dynamically; they do not have to be hard coded into the application. Their system also relies on the system administrator reviewing audit logs in order to extract the full potential of their solution. IBA has been designed so that system administrators do not have to review any logs because in many settings there are no administrators available *e.g.*, in the case of home users.

Cao and Iverson introduced an intention based access management system called IAM [4]. IAM assists the administrator in assigning permissions to objects. Instead of requiring the administrator to assign each user read, write, or execute properties to every object, the administrator, instead, provides his or her intentions in plain English. These intentions are expressed in sentences of the form "`User X [Must Have / Must Not Have] [Privilege] to [Object]`". After a series of intentions have been expressed, the system computes the access control list for the user and informs the user if there are any conflicts. This system is different from ours in purpose and functionality in that it uses intention based questions to generate access control lists, not to provide answers to on-the-spot authorization dialog boxes.

3 Intention Based Authorization Dialog Boxes

Intention Based Authorization is an interaction between a user, a moderator, an application requesting authorization and the task being requested. We define these terms as:

User: The human user of the computer system

Moderator: The application or operating system that is controlling security for the computer or any other device the user is using. All security events must go through this moderator in order for the system to be secure.

Application: An application running on the user's computer or device that is requesting authorization to perform a task that it normally would not be allowed to do. The request from this application will go through the moderator and the moderator, in most cases, will prompt the user to authorize this request.

Requested Task: The task the application wants to perform but cannot without first asking permission of the moderator.

Figure 5 shows an example of a DBA dialog box. In this example, the application is named "WordHoard" and the moderator is the Java Runtime Environment (JRE) installed on a Windows based personal computer. The application has asked for the permission to execute on the system. The moderator (JRE) presents the user with a dialog box asking to authorize the requested task *i.e.*, grant or deny permission to execute WordHoard on the system. If the user clicks on "Run", WordHoard will execute. If the user clicks on "Cancel" or closes the dialog box, the WordHoard application will not be allowed to execute. Now we present the theory of IBA and see how it is different and advantageous over DBA.

3.1 Theory

In decision based authorization when an application wants to perform a secure task the moderator presents the user with a description of the requested task, some amount of advice,

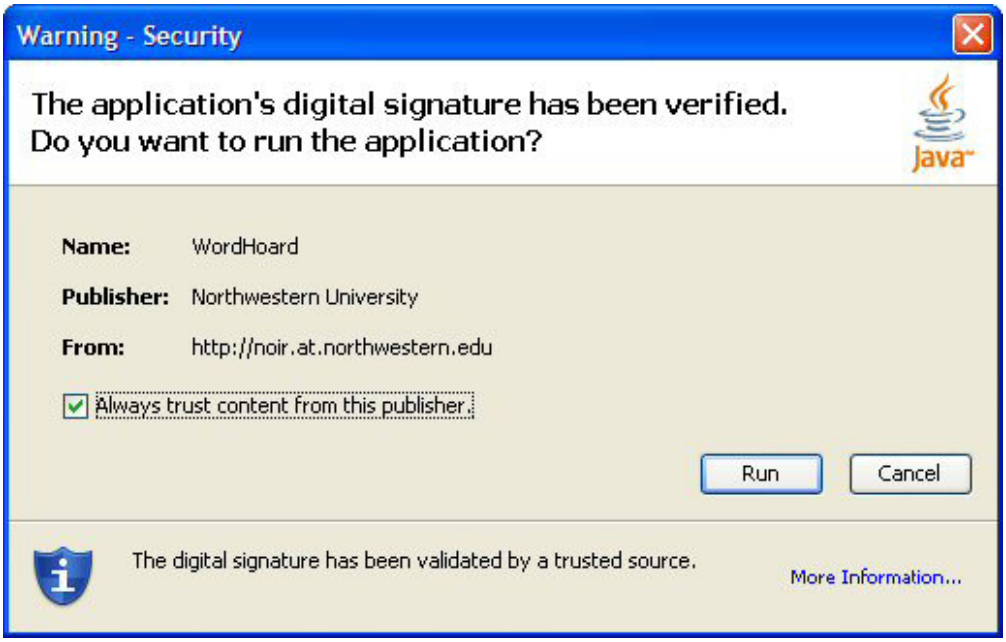


Figure 5: Example of a Decision Based Authorization Dialog Box

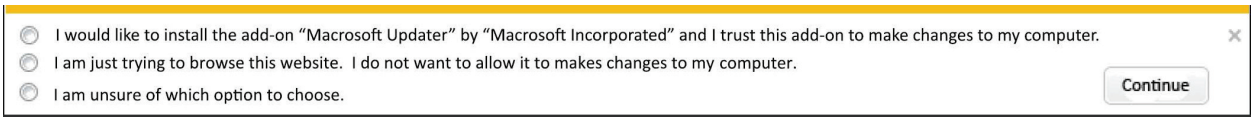


Figure 6: Internet Explorer 9 plug-in Installation, IBA version

and then some form of a Yes/No question as seen in Figure 5. With IBA, the user presents his/her intentions to the system. If the user's intentions match with what the application or website is requesting permission to do, then the action proceeds. If the user's intentions do not match that of the website, the user is informed why the action can not be allowed. It is important to note that it is the user who tells the moderator about the intentions of his actions and not the moderator which tells the user about the intentions of the application.

Because most computer users are not informed well enough to adequately explain their intentions [7], we have designed a multiple choice system that allows the user to select from a list, presented to them, of possible intentions. The action the application is requesting authorization for is presented in the list, along with several other similar but incorrect intentions. If the user selects the intention that matches that of the application, then the application is

authorized and the action proceeds.

In IBA, the moderator displays a list of n intentions to the user, with n being determined by the moderator's developer. The moderator will either generate random intentions or use pre-configured similar intentions. The list of intentions is randomized each time to prevent the user from being conditioned into clicking into a similar location each time. Figure 7 shows a generalized flowchart of an IBA system.

3.2 Generation of Intention Choices

To present the user with a list of intentions, the moderator must first generate these intentions. In some cases, the moderator can generate the list of intentions without any input from the application while in other cases, the application and the moderator must work together in order to generate them. We present two methods of generating these intentions: (1) Inherent Method and (2) Interrogation Method.

3.2.1 Inherent Method

For some requested tasks, it is inherent what the user's intentions are. For example, if the user is prompted to install a web browser plug-in, then he/she either: (1) intended to install a plug-in, (2) did not intend to install a plug-in, or (3) is unsure about installing a plug-in. In this situation, the moderator can display all three of these intentions worded in such a way that they are clear to all users. The application inherently wants to install the plug-in, so only if the user also wishes to install the plug-in will the action proceed.

Figure 6 shows a prototype of an Internet Explorer 9 plug-in installation dialog box. In this case, the user has navigated to a web page that wishes to install a web browser plug-in. The requested task is inherent: the web page wishes to install a plug-in. All of the intentions shown in Figure 6 would be hard coded into Internet Explorer (the moderator in this case) and shown each time a website tries to install a plug-in. If the user selects the first intention,

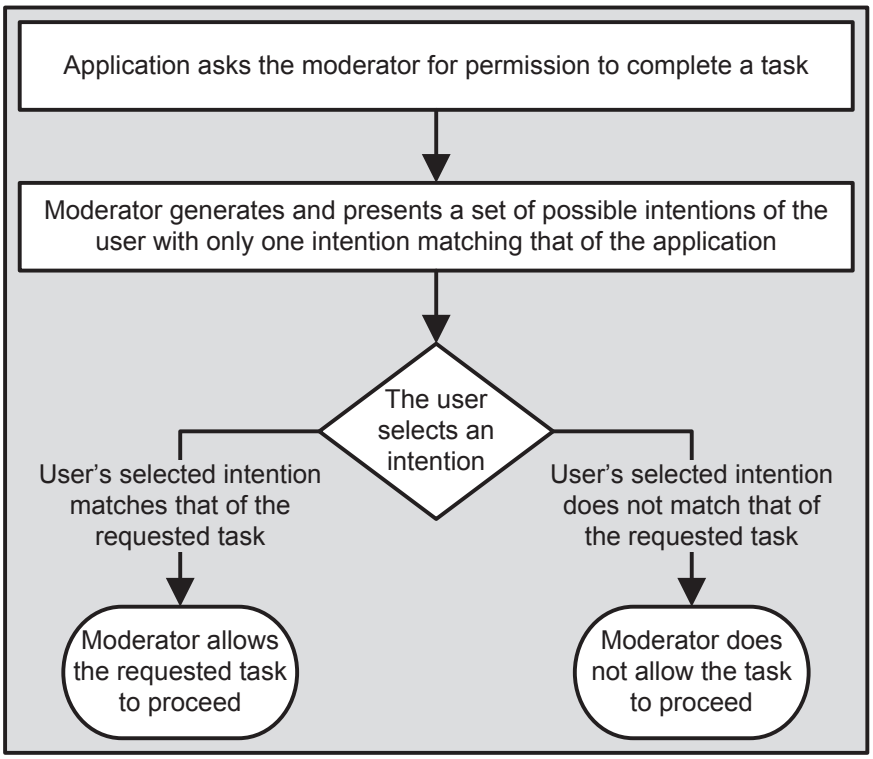


Figure 7: Flow Chart of Intention Based Authorization

that he or she trusts the plug-in, then the moderator will allow the plug-in to install. If the user selects any other option or closes the dialog box then the moderator will not allow the plug-in to install.

