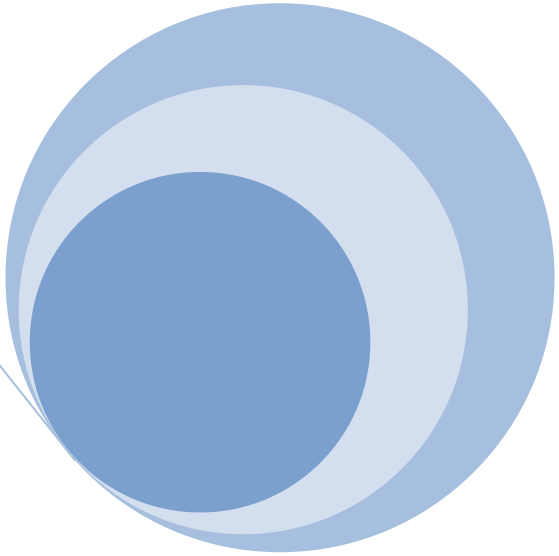
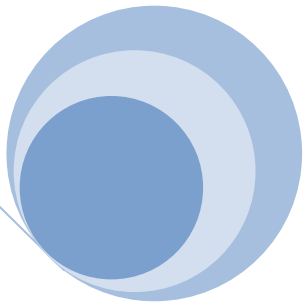


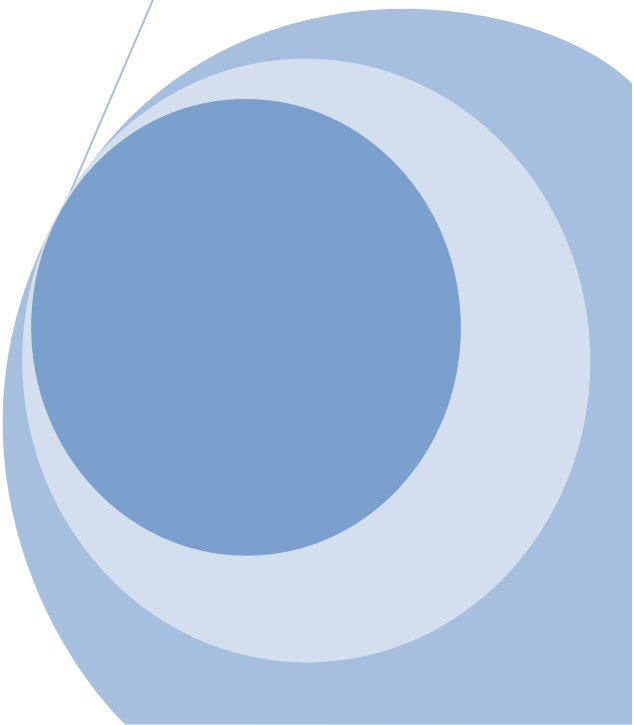


Desensitizing the User

A Study of the Efficacy of Warning Messages

A dissertation submitted in partial fulfillment of the requirements for the degree of Master of Science in Software and Systems Security.

2011



Michele Daryanani
Kellogg College, University of Oxford

Abstract

Responsibility for security is being pushed more and more onto the end user, with the effect that a user sees hundreds of messages warning him/her about the risks involved. Such an incessant bombarding has the potential to desensitize the user to the underlying security risks, thus negating the initial intent of the warning. A study was performed to verify the desensitizing effect on users that were repeatedly bombarded by such messages. This was attained by presenting two sets of users (one that had been “desensitized” and one that had not) with a window that replicated such a message, in a way very similar to a phishing attack. Based on how the users reacted to the window, the effectiveness of prompting a user for elevation was construed; and the use of such methods was evaluated.

The author confirms that:

This dissertation does not contain material previously submitted for another degree of academic award, and the work presented here is the author's own except where otherwise stated.

To my mother, for her guidance, support and incessant pestering.

Special thanks must go to my tutor, Ivan Flechais for his relentless patience and advice; as well as my best friend, Elisa Novelli, for spending long nights reading this dissertation.

Table of Contents

1. Introduction

The motivation behind this project started during Dr. A Sasse's People and Security (PAS) lectures, whereby she felt that the current state of the "security interface" is quite counterproductive as it was too complex – and where it wasn't complex it was "nagging". After a fairly heated debate with regards to user training; the author hypothesised that users could be desensitised by security pop-up windows, and that this could have an adverse effect on the security stance of the system. Hence, it was decided to tie usability with security and explore the link between the two. It was felt that a fairly secure system can fall apart because the users ignore repeated warning messages. Given that the author was working as a system administrator in a company with a large variety of staff (age, ethnicity, sex, education, IT experience, etc) he could get quite a variety of people willing to fill out questionnaires/do experiments. More importantly, the staff was divided quite clearly between two distinct groups; those who use Windows and those who use Os X. He strongly believed that users assign very little value to some data/systems, which at times can have a very high intrinsic value. Examples of this were local passwords to a computer. Most users wouldn't think twice before giving their password to a colleague when they go on holiday; yet these passwords could "unlock" vast amounts of data and privileges.

The author felt that this situation was worsened by Operating System developers' recent attempts to "increase security" by increasing the responsibility of the user for security. A perfect example of this was Apple OsX's password pop-up screen, which appeared whenever the operating system needed to run a program in administrative/elevated mode; or to unlock their keychain. Ironically, this was supposed to reduce the number of passwords a user needed to remember/type in. The author believed that as the window that pops up did not give any information as to what program/service/process spawned it, alongside with the incredible amount of times users saw this window, it desensitised users to the point that they were willing to type their username/password into the window (or a window that looked just like it – a common phishing-type attack) regardless of their activity at the time. I.e. while a user that was not used to seeing that window might have thought twice ("why am I seeing this? Am I installing anything at this moment in time that actually requires elevation?") - a user that was used to seeing that same window multiple times per day may type their password in "just to get it over with". This made the users of such a system more vulnerable to phishing attacks. As such, the author thought of creating a simple online questionnaire, with a phishing attack embedded. The questions themselves served only one purpose, to stratify the respondents (IT-native/expert, child, novice-user, elderly parent, Apple-native, non-apple user, etc). The actual data came from a popup window halfway through the questionnaire (preferably after the classification questions are finished, so that if the user decides to close the window at that point, all the needed data had already been collected). The pop-up window, similar to that which Apple presents when elevation was requested, would ask the user for a username/password. Obviously, for

ethical/legal reasons the experiment would not save the password – the first character would be verified with the user, and then discarded altogether. The actual data was whether the user gave their password (or not) – not the actual password. The author's theory was that the users were so used to seeing the password request window in OsX that they would type their password into any website's pop-up without thinking twice. As telling the user beforehand the premise of the experiment would influence, if not negate, the data collected, the true intent was not be disclosed at first. After the experiment, the true premise/intent was disclosed. As this could have been construed as deception of the respondent/user, formal approval from CUREC had to be secured. The results were then generalised to cover other systems, where repeatedly requesting the user to confirm an action (as a security measure) can be detrimental to the overall security of the system. Going one level further, this could be adapted for windows users; but the author felt that it may be better to have 2-user groups and limit the scope of the experiment to gain more defined results. The two groups were defined as:

- Those who predominantly use Apple, and have seen the popup over the previous twelve months
- Those who predominantly use Windows, Linux or other operating systems, and have not seen the Apple specific popup

This allowed having the non-OsX native users as our control group, and the desensitized OsX users as the experimental group. Clearly, having a large enough user-base to get any data is important. The author had close to 400 Apple users available. With only half of the user-base participating, it would have allowed sufficient data to be extrapolated. Further tests could were be made using "external world" users, where their Apple background was questioned in the survey.

Before the experiment could commence, phishing attacks in general were investigated, and thus the second chapter discusses phishing attacks. In turn, this lead to a brief investigation on user desensitization, and human-computer interaction (HCI), which are also discussed in chapter two. These investigations influenced the choice of questions on the survey, as well as the analysis of the results.

As the experiment was devised, several choices had to be made. The though process behind these choices, which in turn influenced the questions in the cover questionnaire discussed in chapter three.

The data collected would appear to support the author's hypothesis at first sight. This was further supported using Chi Square tests, Mann Whitney U Tests and other statistical analysis methods. As such, subsequent chapters use the data from the questionnaire and analysis thereof to support the author's hypothesis. Several interesting cases from the experiment are included, as these provide exceptions to the norm..

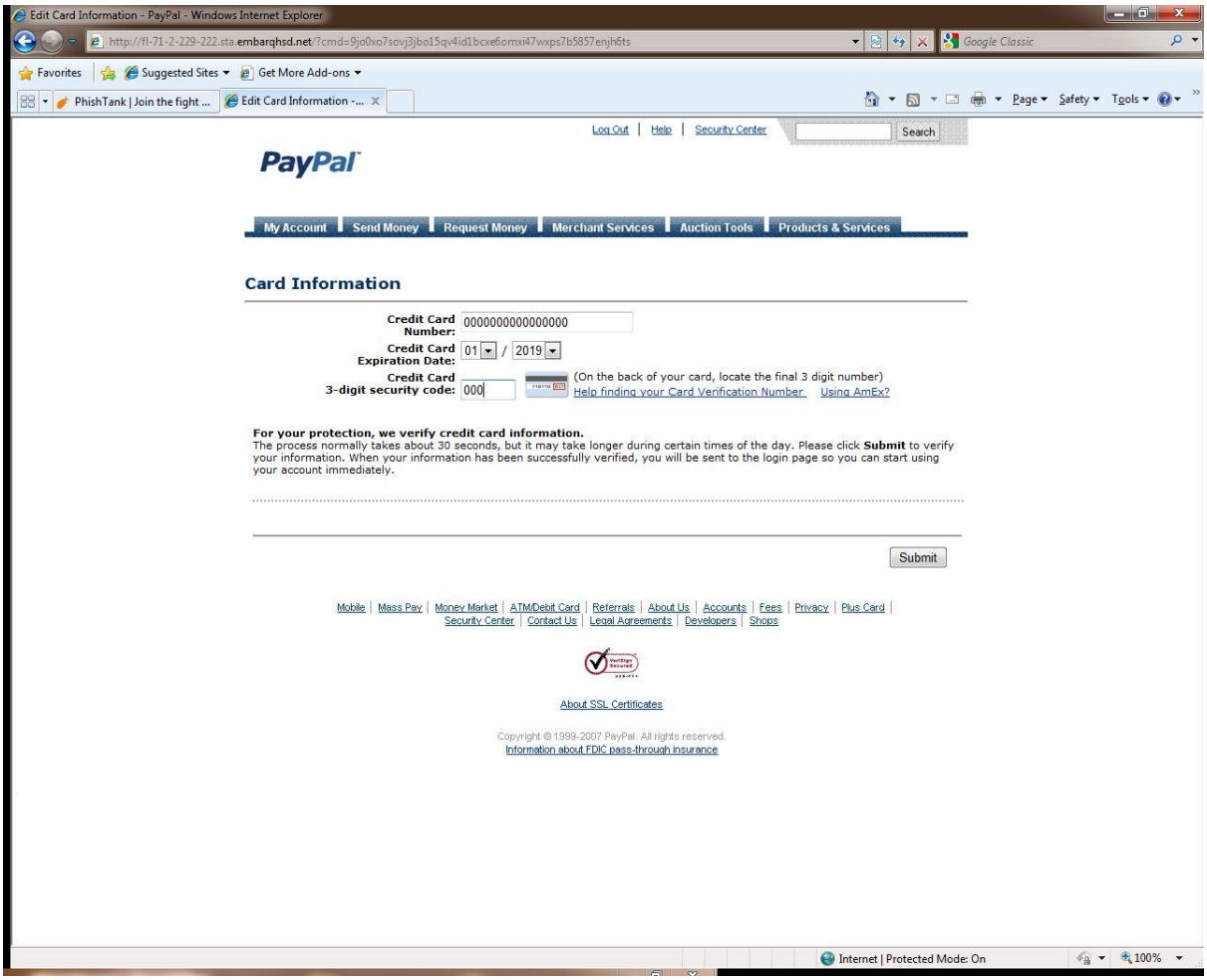
The examples highlight the issues around desensitizing users to other forms of “security” (like the broken padlock in internet-explorer, expired SSL certificates for “secure” websites or the incessant popup of an antivirus programs).

Passwords are not the ultimate answer to security; no-one has come up with the silver-bullet to end all security problems. That said, there are other authentication methods which may be less susceptible to desensitisation and phishing attacks. As such, it must be noted that the aim of the project was not to conclude that “Apple is unsafe” - it merely used Apple as evidence to support a theory and the authors hypothesis. The aim of the project was to discuss and provide scientific backing to the theory that over-exposing a user to security warnings may prove counterproductive. This process eventually desensitizes the users to the point that it negates the usefulness of the security measure put in place.

2. Phishing Attacks

2.1 Phishing Explained

Figure 2-1: Paypal Phishing Page



Phishing, pronounced “fishing”, is a spoofing[1] attack designed to exploit a pre-established trust between a user and a third party, usually a company, to gain access to confidential information. Typically, an attacker uses harvested email address lists, as used by spammers, to target hundreds, if not thousands or even millions of email addresses in one go. While phishing attacks are not limited to propagation via email, this method is extremely popular. Cisco reported of the 800 billion spam emails send every day, as many as 200million are “sophisticated phishing attacks” ('Spear Phishing' Increasing 2008).

In the typical example, a scammer (or more aptly, a phisher) creates an email purporting to be from a bank, company or other well established organisation. Under some pretence, the email asks the users to click a link to verify their personal information. Excuses range from system upgrades, security checks, new messages on the system or even hacker attacks. In

[1] Spoofing is a technique of adapting something, normally a program, to appear to be working and functioning normally, when in reality it has been modified to serve some further purpose.

some rare cases, the scammer will ask the user to open an attachment – this may be an attempt to evade the increasingly complex anti-phishing tools or an attempt at merging a phishing attack with a Trojan infiltration[2]. Once the user clicks on the link, he or she is directed to a website designed to look like the one of the real company; usually, the attacker will use some form of obfuscation on the domain name. Secondly, it's common practice by attackers to host these sites on server's they do not own (but have a backdoor to) or on free hosts that don't require identity verification – this way when the site is discovered by authorities, the real attacker is harder to find. On the fake side, the user is asked to either log in – or for one reason or another release their credentials and details. These are skimmed and stored by the attacker.

It must be noted that while the majority of phishing attacks rely on an email asking a user to click a link which leads to a fake website; there are variations. One such variation is a vanilla-phishing attack, whereby a user is asked to reply to an email giving away details. Another, known as "spear phishing" relies on an attacker targeting (or spearing) a subset of users rather than a large untargeted attack. Commonly, an attacker will collect publicly available data about users within a large organisation or government (e.g. roles, contact details, full names, hierarchy, etc), and then email a set of users pretending to be a senior member of the company asking for further details. Typically, these emails carry some form of urgency attached. Due to the highly targeted nature of these emails, collecting statistics is much harder and many organisations choose not to make the breach public to protect their reputation.

Phishing is real problem nowadays, and a working-group has been established to try and combat it. The Anti-Phishing Working Group has been monitoring phishing attacks since 2003. Worryingly, in December 2008 they recorded 31,173 phishing sites (APWG 2008) – compared to 113 in December 2003 (Consumer.Gov n.d.), an increase of 25,000% . Consumer Reports, estimated that in the U.S.A. alone, USD$2.1 Billion were lost to phishing in 2007 (Consumer Reports 2007). From a more technical side; Sophos, a leading IT security firm, estimated that "Phishers are able to convince up to five percent of recipients to respond" (Sophos 2009).

2.2 Analysing a Phishing Attack

Perhaps the clearest way to explain a phishing attack is by falling for one, and analyse what is happening. The attack actually starts long before the user receives a phishing message; the attack actually starts when the phishing website is created.

A common technique used by scammers is to save the real page of a reputable company, and then edit the code behind the form submissions to submit the data to a compromised

[2] Trojan Infiltration is the process whereby a user installs a program thinking that it is beneficial, when in reality the code is a harmful 'trojan'. The name comes from the renown Tojan horse that ended the war of Troy by allowing the invading Greek army to infiltrate the city.

server. The advantage of this is that the generated phishing site can then be injected into web servers owned by unsuspecting third parties automatically by tools that trawl the internet. This serves multiple purposes;

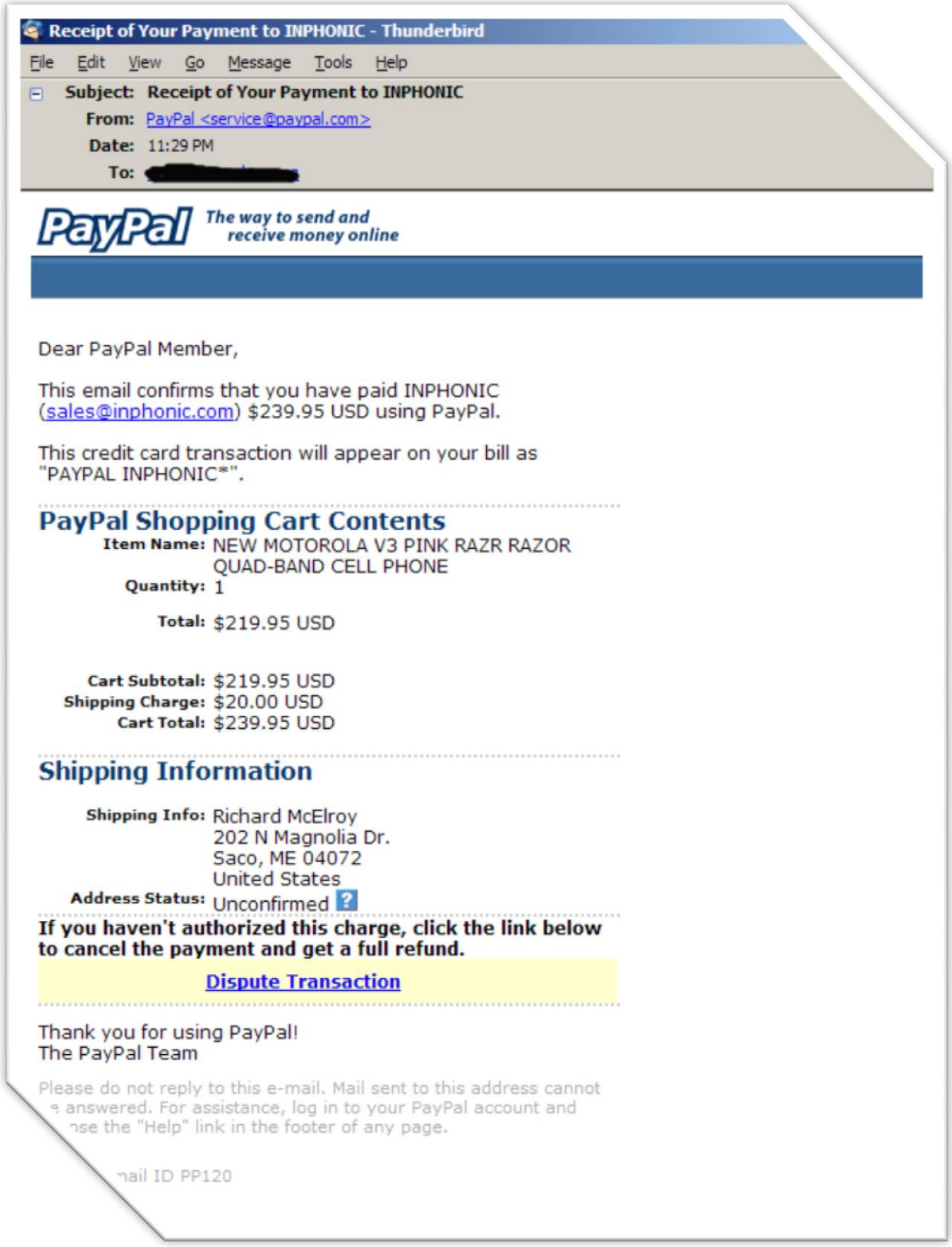
- It hides the identity of the scammer behind an unsuspecting third party
- It scales very quickly, very easily
- It is unattended
- It is very difficult to counteract