The inherent method is suitable when the requested task is obvious and the list of possible intentions is small. The moderator can choose to display all possible intentions or a subset of them in a randomized order.

3.2.2 Interrogation Method

With the interrogation method, the moderator supports a fine grained list of possible requested tasks and maintains a list of textual descriptions of all of them. Some moderators, such as Facebook, already do this *i.e.*, it requires the applications to provide it with a set of actions that the application wants to perform, not just a generic elevation request. [2].

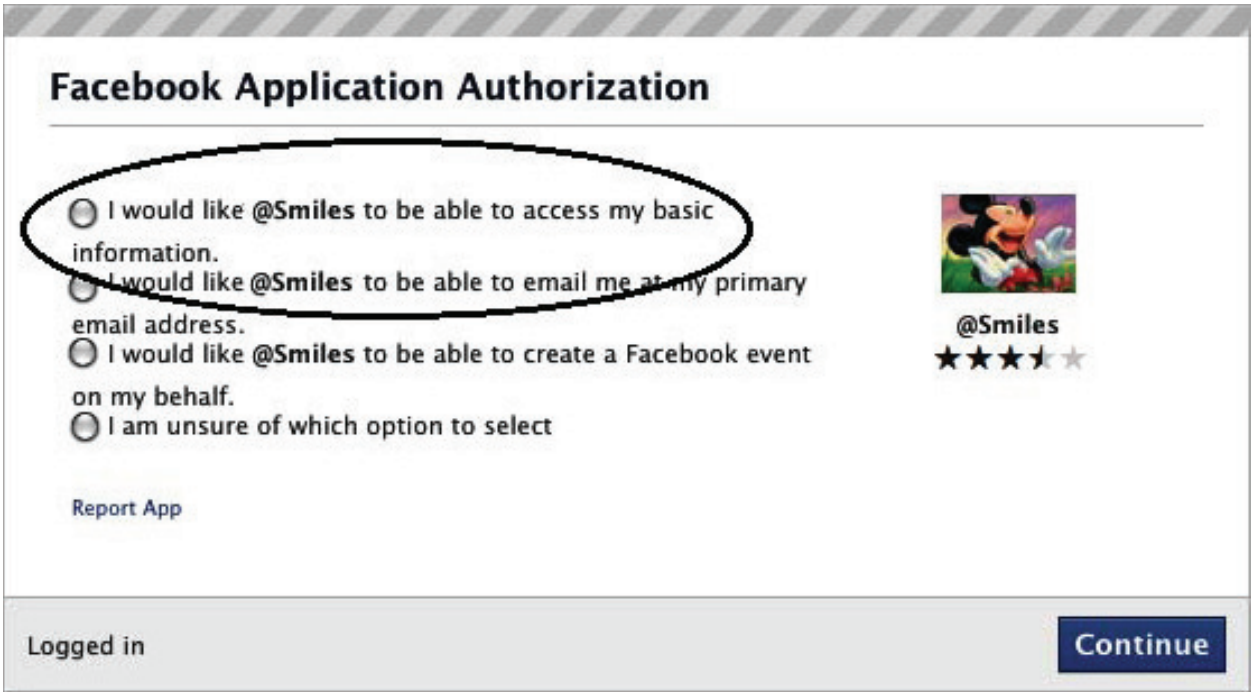


Figure 8: A Prototype Facebook Application Dialog Box.

The moderator always knows every task that is possible due to its design. For example, if the moderator is Microsoft Windows and the application is requesting to make changes to the hard drive of a computer then Windows already knew that an application could request this. Windows also knew that an application could request registry access, network access *etc.*When a moderator is modified to support IBA, the programmer must provide text descriptions of all possible tasks that an application can ask for. The moderator will then have a library of text descriptions for all possible requested tasks. This way, the moderator can present the description of the requested task, as well as other decoy tasks, to the user when it presents him/her with the list of intentions.

As an example, consider Figure 8. The application, @Smiles, has requested permission to access the user's basic information. In this example, instead of showing a DBA dialog box as in Figure 4, facebook should display the IBA dialog box shown in Figure 8. The first sentence (circled), is a description of the requested task pulled from Facebook's list of all

possible requests [2]. The other three intentions are randomly selected from the same list. If the user selects the first option, then Facebook should allow the requested task (accessing the user's basic information) to proceed. If the user selects any one of the other three options, the requested task should not be allowed to proceed because the user's intention and the application's intention do not match.

4 Case Studies

We now provide three case studies in which we convert existing security dialog boxes to IBA. In each case study, we present an existing DBA dialog box and describe the problems associated with it. Following this, we present the conversion of the existing dialog box to IBA. Finally, we discuss how the new IBA dialog box solves the problems associated with the existing one.

4.1 Web Browser SSL Certificate Error

A web browser SSL certificate error occurs when a user visits a website that uses SSL with certificates but there is some sort of error with that certificate, *e.g.*, the certificate could be expired, for a different URL, or not in the user's trust chain. If there is an error with the certificate, it is up to the user to decide whether to continue to the website or not. It is possible that the website, the user's computer, or some other computer along the way has been tampered with.

The current version of these dialog boxes present too much technical information to the user and, thus, most users are unable to make an accurate decision. SSL certificate errors can occur frequently depending on user's browsing habits. Because these dialog boxes are confusing, users become conditioned into not reading them and simply selecting "continue". This leads to users sending confidential personal information to untrusted websites.

Figure 9 shows an existing implementation of an SSL certificate error in Mozilla Firefox 4 web browser. In this example, the user has navigated to `https://www.cacert.org` which uses SSL security. The certificate presented at this URL has been signed by the website itself, which is not trusted by the user's computer. Firefox presents the user with the dialog box shown in Figure 9 and explains the error in the first two sentences and then suggests that someone may be attempting to impersonate the website. If the user selects "Get me out

Figure 9: Mozilla Firefox 4 SSL Certificate Warning

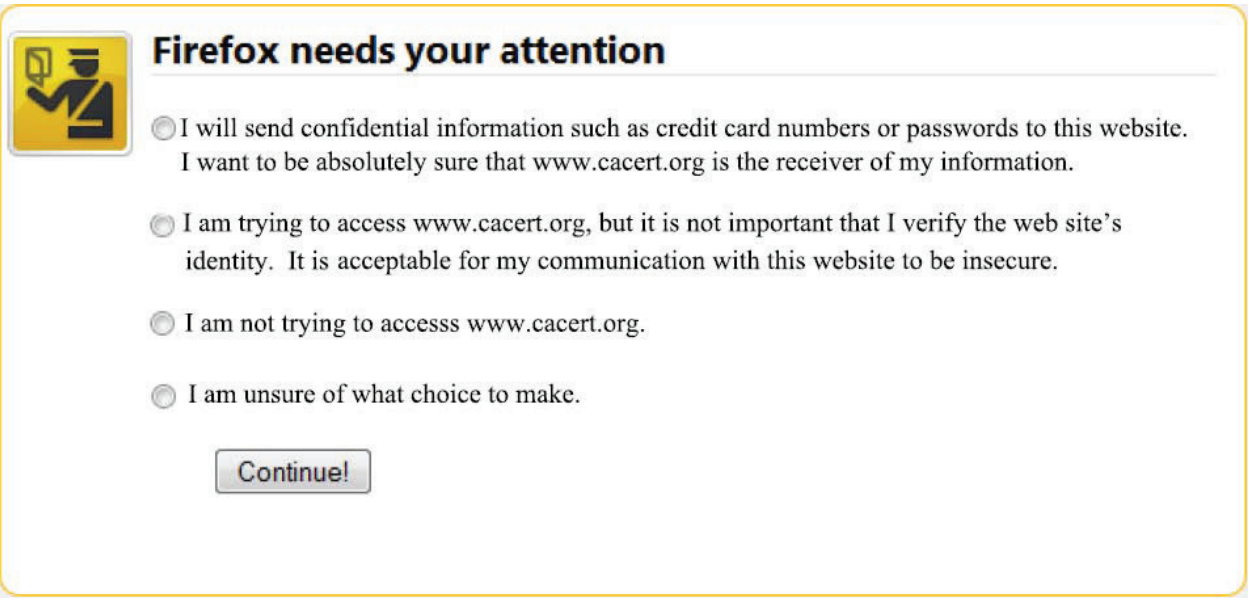


Figure 10: Mozilla Firefox 4 SSL Certificate Warning converted to IBA

of here!", he or she will not be allowed to continue to the website. If the user expands the selection labeled "I Understand the Risks" and then adds the certificate to his trust chain, he or she will then be allowed to browser to the website.

The first step in fixing this dialog box is to enumerate the intentions of the user. Four of the several possible inherent intentions are: (1) the user plans on entering personal information to the website, (2) the user doesn't plan on entering personal information to the website, (3) the user didn't intend on visiting this website in the first place, maybe he or she clicked on the wrong link, or this error is because of a pop-up advertisement window, or (4) the user doesn't know at all how he/she reached this website and what to do.