To automate the attack, the attacker creates the fraudulent website's code once and then lets a script do the rest. The script then scans websites for vulnerabilities, unpatched and out-of-date scripts (popular cases are php bulletin boards), then launches an attack on the vulnerability with the intention of uploading the fraudulent website into a folder somewhere on the server. The server is now compromised and hosting a phishing website; and the legitimate owner of the website may not even realise it. It is extremely rare that the legitimate website itself is taken down or visibly affected by the attack; this is done to prolong the length of time before the fraudulent content is discovered.

The second phase of the attack lies in getting users to actually visit the fake website. This is commonly done by replicating spam-email techniques, but adapting the content. Instead of advertising a product, phishing emails commonly trick users into visiting the fake website.

Taking one of the plethora of phishing emails that arrive on a daily basis to millions of users worldwide:

Figure 2-2: PayPal Phishing Email

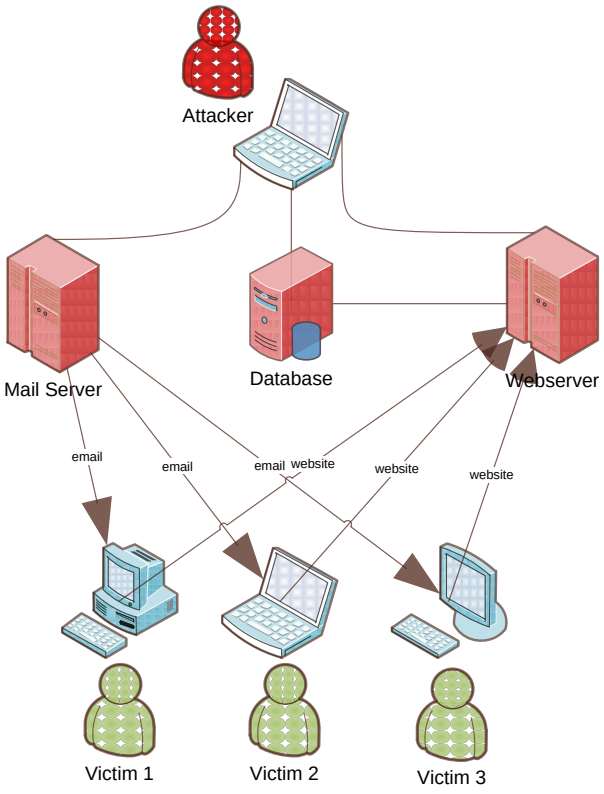


The phishing email above is particularly creative as while many people have gotten used to the classical “click here to verify you details” type emails, the average user will be too

preoccupied trying to figure out why Paypal thinks they sold a phone to think that it may be a scam.

Looking at the headers may help, but in all likelihood this would be a futile task. Given the nature of email transfer systems, the attacker will either use email relay systems without authentication, or compromised "zombie" machines to send the emails. Thus, in a manner similar to the more common spam email, tracing headers would simply lead to a compromised machine or a weakly configured server. It is incredibly rare that a phisher uses their real email account to send emails. By using these methods, the phisher not only conceals their identity making it very difficult to find and stop the source of the emails, but also allows the phisher to send larger quantities of emails. Instead of a script running on a machine in the phisher's basement, the phisher can use a botnet (or indeed the mail server of the compromised websites) to send out a large multiple of the emails he would be able to send from one machine.

Figure 2-3: Phishing Topography



The diagram above shows a typical situation whereby an attacker compromises 3 servers, a mail server to send out the phishing emails, a webserver to host the fraudulent website and a database server to store the captured data. While all three of these can reside on the same box, it is quite common that they are hosted on three different webhosts. This is because the mail server is quite "noisy" when it sends spam, and that will be the first to be shut down. Second, as more users/victims report the phishing site, that will be taken down, and the server patched. If the attacker did not get the data off by this point, his efforts

would be in vain. This is where a third server comes into play, usually in a country with more relaxed IT laws and enforcement. As the phishing site collects data, it automatically passes this to a third server for storage. Even though the webserver is patched and the attacked locked out, the attacker will most likely still have access to the third server with the database, and can gather the information before the administrator of the third system is contacted and that too is locked.

It must be noted that using hosts in different countries makes life much harder for law enforcement trying to take down these systems. While legal regulations surrounding taking down a compromised website owned by an unsuspecting third party may be complicated, doing so across multiple countries, each with their respective legal systems and enforcement methods is even more so. As such, many takedowns happen by contacting the webmaster or webhosts, who willingly correct the issue rather than going through the legal channel. Similarly, there are companies that focus specifically on these services (e.g. MarkMonitor's phishing takedown service).

Going back to the example phishing email above, clicking on the link presented to "dispute (the) transaction" does not go to paypal.com, but to a totally different domain not even remotely owned by PayPal. The front page presented is a very convincing replica of PayPal's, and a very well made phishing attack.

Figure 2-4: Paypal Phishing Frontpage

Even the best made spoof has tell-tale signs. Traditionally, spelling mistakes, grammar mistakes and sentences which didn't quite make sense were the most obvious; but scammers have been improving on all these counts (or simply copying the official communications and sites better). From a more technical perspective, the URL bar is a clear giveaway. The real PayPal uses HTTPS with a signed certificate, which would turn the url bar green. Similarly, the URL is not PayPal.com, not an obfuscated version which would appear to be paypal.com.

When obfuscating a URL, the attacker makes use of the way URLs map to an IP address, or of the users' lack of technical knowledge. For example, the domain www.hsbc.co.uk is owned by the HSBC bank – but a phisher can register www.hscb.co.uk a domain that looks very similar. Alternatively, an attacker could host the phishing site on a subdomain, e.g. http://hsbc.mydomain.com, or even within a folder http://mydomain.com/hsbc/. These are fairly simple methods that require very little technical knowledge by the attacker; going one level up technically, an attacker can user an IP address http://209.85.227.103/[3] or even an IP address in octal, for example http://3512067432 thus cutting out the DNS server

[3] 209.85.227.103 maps to google.com

altogether. Other ways of technically obfuscating a URL include exploiting the way browsers connect to servers with usernames. Hence, http://www.hsbc.co.uk@mydomain.com/ sends the users to mydomain.com, and attempts to sign in as the user www.hsbc.co.uk. Merging two or more of these methods is quite a good way to confuse a less-technologically savvy users, e.g. http://www.hsbc.co.uk@351067432/login.php will just send the user to the page login.php on the server at IP address 351067432 (a.k.a. google.com).

By looking at the source code, particularly the form submission code, one can determine where the data will be submitted to. In this case, the form submits to:

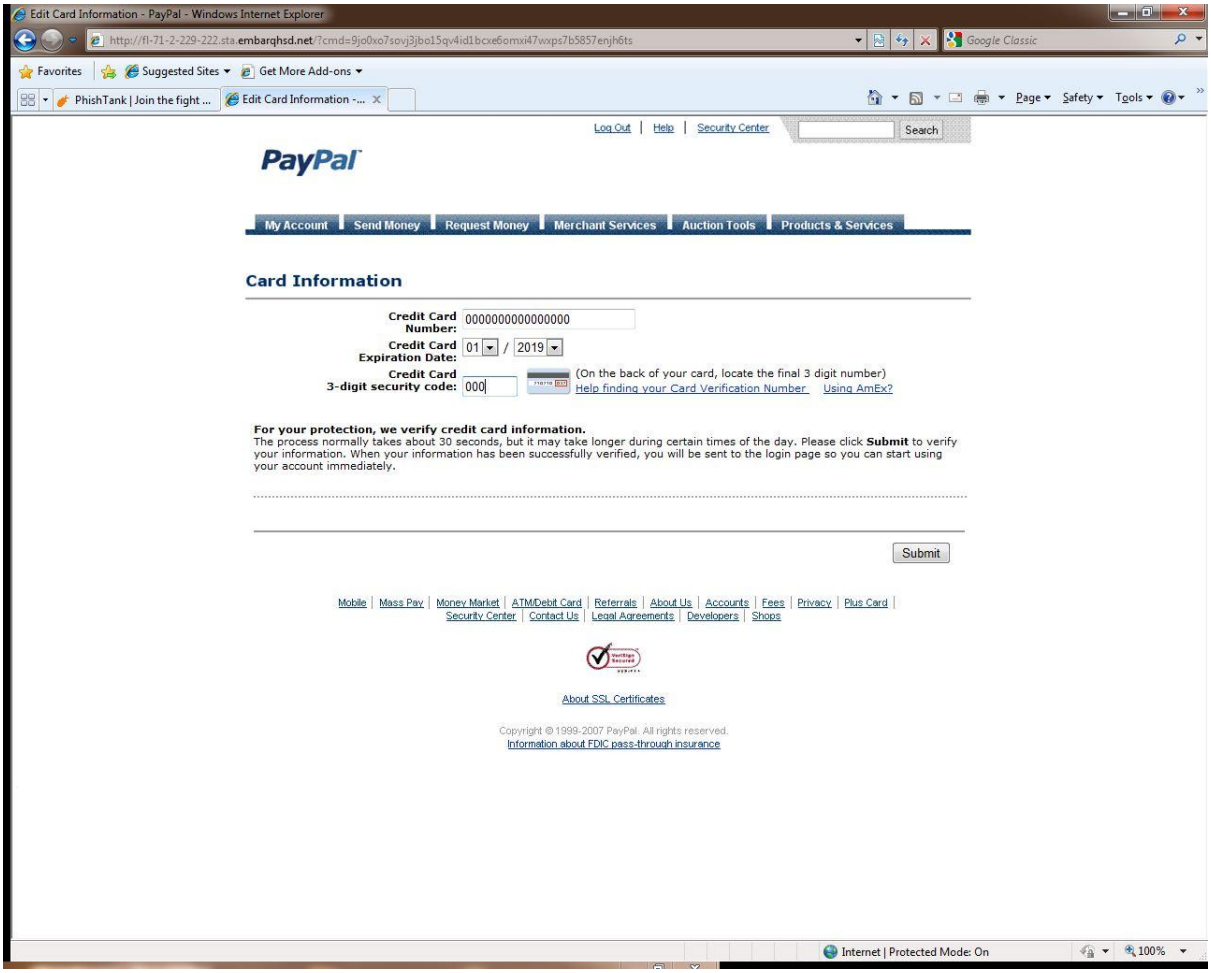
<form method="post" name="login_form" action="http://www.walid-info-uk.com/paypal.com/cgi-bin/us/webscrcmd=_account-run/www.paypal.com/updates-paypal/confirm-paypal/login.php">

The website it submits to is not obviously related to the one running the actual phishing attack. The second website, in all likelihood, was another one compromised by a php script. Not only is the script receiving the files written in php, but so it the real, potentially legitimate website.

It must be noted that just because the script that receives the form's submission is on a second site the data is stored there. It is more than possible that the php script then connects to a MySQL database on another webserver altogether.

Interestingly, once the user submits login data, the user is the redirected back to the original server and asked to submit further data "for security" validation.

Figure 2-5: Paypal Phishing Page



Not content with the username and password for the paypal account, the phishing site proceeds to request the complete credit-card number, expiry data and CVV2 code. While most credit-card payment systems require a full name and postcode to pay; the full name isn't verified automatically. Additionally, the postcode could be obtained from the victim's paypal account which is by now compromised. This cannot be verified without seeing the actual server-side script.

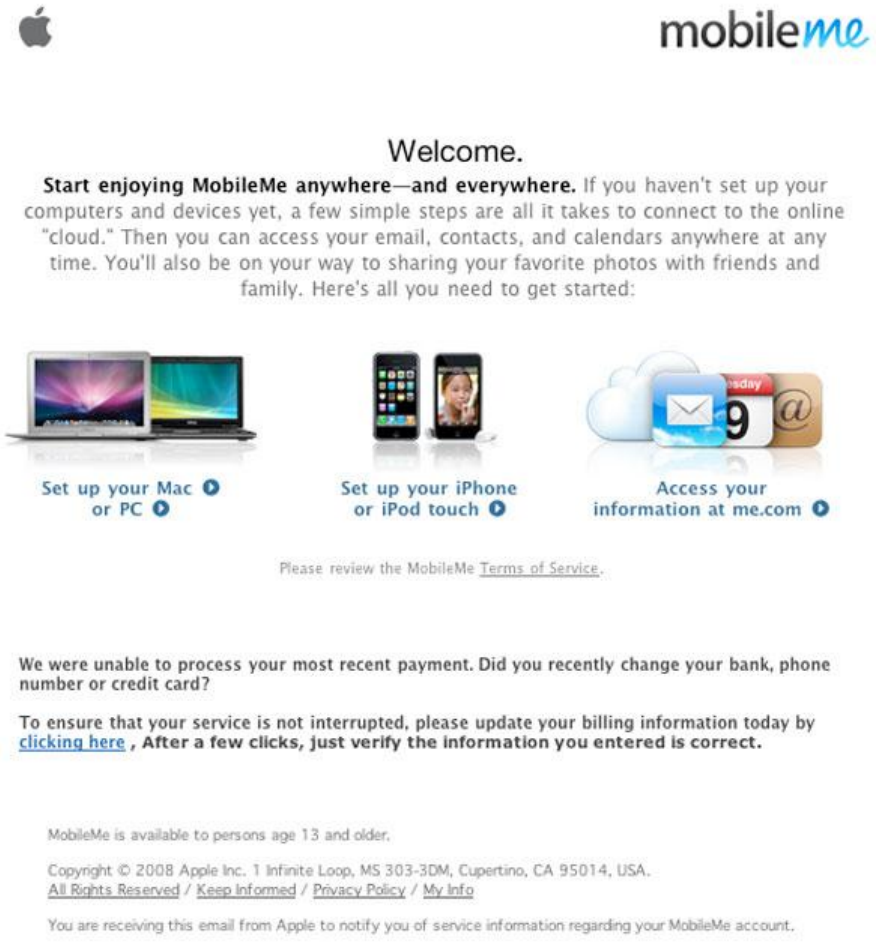
Only once the victim has disclosed their PayPal username, password, credit can number, expiry date and CVV2 are they redirected to PayPal; where they can sign in. By this point, in all likelihood, the user's PayPal credentials were used to automatically make a payment for goods, while credit card data tends to be sold on the black market.

2.3 Phishing Targeting Apple Users

Recently, phishers have realised that realised that Apple users are not as targeted by online malware and scams as windows users. As such, more and more phishing attacks have started targeting Apple users specifically. An example of these is in the email below, which specifically targets MobileMe users. MobileMe is a service available to most Apple users,

including iPhone and iPad users – and as such targets a very large portion of the online population.

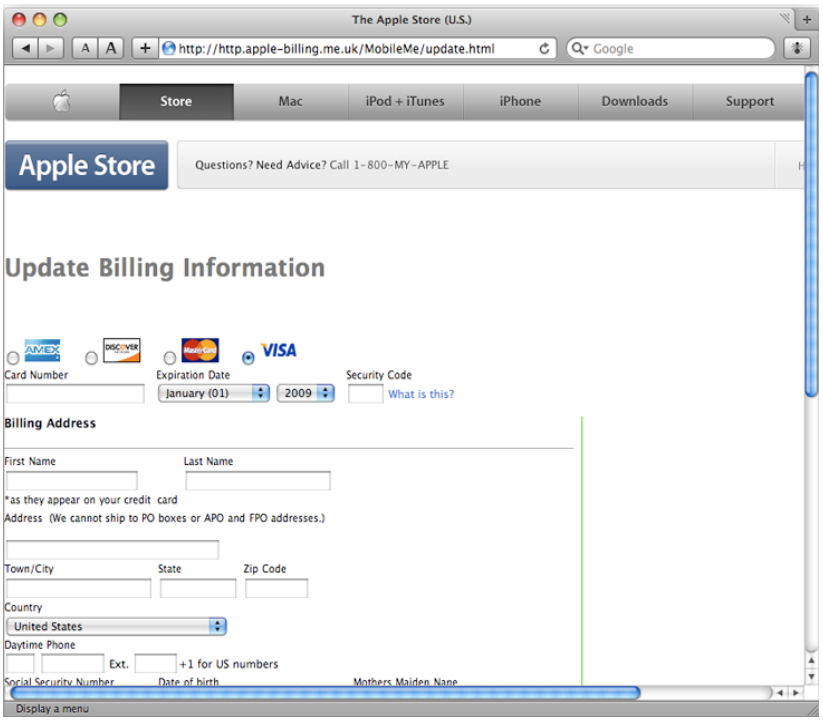
Figure 2-6: Apple MobileMe Phishing Email



(Cheng 2009)

Again, clicking on the link brings the victim to a phishing website designed to look like a legitimate Apple website.