Figure 10 shows an IBA dialog box with these 4 intentions written in a user friendly manner. If the user selects the second option, he will be allowed to proceed to the website because he understands the risks involved in visiting the website. If the user selects the first intention, he will not be allowed to proceed to the website because there is a certificate error and he cannot be sure of where his personal information will go. If the user selects the third or fourth options he will again not be allowed to visit the website because he either didn't intentionally visit the website, or he is unsure of why he is here.

With these changes we have transformed the workflow of this dialog box. Earlier, the moderator (Firefox) attempted to educate the user enough to allow him/her to make the decision; now, the user selects from a simple list of intentions and the moderator makes the decision for him/her.

We have implemented this design in Firefox 4. Source code and binaries (for various platforms) are available here: `URL Removed for anonymity`.

4.2 Software Installation

A common day-to-day task is the installation of software. Modern operating systems have built-in software security that prevents executables from making changes to a computer's

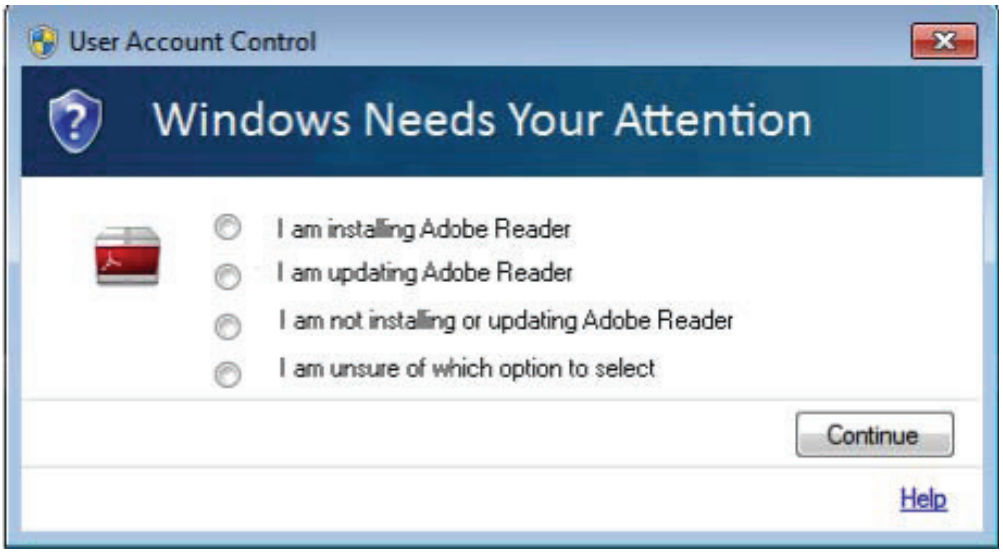


Figure 11: Windows 7 UAC converted to IBA

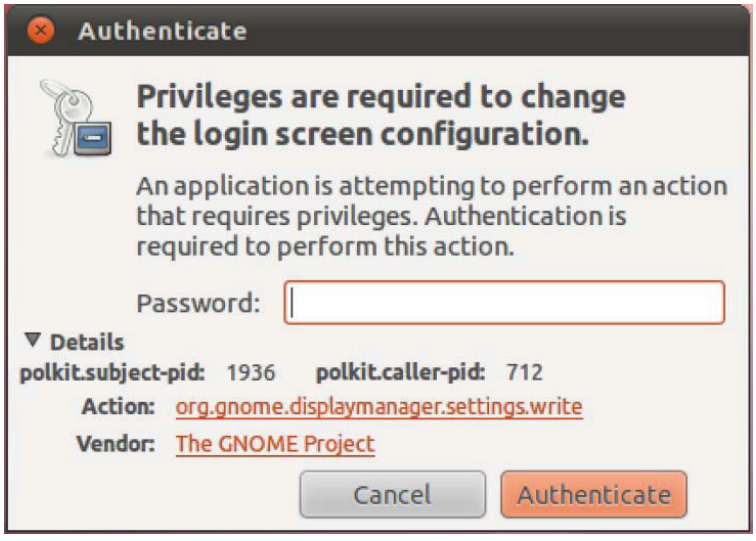


Figure 12: Ubuntu Linux System Configuration Change Dialog Box

filesystem or other data storage locations. When a user attempts to install software on these modern operating systems, the OS acts as a moderator and displays an authorization dialog box asking the user if it is acceptable for the installation executable to make changes to his/her computer.

Figure 2 shows an example authorization dialog box that a user would see on a Microsoft Windows 7 computer when he/she attempts to install Adobe Reader. This security mechanism is branded as User Account Control (UAC) by Microsoft. The dialog box informs the user that an executable is requesting permission to make changes to user's computer. If the user selects "Yes", Windows (the moderator) allows the software to make changes to the user's computer. If the user selects "No", Windows does not allow the software to make changes to the user's computer.

These dialog boxes are problematic for several reasons. The first reason is that they do not inform the user what changes will be made. The user must make a decision based on his own knowledge about software installation. If a novice happens to encounter this dialog box, he/she may not know what the dangers of allowing the changes are. This dialog box does not provide any information to the user about situations in which it is acceptable to click "Yes" and those in which it is necessary to click "No".

We designed our IBA dialog box to take the decision making process away from the user. We chose to use the inherent method of selecting intentions in order to avoid having all Microsoft Windows application re-written to support interrogation. Our version of the dialog box makes the assumption that if an executable is attempting to make changes to the user's computer then this executable is trying to install or update a software.

Figure 11 shows Microsoft User Account Control redesigned to support IBA. In this example, the user is again attempting to install Adobe Reader. If the user knows that he is installing software, he can select the first option and the moderator (Windows) will allow the software installation to continue. If the user is updating software then he can select the

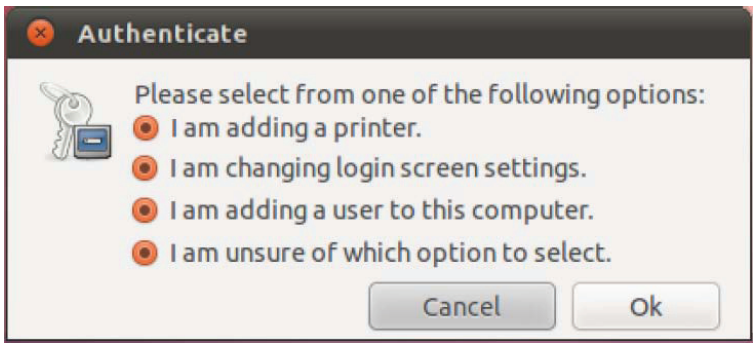


Figure 13: Ubuntu Configuration Change Dialog Box Converted to IBA

second option and Windows will also allow the application to execute it's requested task. If the user is not attempting to install software or is unsure, then Windows will not allow the installation to continue.

4.3 System Configuration

System configuration dialog boxes appear when the user tries to make a configuration change to his or her computer or other device. In modern operating systems, users are not allowed to make configuration changes without first "elevating" to a super-user level. This is either accomplished by simply approving the elevation or by re-entering the user's password. System configuration dialog boxes are important because they prevent rogue users and software programs from making changes to the user's computer without his/her permission.

Figure 12 shows an existing system configuration dialog box of Ubuntu Linux 11.04. In this dialog box, the user is attempting to change settings about the computer's login screen that is presented when the computer turns on. In Figure 12, the user is informed that "Privileges are required to change the login screen configuration". The user can either enter their password and then click on "authenticate", or not enter their password and click "cancel". If the user enters the password and is successfully authenticated, he/she will be allowed to change the configuration of the system's login screen. If the user does not get

authenticated, he/she will not be allowed to change the login screen.

At the bottom of the dialog box shown in Figure 12, we can see that Ubuntu (the moderator in this case) already has plenty of information about the requested task of the application. We can see the moderator's PID, the Applications PID, UID of the requested task, and the vendor of the application.

This dialog box is problematic for several reasons. First, users are not adequately informed that there is a choice to be made. The dialog box implies that they must authenticate. It is up to the user to deduce that he doesn't necessarily have to enter this information. Second, there is no definition of what "privileges" are. Users are not even informed of which password they are being asked of.

In this situation we can use the interrogation method because Ubuntu Linux already has a built-in method to specify what the requested task is when an application requests to elevate. The first step that we took was to create a pool of descriptions of intentions for the requested task. For example, the requested task in Figure 12 is known internally to Ubuntu as `org.gnome.displaymanager.settings.write`. Inside the Ubuntu moderator, we would have a database of mappings from internal names to textual descriptions. Our example requested task could be described as "I am changing login screen settings". This would require that only the moderator be modified.