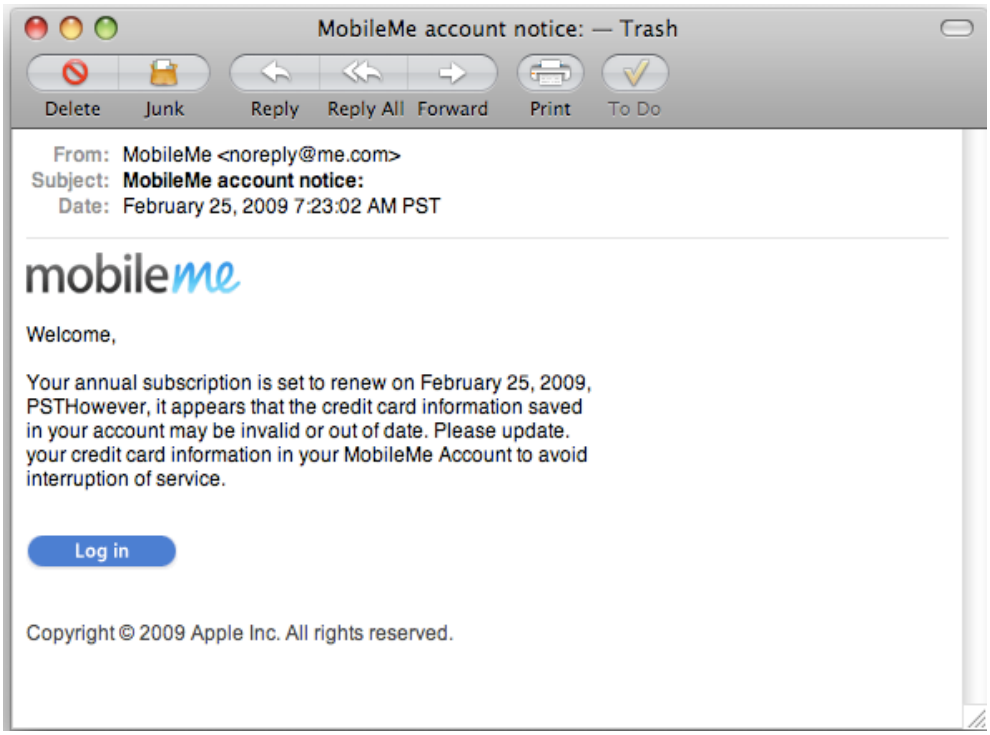
Figure 2-7: Apple Store Phishing page



(McLean 2009)

Similar examples can be found by simply searching online for “Apple Phishing”.

Figure 2-8: Apple MobileMe Phishing Email



(McLean 2009)

When Apple transitioned from Mac.com to MobileMe.com/Me.com, phishers saw an opportunity. Apple users were particularly targeted, with one user stating that “the confusion caused by the MobileMe transition caused her to lower her guard.” (Goodin

2008). An article on TheRegister.com states that "Evidently, Apple users are a popular target." (Goodin 2008)

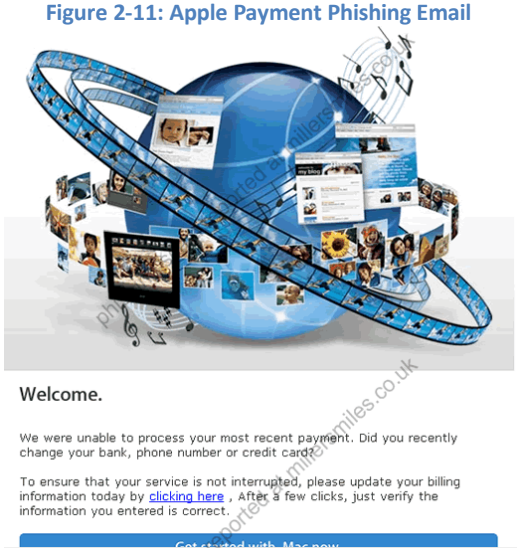
Figure 2-9: Apple Store Phishing Email

(FireHaus Network 2010)

Figure 2-10: Apple account Phishing Email

(Miles 2010)

Figure 2-11: Apple Payment Phishing Email



(Miles 2010)

What may not be immediately obvious are the potential consequences to a phishing attack's victim. In an attack where the victim releases their credit card data, the victim can assigns a

tangible value to that data; nominally the value of the money on their card. Similarly, with the disclosure of a victim's online banking credentials, the victim can immediately identify the value of the money they have in their account, which may be stolen.

2.4 The Value of Phished Data

On the other hand, in an attack where a victim gives out their email account's credentials, or their MobileMe account's credentials, it isn't immediately obvious what the tangible consequences would be. This becomes particularly obvious when less technically minded users, such as Dr. Dal Conte (Dal Conte 2010), say "I don't care if someone logs into my Facebook page, I have nothing to hide". As the intrinsic value of data is not immediately tangible, it becomes harder to visualise material losses associated with data loss. In the example with Dr. Dal Conte, compromise of her Facebook credentials would open her to identity theft, posting of spam, advertising, Trojan or other content to her contacts, further attacks on her contacts, etc. Similarly, from the data harvested (e.g. full name, birth-date, mother's name & address) an attacker could reset Dr. Dal Conte's online banking password; or indeed, open a new bank account in her name.

Interestingly, a number of trojans are starting to target Apple OSX users. A perfect example of this is shown in Figure 2-12: OSX Scareware, which is an example of a trojan that scares the user into thinking they already have malware installed on their machine. The window emulates an OSX Finder window, while in reality it is nothing more than a webpage within Safari. If the user falls for the trick, they are prompted to install a bit of software to clean up the machine – while in fact, they are infecting the machine.

Figure 2-12: OSX Scareware

The software that is installed will allow a remote party unlimited access to the system, including (but not limited to), key logging, viewing the screen, installing further malware, and more. In essence, any personal details typed or stored on the system are available to the attacker.

Symantec's annual Internet Security Threat Report paints a grim picture. In the top 10 "Goods and services advertised on underground economy servers" list (Table Table 2-1: Goods and services advertised on underground economy servers), Symantec reports stolen credit cards, bank account credentials, full identities, email accounts, and website administration credentials - all for sale. (Symantec 2010)

Table 2-1: Goods and services advertised on underground economy servers

Overall Rank			Percentage		
2009	**2008**	**Item**	**2009**	**2008**	**Range of Prices**
1	1	Credit card information	19%	32%	$0.85–$30
2	2	Bank account credentials	19%	19%	$15–$850
3	3	Email accounts	7%	5%	$1–$20
4	4	Email addresses	7%	5%	$1.70/MB–$15/MB
5	9	Shell scripts	6%	3%	$2–$5
6	6	Full identities	5%	4%	$0.70–$20
7	13	Credit card dumps	5%	2%	$4–$150
8	7	Mailers	4%	3%	$4–$10
9	8	Cash-out services	4%	3%	$0–$600 plus
10	12	Website administration credentials	4%	3%	$2–$30

(Symantec 2010)

Similarly, according to a report by Trusteer, an anti-phishing organisation, "between $2.4M-$9.4M" (Trusteer 2009) are lost per million online banking clients. On the same report, Trusteer found that "for every one million users, 4,700 login details are lost to criminals each year…. (which translated to) 0.47% of a bank's customers fall victim to Phishing attacks each year" (Trusteer 2009).

Trusteer is not the only organisation producing a tool to fight phishing. In fact, there are a plethora of automated tools on the market to help fight phishing attacks. Trusteer's Rapport is widely used in the financial sector, with Natwest, HSBC, RBS and Standard Bank being just

a small subset of the banks that have adopted Trusteer's services (Trusteer 2010). The reason behind financial institutions adoption of Trusteer is quite reactive. During the first quarter of 2010, 73% of all brands used in phishing attacks were from the financial sector. (Symantec 2010). Similarly, Trusteer estimates that every year, banks lose $9.4 for every customer to phishing attacks. As such, banks have had to react to this very serious threat.

That said, as soon as a tool is released, an existing attack is modified or a new one devised to cope with the tool. For example, by searching MillersMiles.co.uk, a repository for identified phishing attacks, for the term "rapport" there are 40 distinct emails which use Trusteer's anti-phishing tool "Rapport" as the brand to purport the attack. Most of these claim to the victim that the bank in question is deploying a new security measure, and the user has to either verify their credentials (phishing) or download the tool (Trojan). Similarly, by searching for the term "avoid phishing" brings 101 different emails where the attacker warns the victim to phishing attacks. (Oxford Information Services Ltd 2010). One particular email states:

"Due to the recent phishing attacks and attempted identity theft associated with them, we have decided to verify identity of our members." (Millers Miles 2010)

This email relies on the victim having heard of phishing attacks and banks being targeted, but not really understanding the nuances of what a phishing attack is; and thus, ironically, becoming the victim of a phishing attack.

The core issue is that phishing attacks rely on the human-being as the "weakest-link", not on a technical fallacy. Hence, we must analyse what affects the user's response to phishing attacks and tackle the issue at its weakest link.

3. Desensitizing users

A well established psychological process is defined by desensitisation, and as phishing attacks target multiple components of a system, especially the non-technical, human, component, desensitisation cans play a vital role in the success of the attack.

The non-technical side of phishing attacks relies on conning a user. With the advent of more advanced threats in IT, security software has had to get increasingly advanced. Similarly, as the quantity of threats has increased, so have the messages which a user sees warning him that a threat was blocked, or warning the use that there may be a threat. This repeated exposure to warning messages and security windows leads to a desensitisation of the user.

Desensitisation is the process whereby a reaction is attenuated due to repeated exposure to a stimulus. It is often employed as a technique to combat a phobia, by repeatedly exposing a patient to the stimulus.

Psychological desensitisation is sometimes referred to as the opposite of addiction. From a biological point of view, desensitisation if believed to be caused by neurons becoming less sensitive to dopamine, a chemical messenger similar to adrenaline. (Addiction Science Research and Education Center, University of Texas 2009). At a more abstract level, desensitisation refers to a subject becoming less sensitive or less perceptive to a specific stimulus after being exposed to that (or similar) stimuli over time.

While widely used as a technique to combat phobias, (Fear Free Flying n.d.) desensitisation has been the topic of recent debates with regards to violence in videogames. (MailOnline 2006) A study at Iowa state university was the first in its kind proving a link between violent videogames and user desensitisation to real-world violence (Carnagey 2006). That said, very little research has been done on the link between desensitisation and user-interfaces (even though there are large amounts of data on human reactions to interfaces).

In psychotherapy, systematic desensitisation involves a patient creating a hierarchy of fear, from least fearful to most fearful. For example, a patient with a phobia of spiders could have the least fearful being a small spider 5 meters away, while the most fearful being a 5 meter spider at a close distance. The patient is then repeatedly exposed to the cause of their fear, starting with the lowest level – at the same time, support is given to help the patient cope with the fear. After repeated exposures, the patient becomes less sensitive to the stimulus causing the phobia, and they can move up in level to the next “fear level”. The patient does not always need to be exposed directly to the cause of their phobia, for example, with a fear of spiders merely thinking of a spider or looking at a picture of a spider would be sufficient at the lower fear levels.

With relation to security, it was the researcher’s hypothesis that displaying a large quantity security warning messages, confirmations and password requests would cause just such an effect – i.e. it would be detrimental to the very security that the alert would be trying to

maintain. That is to say, the phobia is an almost subconscious fear of being cheated or scammed – and the desensitizing factor is the repeated security warning messages. When a new user sees their first warning message, their fear-factor kicks in and they give it their attention; but as time progresses, and the warning messages become routine, a combination of desensitisation and laziness kick in thus making the normal error messages routine. This could potentially be exploited by an attacker by replicating just one such message and targeting users that see that message as part of their routine day to day activities. (Nicholas L. Carnageya 2005)

As such, an experiment was set up in an attempt to collect empirical evidence. The evidence collected would appear to support the hypothesis. The users which were part of a group of users routinely exposed or “desensitised” released their credentials more than those who had not.

4. Experimental Design

4.1 Preliminary Considerations

Security popup messages that appear on a daily basis are not the user's primary concern when the message appears. As a system administrator, the researcher had observed that pop-ups tend to interrupt the normal workflow at the time they appear. As such, giving a user a task to complete, then interrupting it with a popup and observing how the user reacts is the ideal scenario in replicating what would regularly occur. At the same time, or in conjunction with this, data collection was required to stratify the subjects in order to eliminate as many external variables as possible, leaving only the subject's "history" in seeing the pop-ups as the independent variable (Creswell 2009). Additionally, verifying the user's history in seeing the popup would be ideal. As such, a questionnaire/survey seemed to be the logical choice, as it fulfilled two of the three criteria; nominally it collected the data required to stratify the subjects and it provided the users with a psuedo-task that would then be interrupted by a security popup.

After reviewing a range of methods on how data could be collected with regards to desensitizing users, it became clear that a conventional test or quantitative survey alone would not suffice in disproving the hypothesis. As such, the experiment was based on blend of a psychology experiment, a phishing attack and a method used in marketing to test effectiveness of new systems

In an attempt to eliminate subjective bias, a double-blind experimental methodology was selected. Following double-blind protocol, until all the data was collected from all the subjects, the researcher did not know which subjects were in which group, or even how many subjects were in each group. This allowed collection of the data while minimizing the risk that the user's responses were being influenced by the fact that their actions were being observed.

Based on controls from an experiment on DNA, the responses from each subject are interpreted independently from other "items of evidence or reference samples" (D. Krane 2008) to identify what operating system the subject was using and most familiar with.

Secondly, by taking a page out of advertising tactics, it was decided to use a variation of split testing (also known as A/B testing) (Vertster 2009). A/B testing tests the effects of single-variable changes, and is usually used to improve response rates on websites. That is to say, two groups are presented slightly different content and the response rates are recorded for both. As the difference between the two groups is the variable, any difference in results is depended on and due to that variable. (Ronny Kohavi 2009) The multivariate variation employed in the experiment essentially comprised of multiple A/B tests in close succession. Thus all the variables were blind to the researcher until all the data was collected. The principal variable being tested was the operating system that the respondent was using. The

advantage of split testing (aka A/B testing), in addition to simplicity and limiting the factors to a single variable, was that the results could be analysed using Pearson's Chi-Square tests and then compared.

The experiment was set up as a survey; as this was the ideal method of collecting the data required from a large enough sample of the internet connected population. Additionally, it gave the respondents a "fake" website to base the phishing attack on under a false pretence; thus making the phishing attack as realistic as possible.. The survey was self-administered; based on Bourque and Fielder, who "believed that people are more likely to give complete and truthful information on such sensitive topics in a self-administered questionnaire" (Fielder 2003).

4.2 Survey Design

A survey was selected as it had two further pragmatic advantages: the "economy of the design (...)[and the] rapid turnaround time" (Creswell 2009). This was further complemented by running the survey online, as it allowed the subject selection to be pseudo-random (i.e. Anyone from the global population of computer users could complete the survey) with minimal additional cost. Having 3, 30, 300 or indeed 3,000 respondents would require negligible additional resources in data collection.

By filling out the survey online/remotely (as a pose to on a local machine), there was less risk associated to the user as there was nothing to install on the user's machine. As such, the user would be more likely to start the questionnaire.

In order to surf online, a user would require a browser, most of which have JavaScript as standard. As such, the data collection was designed to require only a basic HTML browser and JavaScript. Additional content (e.g. flash) was not included given its additional cost to design, but more importantly, its restriction of the user-base that could respond to the survey. As the data was collected through an online survey, the user did not need to install any additional software.

This also aided the realism of the phishing attack, as phishing attacks in the wild are done on remote machines. It is important to denote that this relates to phishing attacks, where a user relinquishes data/information as a pose to a trojan infection, where the user is tricked into install malware.

Another decision faced was whether the survey needed to be semi-supervised or unsupervised, as both methods have advantages and disadvantages (Fink 1995). Fink identified four types; "self-administered questionnaires, interviews, structured record reviews and structured observations" (Fink 1995). While the semi-supervised approach allows the administrator to answer any questions the respondents may have, this can bias answers and at the same time "samples are frequently unrepresentative" (Fielder 2003). On the other hand the unsupervised approach meant that "there is no direct information on

the answerability of questions" (Fielder 2003) ; and one "lacks control over who responds" (Fielder 2003) . That said, having no control over respondent choice is actually a positive factor, as it inserts a bit of randomness to the selection process, which in turn means that the bias caused by respondents being only those that the administrator frequents is reduced (Lee 2008). Interviews would not allow us to observe how a user would react to the popup, while observing the user in a controlled environment would affect how the user reacts. Additionally, any form of supervision would detract from the realism of the phishing attack; thus inserting further variables from a real phishing attack. Finally, adding a researcher would have compromised the blind nature of the experiment.

Designing a questionnaire to ask a group of volunteers to answer it in the hope that a security popup would appear in precisely the right moment would have been impractical, unscientific and not a realistic simulation of a phishing attack. Similarly, creating a situation where the subjects' machine would be forced to display a security warning would be problematic both from a technical point of view and from an ethical/legal point of view. From a technical point of view, tracking how the user decided to act on the warning would have been difficult. In addition, even tracking whether the pop-up had been displayed would have been technically difficult.) From a legal and ethical point of view (would it have been ethical or legal to cause a system harm to collect data in an experiment?). Hence, simulating the warning/popup became the only reasonable alternative.

That said, simulating a window created two new issues, nominally, how to simulate a window that a user had seen repeatedly; and how to measure the user's reaction to the window in a quantifiable manner.

Apple's elevation request window provided just such an opportunity, as the window (and indeed the concept) was added as an attempt to increase security (Grimes 2006). The popup was displayed to the user every time they attempted to install or run certain programs, edit certain file/filenames, or in fact anything that required administrative privileges (escalation). Additionally, when a user changed his/her password, there was the potential (especially on network accounts if the password was changed while the user was not logged in) that the local keychain[4] fell out of sync with the account; thus prompting the user to enter their password even when normal, non-administrative, tasks needed to be performed. The two windows are very similar, and to a busy user attempting to perform a task the two may be mistaken.

Consequently, it was decided that by presenting the user with a simulated elevation window during the questionnaire and recording whether the user entered their account details (or closed the window) would adequately represent the "real world" phishing interaction in a similar scenario after the user had been exposed to repeated alerts of the kind.

[4] Keychain is a password manager native to OsX

To validate the conjectures, a second group was needed to be exposed to the same popup window. That said, the second group should not have been exposed to the stimulus of that specific security popup. Hence, the division between Windows and Apple users was a perfect sample case. A predominantly Windows user would not have been familiar with Apple's elevation popup. On the other hand, a predominantly Apple user would have been quite familiar with the popup. Thus, the independent variable[5] would be the user's operating system. Additionally, identifying which operating system a user was accessing the site from could be done technically, and thus without any human bias.

To further enforce the distinction between a user that had been exposed and one that had not, a section of the questions set in the questionnaire were designed to ensure that the user fell clearly into one of the two categories (exposed and non-exposed). Three questions were devised to this purpose.

The first question was designed to ask the user directly what operating system they predominantly used.

Table 4-1: Predominant OS Question

What Operating System do you predominantly use? • Apple OsX • Microsoft Windows • Linux • Other • Don't know • Prefer Not Answer

While the study needed to compare OsX versus other operating systems, it was felt that by giving the option of "Apple OsX" versus "Other" would prove detrimental and may influence answers. Additionally, by providing the other options, it may have been possible to analyse the data with more accuracy.