Figure 13 shows the Ubuntu security dialog box converted to IBA dialog box. The new dialog box selects from a pool of intention descriptions: It displays the correct intention, "I am changing...", and three incorrect intentions. In Figure 13, The first and third options are incorrect. Selecting either of these options, clicking on cancel, or closing the dialog box will not allow the requested task to proceed. The second option matches the intentions of the application. Selecting the second option will allow the requested task to proceed. Selecting the fourth option will not allow the requested task to proceed because the user is unsure (or his/her intentions were not in the list). If the user selects the intention that matches that of

the application, the moderator will ask the user for his or her password in a new dialog box before it allows the requested task to execute.

5 Evaluation

5.1 Methodology

We conducted an internet-based survey of 93 participants to study four different scenarios to assess IBA in comparison to DBA. These four scenarios are:

(1) The user is attempting to install software (Adobe Reader). In this case, the end goal is for the user to successfully install the software.

(2) The user is attempting to play an online game on a website when a malicious web browser add-on attempts to install itself. In this case, the end goal is that the malicious add-on does not get installed.

(3) The user is attempting to log-in to a website using SSL that has an invalid certificate. In our survey, the user is attempting to access their university webmail and their session has been compromised. In this case, the end goal is that the user does not pass any personal information to the website.

(4) The user is attempting to access a Facebook application and wants to make sure that an optimal amount of personal information is shared.

For each of the four scenarios above, we presented an existing DBA dialog box and an IBA dialog box. The survey participants were asked what action they would take when faced with each different dialog box and how confident they were of their choice. After each scenario, the participants were asked if they preferred DBA dialog box or IBA dialog box.

The only demographic data that we collected was one question intended to gauge whether or not participants were computer experienced or not. The question asked the participants if they had ever installed an operating system on a computer, or if they felt they were comfortable configuring computers. Participants that responded "yes" to this question were marked as computer experienced and the participants that responded "no" were marked as computer inexperienced.

Each scenario has 5 questions. We represent them as $\texttt{S}x\texttt{Q}y$, where x represents one of the 4 scenarios ($1 \leq x \leq 4$) and y represents the specific question of that scenario ($1 \leq y \leq 5$). Question 1 of each scenario asks the participant what option would he/she select for DBA and Question 2 asks how confident the participant was about the answer to Question 1. Question 3 of each scenario asks the participant what option would he/she select for IBA and Question 4 asks how confident the participant was about the answer to Question 3. Finally Question 5 asks the participant whether he/she likes IBA or DBA. The full text of the survey can be found in the Appendix.

In the next section, we first present a univariate analysis of the data we collected through the survey. For this, we consider each of the 4 scenarios individually and see whether the the users prefer DBA or IBA. Then we see what the experienced users prefer between DBA and IBA for each scenario and how the inexperienced users think. Following this, we study which of the two schemes enabled the participants to take more correct decisions compared to the other.

Following this, we perform multivariate analysis on the collected data using WEKA [5]. Specifically, we first cluster the users into groups and study the properties of the users belonging to each group. To cluster the users, each question is considered an attribute except those questions which ask the users whether they prefer IBA or DBA for the given scenario. We do not use these questions as attributes because they formulate the ground truth in our univariate analysis and eventually we will compare the preference of users of each cluster with this ground truth. Thus, if these values are used, we will essentially be biasing the clustering because we use k-means clustering which is an unsupervised clustering algorithm. We use hamming distance as the distance measure to minimize the the sum of the within-cluster sums of point-to-cluster-centroid distances over all clusters. We term the sum of the within-cluster sums of point-to-cluster-centroid distances over all clusters as intra-cluster dissimilarity. Note that all our attributes are nominal and thus hamming distance is the

only feasible distance measure in this case. To determine the optimal number of clusters, we utilized the intra-cluster dissimilarity (d_i) where i represents the number of clusters. We increased the number of clusters from 1 onwards until $d_i - d_{i-1} \rightarrow 0$. Such a value of i shows that increasing number of clusters from $i-1$ to i did not decrease the intra-cluster dissimilarity, and thus having $i-1$ clusters is most optimal. We then take each cluster and study the preference of experienced and inexperienced users and correlate with our ground truth.

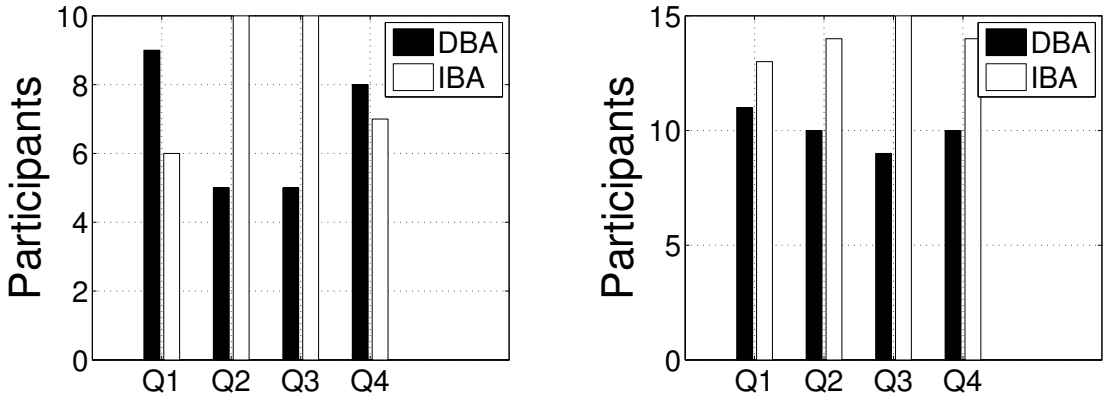
Following this, we perform association rule mining to extract interesting association rules about user behavior towards IBA and DBA. We used implementation of Apriori association rule mining algorithm in WEKA to extract the rules with confidence greater than 95% [3, 5]. For association rule mining, we generated rules by using various attributes as classes one by one and present some interesting findings. The attributes that we used as class attributes include user experience level and the four questions, one for each scenario, asking whether the user liked DBA or IBA. To obtain interesting rules, we used a maximum support of 0.2 and confidence level of 99% as thresholds.

5.2 Results: Univariate Analysis

We collected 93 total responses, with 54 participants identifying themselves as computer experienced, 26 as computer inexperienced, and 13 not answering.

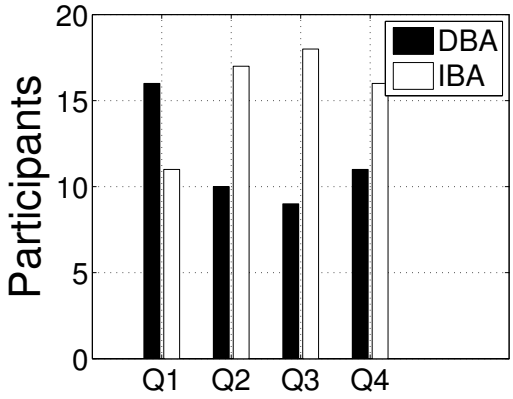
Figure 15(a) shows that in three out of the four scenarios, participants preferred IBA dialogs versus DBA dialogs. Scenario one was the only scenario where more participants preferred DBA over IBA. Figure 15(b) shows that in scenario 1 computer experienced participants preferred the existing DBA dialogs over the proposed IBA dialog. Computer inexperienced participants preferred the IBA dialogs in all of the scenarios.

Scenarios 1 through 3, each had a "correct" answer in the sense that the participants were asked to perform a task. If the participant selected a response that completes the