[5] Independent variable is the aspect which is changed during the experiment. The effect of the change of this single variable is then recorded an analysed.

The second question to ensure a distinction of predominant operating system relies on the length of time the user had used the operating system for.

Table 4-2: How Long Has OS Been Used Question

How long have you been using (OsX, Windows, Linux, Other) as your main operating system? • Less than a month • 1-6 Months • 6-12 Months • Over a year • Prefer Not Answer • Don't know

This question was included because a user who predominantly used Apple, but at the same time had just starting using apple less than a month before would not have been desensitized the same way as more Apple experienced users.

Finally, a slightly more technical question was posed, designed to find a user's predominant operating system without directly asking them. By asking a user how they accessed their home folder, it was possible to differentiate between Windows, Apple GUI or *nix (and apply terminal user) users.

Table 4-3: Home Folder Access Question

How would you access your home folder? • Click on Go>Home • Click on Start>{username} • Type cd ~ • CMD+Shift+H • Other • Don't know • Prefer Not Answer

There are two main factors to the user's response; which operating system the user is most used to and experience. If the user chose Go>Home, it would have been safe to assume that they are used to OsX as this was the default method to access a home folder in the Apple OsX GUI. Similarly, CMD+Shift+H was the shortcut in the apple operating system to get to the home folder. 'cd ~' was a bit more complex, as this could be used both by advanced Apple OSX users in the terminal window, as well as Linux users. Start>{username} was the most common method for windows users. Additionally, three further options were given; "Other" and "Don't know" for people that used other operating systems or didn't know as well as "Prefer Not Answer" as per CUREC guidelines.

The three questions were located just before the phishing popup, such that by the time the pop-up appeared, the respondent would have already answered the questions. As such, the questions to verify the user's predominant operating system were placed 6th, 7th and 8th.

Additionally, a question was created asking the user how often they saw a "keychain reminders", this can be seen in Table 4-4: Keychain Reminder Question.

Table 4-4: Keychain Reminder Question

How often do you see keychain reminders? • Never • Monthly • Weekly • Daily • Don't know • Prefer Not Answer

Keychain reminders were Apple's window requesting users to enter their password to unlock the keychain. A user who had never used an apple machine should have selected "Don't know", while a user who had used apple could have selected anything. I.e. an apple user who did not know that the specific popup window was called a "Keychain Reminder" may have selected "Don't know" even though they may have seen the window every day for the previous year. This would have been further highlighted by users selecting "never" – technically impossible - as the user would have seen the keychain reminder window only a few questions before. As such, this question was given less weighting; but included as through the answers would allow the research to show how many Apple users didn't know what a keychain reminder is.

The remainder of the questions were designed to serve two purposes, in part as filler to set the scene of a survey and in part to stratify the users. Most of the questions in the survey were set up as closed-ended questions (i.e. questions with a finite set of answers); the benefit of this was that they were easier to standardize, and the results lend themselves well to statistical analysis[6].

There were some open-ended questions in the survey – i.e. questions after the popup has been displayed. These were included to reduce the "suspiciousness" of the popup, which could have otherwise distressed the subjects (as per CUREC/Psychological society guidelines). The responses to the open ended questions were not intended to be analysed.As such, a total of three filler questions were included, with the answers not intended for analysis. These were:

[6] (Fink 1995)

Table 4-5: Filler questions

How many times do you shop online? • Never • Monthly • Weekly • Daily • Don't know • Prefer Not Answer
What is your most visited website? <Open Ended Answer (text box)>
How many spam emails do you receive per day? • None • 1-5 • 6-10 • 10+ • Don't know • Prefer Not Answer

These were selected due as being topics most internet users would associate with, but without any real relevance to the data collection.

Finally, a set of stratifying questions were included before the phishing attack was presented, aimed at eliminating the possibility that the responses were influenced by other factors. These were taken from existing surveys and books (Connolly 2006) (Couper 2008) (Creswell 2009) (D. Krane 2008) (Fink 1995) (R. Kohavi 2007) (Ritter 2007) (Ronny Kohavi 2009) (www.data-archive.ac.uk 2008).

Table 4-6: Stratifying Questions

What is your age? • Between 18-25 • Between 26-35 • Between 36-45 • Between 46-55 • Between 56-65 • Between 66-75 • Between 76-85 • Over 86 • Prefer Not Answer
How would you describe your computer comfort level (IT literacy level)? • IT Native • IT Expert • Intermediate • IT Novice • Prefer Not Answer
What is your sex? • Female • Male • Prefer Not Answer
Please indicate the highest level of formal education attained? • No High School • Some High School • High School Graduate • Some College/University • College Graduate • Postgraduate Degree • Prefer Not Answer
How many children do you have? • 0 • 1 • 2 • 3 • 4 • 5 or more • Prefer Not Answer

As such, by collecting data on a user's predominant choice of operating system along with some data on verifying this and directly asking respondents how often a specific security warning was seen the subjects were divided into two groups; the experimental group and the control group.

The experimental group consisted of users that had seen the password request popup before the experiment, while the control group consisted of users that had not been exposed to the popup window as part of their routine use of a computer. Due to constraints on cost, time and technical feasibility, it would have been unfeasible to test every single internet user during the experiment, and to classify them into the control group or the experimental group. As such, the test was run on a subset of the population. This followed the proven standard practice, while offering some statistical assurance. (Creswell 2009)

Due to the speed at which technology in general develops, in particular the speed at which the security landscape changes, the survey was designed to be cross-sectional. That is to say the data was collected at one point in time as a pose to being collected over time (longitudinal). In this way, it further ensured that the variable being tested didn't change during the course of the experiment. For example, Apple could have disabled the specific warning message which was used to test user's reactions. As the experiment attempted to gauge the effect of the desensitizing pop-ups, changing the pop-ups halfway would lead to unpredictable results.

Additionally, after at least a year of seeing the popup, the user will either be desensitised or the effect would be considered to be negligible. I.e. If after a year of seeing the popup almost every day the user still reacted to the popup the same way they did on the first day (before seeing any pop-ups), then it would have been highly unlikely that their reactions would alter after a more pop-ups. Thus, the experiment was set to collect data for a brief period.

4.3 Technical Design

The data was collected through an online phishing attack, embedded inside a questionnaire. Due to the sensitive nature of performing an exploit/attack, CUREC approval was required. In order to gain CUREC approval, the CUREC committee needed to approve the survey, as well as two consent forms and two explanations to the experiment. Both sets were to be shown to the respondents. One set was designed to be shown at the beginning of the experiment under a false pretence. The second set, was designed to be shown at the end alongside the debriefing (as per CUREC guidelines). This was done as explaining to respondents the true purpose of the experiment before the experiment began would have altered the responses. CUREC approval was granted to the experiment.

Given that several questions were designed to revolve around operating systems, and that the differentiating factor between the test group and the control group was almost precisely what operating system they use, the pretence under which the survey was advertised was

"Operating System Survey". The initial explanation and consent form shown to users (i.e. the landing page) is shown in Appendix 1. The final debriefing and real consent form is shown in Appendix 2. Both consent forms were based on a standard sample consent form by the UK Data archive (www.data-archive.ac.uk 2008).

The first four questions are designed to give the illusion of a realistic survey as well as to stratify the respondents. Question 5 was the first question that revolved around operating systems, and asked how computer literate the user was. While this alone was not critical, this can then be cross referenced with question 7, which asks how long they have been using a particular operating system. Question 6 directly asked what the main operating system the respondent used predominantly – this would then be correlated with the operating system that the server detected using the HTTP command (HTTP_USER_AGENT). In theory, it would be possible to go as fine grained as detecting the browser; but this did not have a relevant impact on the results. Questions 8 and 9 were the same, but in front of question 8 a popup was displayed, i.e. the phishing attack. If the user moved the popup out of the way, and continues with the survey, question 10 became question 9, and the survey continued as normal. On the other hand, if the popup was filled out (or closed), the user could then answer the question hidden underneath and continue the survey as normal. Duplication of the question was done to resolve a technical issue, and had no real impact on the survey's questions (or the experiment's results, as a phisher could do the exact same thing). Questions 10, 11 and 12 served as padding, and the results were not designed to be analysed as they came after the attack – where some respondents may have closed the window. That said question 13 was quite important – in fact, only someone familiar with OsX would know what a keychain reminder is, and even so, they would need to be very familiar with OsX; hence, any answer except Never, Don't know or No Answer would imply a user familiar with OsX. This was designed in such a way that if a user answered it would benefit in honing the accuracy of the previous answers, but if they did not answer (e.g. closed the window) it would not negatively affect the validity of the response data. Finally, question 14 attempts to verify the password given in the phishing attack. This is done by displaying the first character typed into the password field in the phishing popup. As with question 13, a user not answering the question would not affect the validity of the response. It must be noted that if the user did not type anything in the phishing window, this question was not displayed.

The actual phishing window was made using a combination of DHTML and JavaScript. This ensured that a popup blocker wouldn't affect it as there was no new window, nor was there an invocation of the "popup" (Window.Open) command. At the same time, this emulated a real window floating over the original window and allowed the respondent to move it just like a real window. To create the window, a screenshot was taken of a real Keychain authentication request window - something anyone with a Mac running OsX can do, including a potential phisher. This was then used as the background for the form, the text-

boxes overlapping exactly those in the image and the buttons being replaced with HTML images-maps and a JavaScript submit function. Additionally, a timer was added to all the questions survey, the intention being that the time taken by the user before acting when the popup appeared would be less if the user didn't read the pop-up's message (and longer if they did).