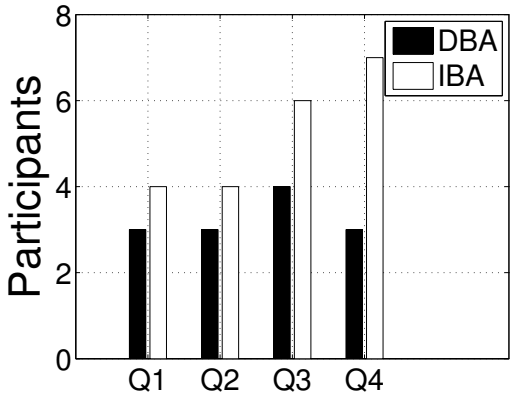


(a) Cluster: VCC

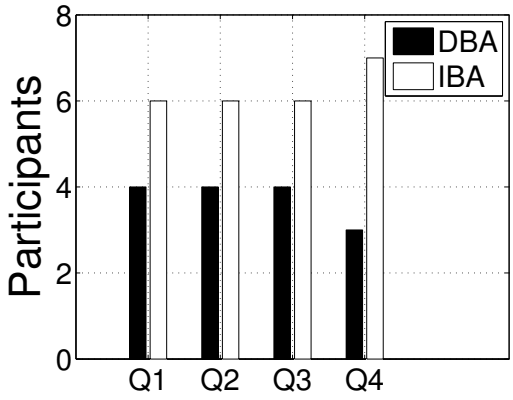
(b) Cluster: FCC



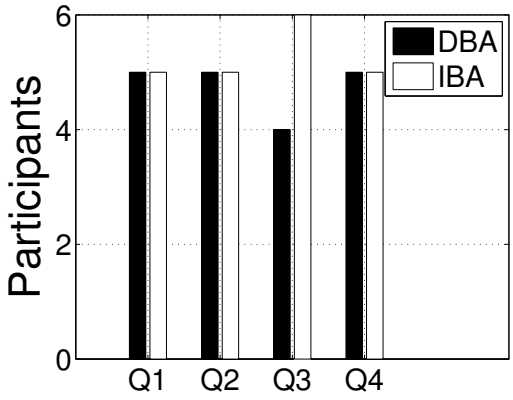
(c) Cluster: VCN



(d) Cluster: NC

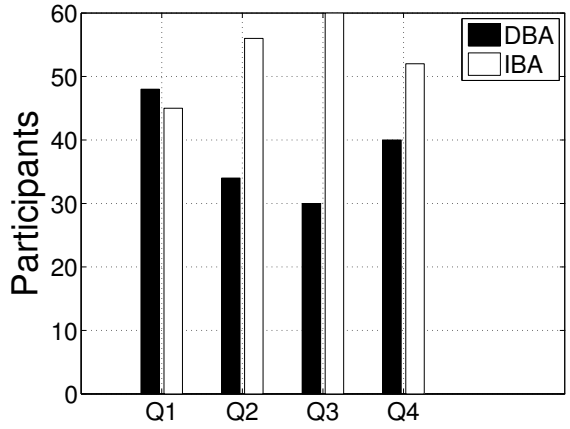


(e) Cluster: VCiC

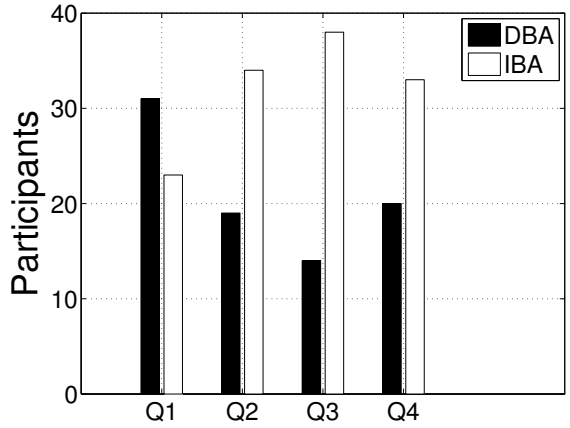


(f) Cluster: Misc

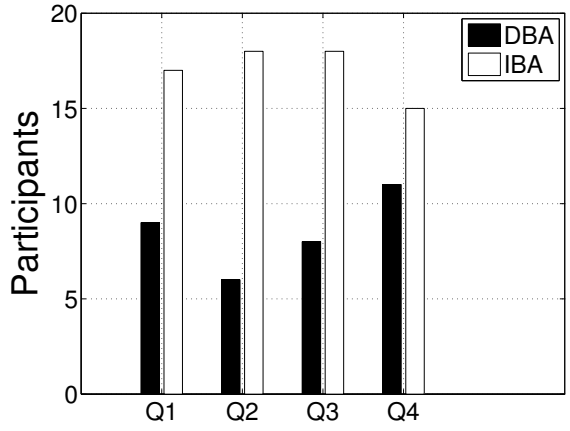
Figure 14: Distribution of number of participants of each cluster for each scenario that selected IBA and DBA dialogs



(a) All Participants



(b) Computer Experienced



(c) Computer Inexperienced

Figure 15: Visualization of how experienced influenced survey answers

assigned task without putting his/her security at risk then this was considered a "correct" answer. Figure 16 shows this data. For question 1 (Adobe Reader installation) and question 2 (malicious browser plug-in installation), the ratio of correct to incorrect responses was about the same. For question 3, the SSL certificate error, 40% of respondents chose to avoid accessing a malicious site when using the existing DBA warning dialog. 74% of participants selected options in the IBA dialog that would lead to them leaving the site.

Scenario 4 had no "correct" answer but instead was intended to gauge whether or not participants would allow access to more or less personal information on Facebook when faced with a DBA or IBA dialog. Figure 17 shows that with both methods, the number of participants who allowed Facebook to send "Basic Information" to the application was about the same. These results are totals based on whether or not their responses would have the moderator (Facebook) give access of the basic information to the application.

5.3 Results: Clustering

Figure 18 plots the difference between intra-cluster dissimilarity of consecutive clusters. The bar above a given number i on x-axis gives the intra-cluster dissimilarity between i clusters and $i+1$ cluster. As it can be seen, the bar above 6 is almost equal to 0 showing that having 6 clusters or 7 clusters results in same value of intra-cluster dissimilarity. Thus, having 6 clusters is most optimal.

We first present some insights into the clusters that we obtained and later discuss how the experienced and inexperienced users belonging to each clusters prefer IBA or DBA.

Cluster 1: this cluster contains those people that declared themselves as highly confident about their choices. We can consider this group to be representing those users that are cautious about the choices they make. For example, although declaring themselves to be very confident about what they are doing, they closed the windows altogether in scenario 2 and scenario 4, probably to protect their personal information and computing device. We call this group of users to be "very confident and careful (VCC)". This cluster contains 16.1% of all the users.

Cluster 2: this cluster contains those people that declared themselves as fairly confident but not highly confident about the choices they made. They seem to trust the famous applications like Adobe and university webmail but, like VCC, close the windows all together in scenarios 2 and 4 where the applications are not very trustworthy. We call this group "fairly confident and careful (FCC)" This cluster contains 25.8% of all the users.

Cluster 3: this cluster contains users that declared themselves highly confident about the choices they make but at the same time never deny any application from anything in either

Table 1: Distribution of experienced and inexperienced users in clusters

	VCC	FCC	VCN	NC	VCiC	Misc
Exp	12	16	22	3	6	8
InExp	3	8	5	4	4	2

IBA or DBA. They allowed Adobe to install in Scenario 1, they allowed the website to install the add-on in Scenario 2, they proceeded to the university mail in Scenario 3 even though the security certificate was invalid, and they allow @Smiles to access the basic information in Scenario 4. Since, they allow any application to do anything, they are not cautious users. We call this group of users to be "very confident but negligent (VCN)". This cluster contains 29% of all the users.

Cluster 4: this cluster contains users that declared themselves to be not confident about all the choices they make and chose to close the window in case of web browser and dialog boxes in case of applications. This behavior can either be because the users did not understand at all what the application was trying to do or they are too cautious about every thing. We simply call this group of users as "not confident (NC)". This cluster contains 7.5% of all the users.

Cluster 5: this cluster contains users that declared themselves to be very confident and correctly allowed Adobe to continue with the install with DBA as well as IBA but chose to close the web pages in Scenarios 1, 2, and 3. This shows that this group of users is more cautious on internet. We call this group of users to be "very confident and internet cautious (VCiC)". This cluster contains 10.75% of all the users.

Cluster 6: this cluster contains users whose confidence levels range from not confident to highly confident in different scenarios. This group of users makes incorrect choices using DBA but never makes incorrect choices with IBA. At most, the users leave the web page all together in some cases but never selects an incorrect choice through IBA. We call this group to be "miscellaneous' (Misc)'. This cluster also contains 10.75% of all the users.

Table 1 gives the distribution of experienced and inexperienced users for each of the 6 clusters. We can see from this table that VCC and VCN are the clusters dominated by the experienced members. It coincides with our earlier observation that the users belonging to these clusters were very confident about their choices. From Figures 14(a) and 14(c), we can

see that in both these clusters, users generally like IBA for scenarios 2 through 4 especially 2 and 3. A plausible reason is that even the users who consider themselves to be experienced may not be familiar with the SSL certificates and the way add-ons work. Since IBA gives more elaborate details of the task the application is trying to do along with matching tasks for the user to understand the difference, the experienced users of these two clusters prefer IBA. For Adobe Reader, the updates come so frequently that users especially the experienced ones are so used to of this dialog box that they prefer it over IBA. Similarly, the experienced users are likely to spend more time on Facebook since many of them belong to the computer science department, therefore, the conventional design of Facebook might feel better to them since they are very used to it. That is why we see kind of a mixed response from members of experienced users clusters about IBA and DBA for @Smiles application.

In the FCC cluster, as can be seen from Table 1, although there are more of experienced users, they are not that larger enough than inexperienced users to let us to call the FCC cluster to be experienced user dominant cluster. We also see that although some of the users declared themselves to be experienced, they may be somewhat in between experienced and inexperienced. Since we have only two options in our survey, some of them might have selected experienced and some selected inexperienced. But since we see that the users said that they are only fairly confident about their choices and not highly confident, we can assume that users belonging to this cluster are not as experienced as those in VCC and VCN. In this cluster, more users prefer IBA over DBA in all scenarios as can be seen from Figure 14(b).

NC cluster contains only 7 users with very unique behavior. These users tend to close the dialog box or web page whenever they encounter such a question, be it IBA or DBA. This cluster is an outlier which contains just the same number of experienced and inexperienced users with similar behavior. We can deduce from this cluster that such users exist that never take a risk of making changes to their computers. They also prefer IBA over DBA as can be

seen in Figure 14(d).

Finally, VCiC is also not a dominant cluster because it contains comparable number of experienced and inexperienced users. We can see from Figure 14(e) that still a larger number of users prefer IBA over DBA in all scenarios. Thus we have seen that only the users from VCC and VCN have a slightly larger preference of DBA over IBA in scenario 1 otherwise, IBA is always preferred over DBA in all other scenarios including scenario 1 by rest of the users. Even the users from VCC and VCN prefer IBA over DBA for the remaining scenarios.

We have seen that in each cluster, people have always preferred IBA for Scenarios 2 and 3 since these scenarios are comparatively tougher for users to decide. The experienced users seem to prefer IBA when they encounter complex question. Similarly, inexperienced users also prefer IBA. This is because IBA makes it easier for the users to simply select their intention instead of understanding complex information the moderator gives them. We have observed that our inferences from the study of clusters match the ground truth that the only time the users prefer DBA over IBA is for scenario 1. Through clustering, we have identified the exact categories of experienced users who do that.

5.4 Results: Association Rules

We will now discuss some findings that we obtained after association rule mining. Using level of experience of the user as class attribute, the rules that we obtained showed that when presented with a hard question using DBA, the inexperienced users chose not to proceed but were not very confident about their decision. One such rule that we obtained was `S3Q1=Close web page S3Q2=Fairly Confident` $\rightarrow$ `Experienced= No`. When using Question 5 of scenario 1 as the class label, *i.e.*, the question asking about whether user liked DBA or IBA, we observed that the users who are not confident prefer IBA over DBA. This was precisely the objective of developing IBA *i.e.*, to make it easier for those users that are not very experienced to take informed decisions. One such rule that we obtained is `S1Q2=Not Confident S2Q2=Not Confident`

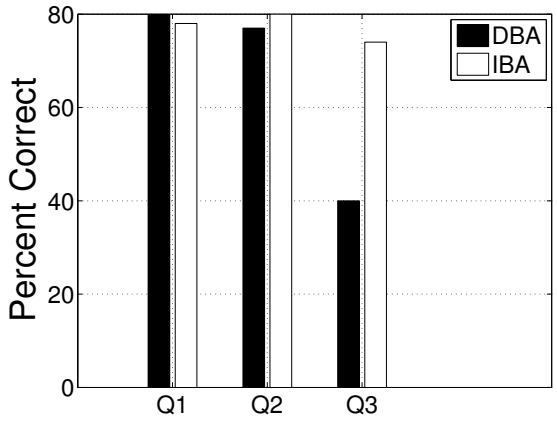


Figure 16: Number of participants who allowed access to basic Facebook information

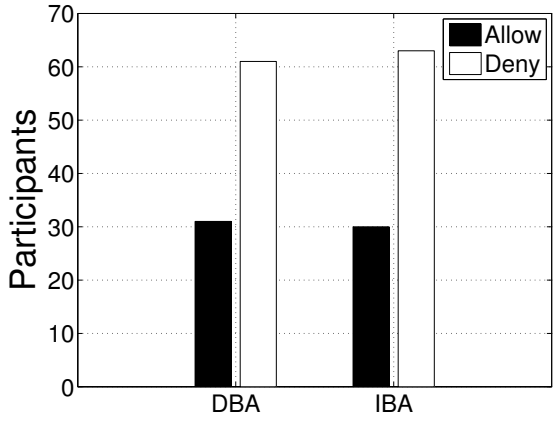


Figure 17: Percentage of participants making the correct answer for the first three questions

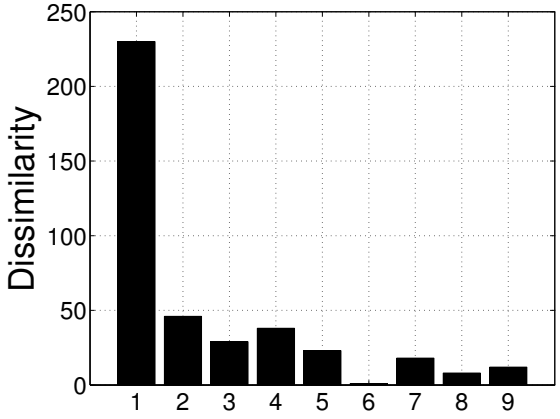


Figure 18: Difference between intra-cluster dissimilarity of consecutive clusters

→ `S1Q5= IBA`. We observed similar rules when we used Question 5 of scenario 2 as the class label *e.g.*, `S1Q2=Not Confident S2Q2=Not Confident` → `S2Q5= IBA`. Another rule that we observed stated that the users that are not confident and close the dialog box of DBA prefer IBA over DBA. The rule is: `S1Q2=Not Confident S3Q1=Close web page` → `S2Q5= IBA`.

When using Question 5 of scenario 3 as class label, we make the exact same observation again that the users that are not very confident about their decisions irrespective of whether they are experienced or inexperienced prefer IBA. We also make another similar observation that the users that are not confident and close the dialog box of DBA prefer IBA over DBA. Similarly, using Question 5 of scenario 4 as class label, we make the same observation that the users not confident of their choices like IBA.

From this discussion, we make two dominant conclusions: (1) Whenever the users are not confident about the choices they make, be them experienced or inexperienced, they prefer IBA over DBA, and (2) the users that are not confident at all about their choice in DBA and close the dialog box of DBA prefer IBA over DBA.

6 Conclusion

In this paper, we have proposed and extensively evaluated Intention Based Authorization dialog boxes, a more user friendly design in comparison existing decision based dialog boxes. Using the DBA dialog boxes, users are unable to make proper decisions because most of the users are not very knowledgeable and thus get confused by the technical information they see in DBA dialog boxes followed by the requirement to answer a Yes/No questions. We have shown that the intention based dialog boxes are not only more intuitive and user friendly, but also resolve the problems with DBA dialog boxes. IBA is preferred by the inexperienced users in all the situations as well as the experienced users in most of the cases especially in the scenarios that are not very common and the users are not already trained on them. IBA dialogs can effectively solve the problem of making uninformed and incorrect decisions that most users currently do due to the lack of user-friendliness of existing DBA dialog boxes.

APPENDIX

Survey Information

You are being asked to participate in a research study of computer security dialog boxes (messages boxes). You will be shown a series of proposed and existing dialog box designs and then answer several questions about each design. You must be at least 18 years old to participate in this research.

Participation in this research project is completely voluntary. You have the right to say no. You may change your mind at any time and withdraw. You may choose not to answer specific questions or to stop participating at any time.

You will not be compensated for this research.

If you have concerns or questions about this study, such as scientific issues, how to do any part of it, or to report an injury, please contact the researcher (Dr. Alex Liu, alexliu@cse.msu.edu, (517) 353-5152, 2132 Engineering Building, MSU Campus).

If you have questions or concerns about your role and rights as a research participant, would like to obtain information or offer input, or would like to register a complaint about this study, you may contact, anonymously if you wish, the Michigan State Universitys Human Research Protection Program at (517) 355-2180, Fax (517) 432-4503, or email irb@msu.edu or regular mail at 207 Olds Hall, MSU, East Lansing, MI 48824

You indicate your voluntary agreement to participate by completing this survey.

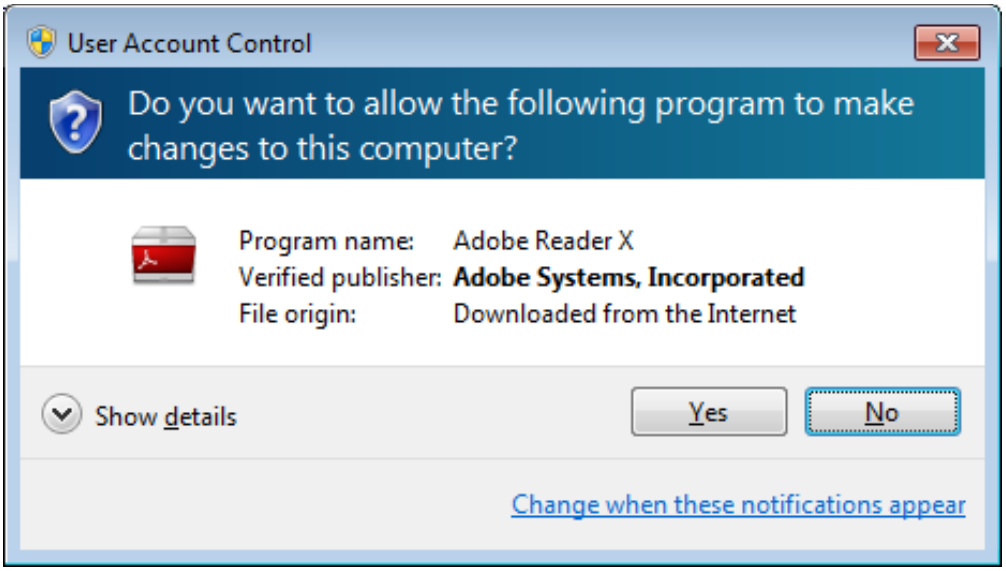
Participant Information

Which of the following best describes you:

- In the past, I have installed an operating system on a computer. I am comfortable configuring computers.
- I have never installed an operating system on a computer or I do not know what this means. I do not have large amounts of experience configuring computer systems.

Question 1

Suppose you are attempting to install Adobe Reader on your PC and the following dialog is displayed:



Upon viewing this dialog, which of the following actions would you take?

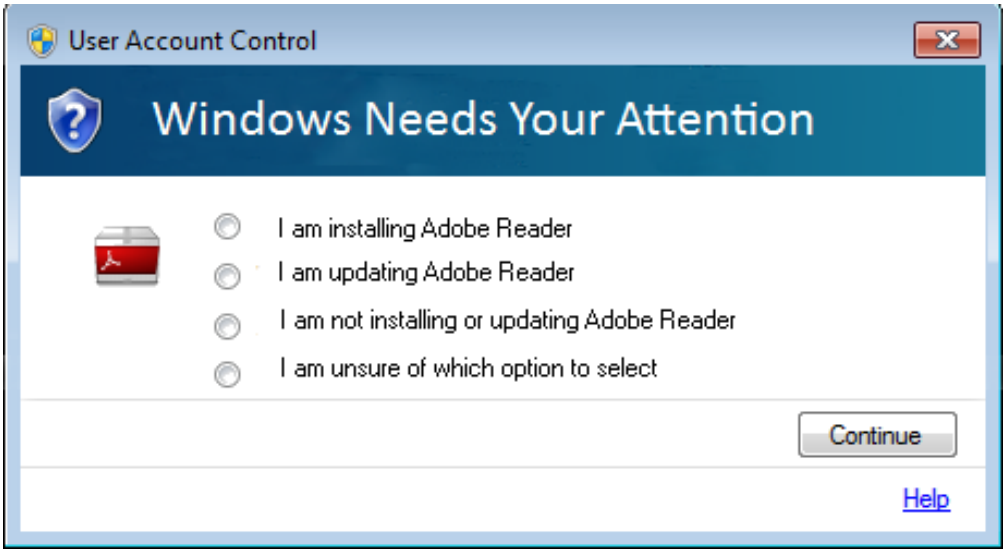
- Click the Yes button
- Click the No button
- Close the dialog box

How confident are you with your choice?

- Highly confident
- Fairly confident
- Not confident

Question 2

Suppose you are attempting to install Adobe Reader on your PC and the following dialog is displayed:



Upon viewing this dialog, which of the following actions would you take?

- Click option 1
- Click option 2
- Click option 3
- Click option 4
- Close the dialog box

How confident are you with your choice?

- Highly confident
- Fairly confident

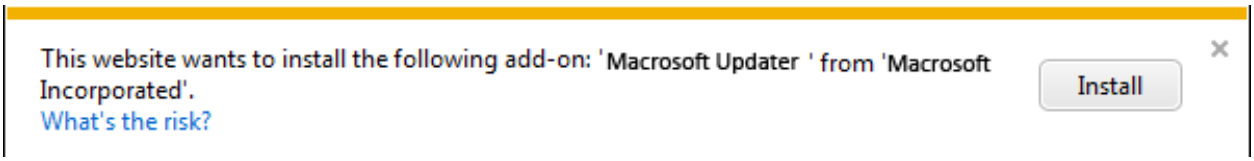
- Not confident

Comparing the options in Question 1 and Question 2, which set of options do you prefer?

- Question 1
- Question 2

Question 3

Suppose you are attempting to play an online game on an Internet website. Your receive the following dialog:



Upon viewing this dialog, which of the following actions would you take?

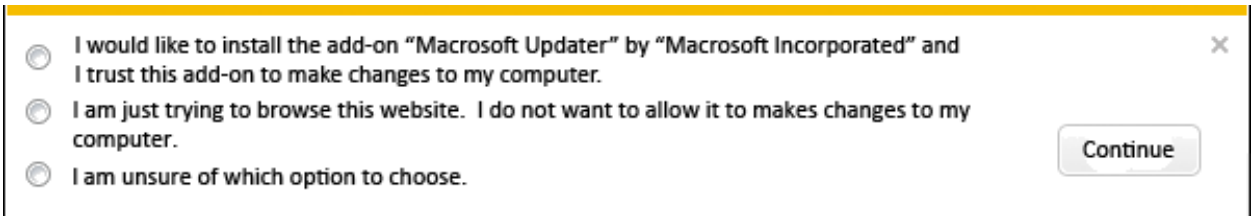
- Click "Install"
- Click the "X" in the upper right hand corner of the dialog.
- Close webpage

How confident are you with your choice?

- Highly confident
- Fairly confident
- Not confident

Question 4

Suppose you are attempting to play an online game on an Internet website. Your receive the following dialog:



Upon viewing this dialog, which of the following actions would you take?

- Click option 1
- Click option 2
- Click option 3
- Click the X in the upper right hand corner of the dialog.
- Close the webpage.

How confident are you with your choice?

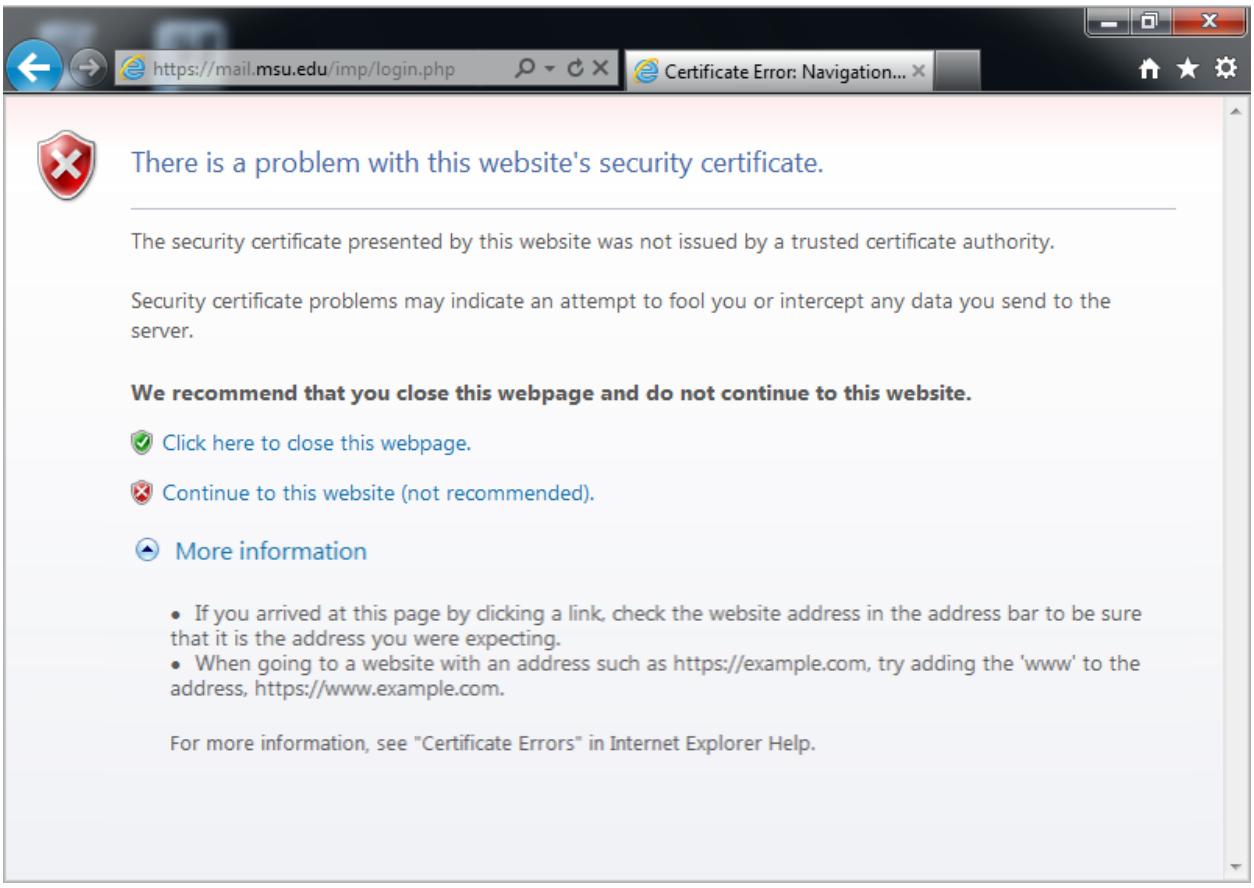
- Highly confident
- Fairly confident
- Not confident

Comparing the options in Question 3 and Question 4, which set of options do you prefer?

- Question 3
- Question 4

Question 5

Suppose you are attempting to access your MSU E-mail via the Internet. You navigate to http://mail.msu.edu and receive the following warning:



Upon viewing this dialog, which of the following actions would you take?

- Click Click here to close this webpage.
- Click Continue to this website (not recommended).
- Close the webpage

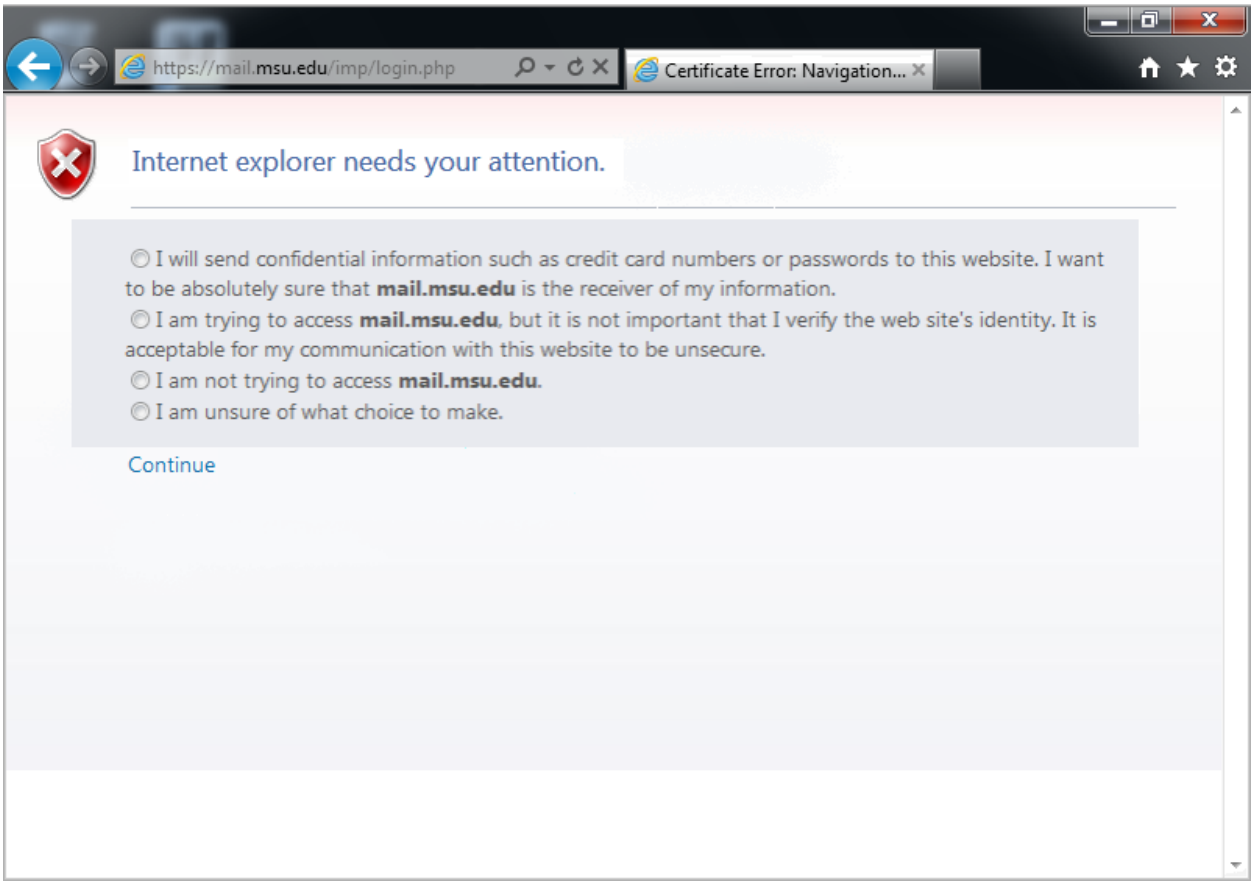
How confident are you with your choice?

- Highly confident

- Fairly confident
- Not confident

Question 6

Suppose you are attempting to access your MSU E-mail via the Internet. You navigate to http://mail.msu.edu and receive the following warning:



Upon viewing this dialog, which of the following actions would you take?

- Click option 1
- Click option 2

- Click option 3
- Click option 4
- Close the webpage.

How confident are you with your choice?

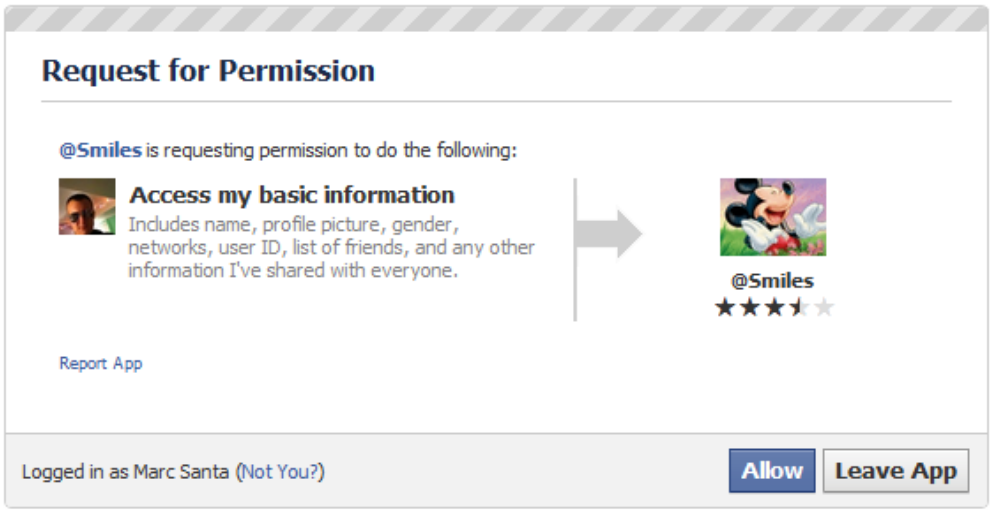
- Highly confident
- Fairly confident
- Not confident

Comparing the options in Question 5 and Question 6, which set of options do you prefer?

- Question 5
- Question 6

Question 7

Suppose you are attempting to install a Facebook application named **@Smiles**. This application claims to brighten up your friends day by sending them cute pictures. Your receive the following dialog:



Upon viewing this dialog, which of the following actions would you take?

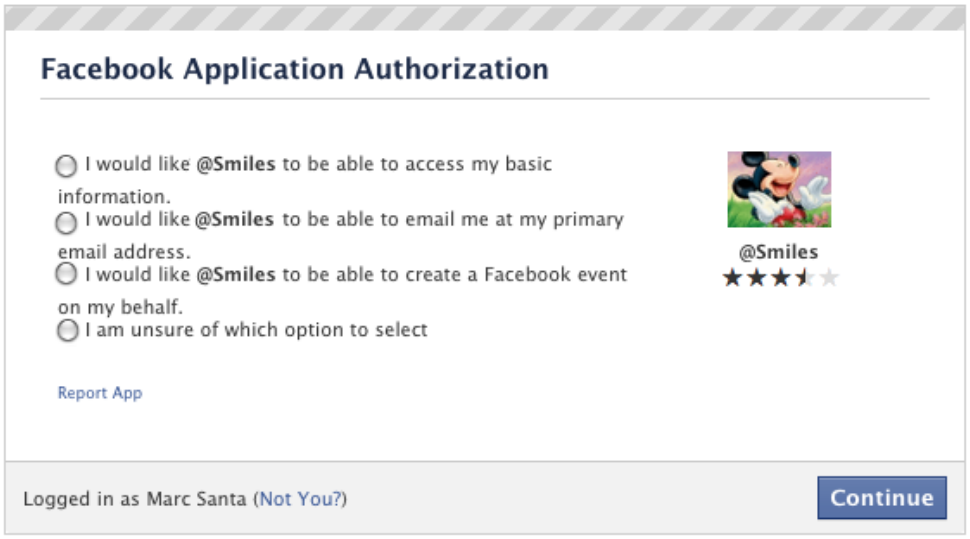
- Click the Allow button
- Click the Leave App button
- Close the webpage.

How confident are you with your choice?

- Highly confident
- Fairly confident
- Not confident

Question 8

Suppose you are attempting to install a Facebook application named **@Smiles**. This application claims to brighten up your friends day by sending them cute pictures. Your receive the following dialog:



Upon viewing this dialog, which of the following actions would you take?

- Click option 1
- Click option 2
- Click option 3
- Click option 4
- Close the webpage.

How confident are you with your choice?

- Highly confident

- Fairly confident
- Not confident

Comparing the options in Question 7 and Question 8, which set of options do you prefer?

- Question 7
- Question 8

References

References

[1] Improving security decisions with polymorphic and audited dialogs. In *Symposium On Usable Privacy and Security* (2007).

[2] Apps on facebook.com, July 2011. http://developers.facebook.com/docs/reference/api/permissions/.

[3] Agrawal, R., and Srikant, R. Fast algorithms for mining association rules. In *20th International Conference on Very Large Data Bases* (1994), 487–499.

[4] Cao, X., and Iverson, L. Intentional access management: Making access control usable for end-users. In *Symposium On Usable Privacy and Security (SOUPS)* (2006).

[5] Witten, I. H., Frank, E., Trigg, L., Hall, M., Holmes, G., and Cunningham, S. J. *Weka: Practical Machine Learning Tools and Techniques with Java Implementations*. Citeseer, 1999.

[6] Wu, M., Miller, R. C., and Little, G. Web wallet: Preventing phishing attacks by revealing user intentions. In *Symposium On Usable Privacy and Security* (2006).

[7] Zurko, M. E., Kaufman, C., Spanbauer, K., and Bassett, C. Did you ever have to make up your mind? what Notes users do when facedwith a security decision. In *18th Annual Computer Security Applications Conference* (2002), 371 – 381.