While the survey itself was closed in accordance with the timescale approved by CUREC, a non-functional copy was created after the data-collection ended. This can be found at: http://www.europoli.org/ThesisSurveySample/

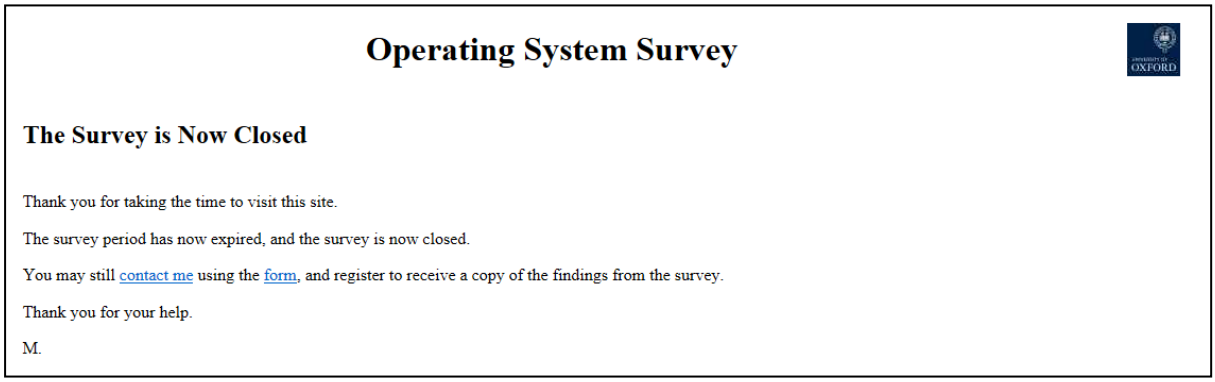
The credentials required to access this are:

Username: Oxford Password: K3llog

The survey found at this location was purely for demonstration purposes for this document, is non-functional and was not part of the data collection (though an identical copy). During the experiment, this was not available to the respondents. Additionally, the source code is included in an attached CD-rom[7].

The message displayed to users who access the original survey link is displayed below in Figure 4-1: Survey Closed Message.

Figure 4-1: Survey Closed Message



[7] The password to access the database backend has been redacted from the source code. Additionally, the .htaccess and .htpasswd files which store the MD5 hashes of the credentials to access locked sections have been removed.

5. Data Collection

The experiment was live for a 2-week period, in which 310 responses were collected following CUREC guidelines. The respondents were found in a variety of places to reduce the potential of introducing another reason for any correlation found.

During this period, 310 responses were gathered, 237 from respondents on Windows machines, 69 from Macintosh machines and 4 from Linux/other machines. As per CUREC guidelines, respondents had the option to withdraw from the experiment at any time, at their request without a reason.

Six responses were withdrawn after the debriefing at their request, and as such got deleted. Potentially, the answers got withdrawn once the respondents realised that they had been "victims" of a phishing attack, and felt that their personal information was at risk. It must be noted that no personally identifiable information was gathered or stored linked with the data, as per guidelines.

Further following established guidelines, and in an attempt to eliminate bias due to the selection of the respondents, the respondents were not selected by the researcher; but by posting multiple adverts for the survey. Adverts to the survey were posted in a selection of sites; some not even known to the researcher. These included:

- www.mbclub.co.uk (Mercedes Enthusiasts Website, with authorisation from site administrators, albeit with regards)
- www.facebook.com (Social networking site)
- 100,000 Bulk Untargeted RON hits (paid-for adverts on random sites, unknown to the researcher)
- www.ducatisti.co.uk (Ducati Owners Club Site, with authorisation from site administrators)
- www.pocuk.co.uk (Pajero Owners club – link withdrawn by site administrators within 24 hours of posting)
- www.torrentleech.org (BitTorrent site)
- www.bitme.org (BitTorrent e-learning site)
- www.stumbleupon.com (pay-per-click advert site – target sites unknown to researcher)
- ACS International Schools' email system (email sent from a fake account, with authorisation from the System administrator and headmaster of the school)

This created a range of visitors to the survey – the 100,000 untargeted adverts (probably pop-unders in other sites, but as this was subcontracted to several advertising firms this cannot be verified) creating the bulk of the visitors, and the rest complementing the numbers. In total, almost 110,000 visitors saw the landing page for the survey. When one considers how many people actually clicked-through and completed the survey, this meant

a return of just over 0.27%. This was especially good when compared to the average click-through return of 0.01% (Cherecwich 2009).

From a technical perspective, as the popup required JavaScript to be active, any data collected where the user did not have JavaScript active, as valid as it may be from an overall security perspective (i.e. what percentage of users don't have JavaScript enabled when browsing untrusted sites) – it is not relevant to this experiment as the decision point, or the pivotal factor, needs to be the popup and the user's familiarity with the looks of said popup. There were 2 respondents who did not have JavaScript enabled. Those answers were excluded. Interestingly, all were Windows users.

From an operating system perspective, the experiment was not targeted at what specific operating system the user was familiar with, be it Windows XP, Mac Os 10.5.7 or one of the many flavours of Linux. Hence, even though data was collected on precisely what operating system the respondent was using, the results were then aggregated into two groups; people who are familiar the real keychain popup (i.e. OsX users and familiars) and those that were not. Of the 69 respondents accessing the survey from an Apple machine, only 3 stated that they were predominantly Windows users (4.3%). On the other hand, of the 237 respondents on a Windows machine, 4 (1.7%) stated that they were predominantly Apple OsX users, half of which disclosed their password in the phishing attack.

To accurately and impartially judge which Operating system was the predominant one for a user, a sliding scale was devised. By using the responses to the 2 questions on operating system ("How would you access your home folder?" and "What operating system do you predominantly use"), along with the operating system reported by their browser a scale from -3 to 3 was adopted. For every answer which supported OsX, +1 was added to the tally. For every blank, "I don't know" or "Prefer Not Answer" answer, 0 was added, while other operating systems subtracted 1. Thus, the three numbers formed a tally from -3 to 3 depending on how familiar the respondent was with the Apple OsX. Thus, 58 predominantly OsX users responded, and 246 predominantly non-OsX users responded.

Once the data was collected, trends were mapped using the data, in an attempt to find matches. The trends supported the hypothesis – i.e. the responses support that users familiar with the keychain window were more prone to giving out their password to an untrusted website (like the one set up for this experiment). It was hoped that no other trends would match, that is to say, not all the "password-givers" were be male, or under 35, or with college education – i.e. a random distribution would have been ideal. That said, given the fairly small dataset (just over 300 responses, compared to 6,000,000 people on earth) getting a normal distribution proved difficult.

In total, three operating system families were reported by the browsers, Linux, Windows and Macintosh. The distribution between what the user reported and what the browser reported is shown below.

Table 5-1: Comparison Table of User and Browser Reported O.S.		User Reported Predominant OS					
		I don't know	Linux	Prefer Not Answer	OsX	Other	Windows
		Freq	Freq	Freq	Freq	Freq	Freq
Browser Reported OS	Linux	0	3	0	0	0	1
	Macintosh	1	0	1	50	0	9
	Windows	1	5	2	4	1	175

Similarly, the three families of operating systems reported by the browser would appear to indicate that 6 users accessing the survey from a windows platform released their credentials to the phishing attack, while 10 respondents running Macintosh did so. Curiously, none of the 4 Linux respondents gave their credentials.

Table 5-2: Comparison Table between Reported OS and Password Given		Password Given	
		False	True
		Freq	Freq
Browser Reported OS	Linux	4	0
	Macintosh	53	10
	Windows	231	6
Predominant OS	Apple	48	12
	Not-Apple	240	4

Displayed in another way, half of the users on a windows platform that reported having used primarily Macintosh released their credentials. On the other hand none of the users that took the survey from a Macintosh but were primarily non-Macintosh users released their credentials.

Table 5-3: Comparison Table Between Predominant OS and Password Given				Password	
				False	True
				Freq	Freq
Predominant OS	Apple	Browser Reported OS	Macintosh	47	10
			Windows	1	2
	Not-Apple	Browser Reported OS	Linux	4	0
			Macintosh	6	0
			Windows	230	4

A set of all the raw data can be found in Appendix 4 : Raw Data & Source Code (CD-Rom)

One final point to keep in mind was that the figure of 310 responses did not take into account the responses from those respondents that chose to withdraw their answers at the end of the survey. As per CUREC guidelines, no data was collected from those respondents.

Finally, while the demographic questions were not absolutely pertinent to this experiment, ensuring that that age, sex, or educational background were not contributing factors (and if they were, ensuring these are recorded) was important.

Table 5-4: Respondent Sex

		Freq	Col %
Sex	Female	102	38.9%
	Male	156	59.5%
	N/A	4	1.5%

Table 5-5: Respondent Age Group

		Freq	Col %
Age Group	18 - 25	85	33.2%
	26 - 35	50	19.5%
	36 - 45	42	16.4%
	46 - 55	40	15.6%
	56 - 65	37	14.5%
	66 - 75	2	0.8%

Table 5-4: Respondent Sex shows that there was a majority of male respondents, (by 20.6% plus-or-minus 1.5%). Similarly, the respondents ages would appear to be incremental – as the age of the group increased, the number of responses decreased (see Table 5-5: Respondent Age Group).

As such, the data collected was sufficient and above the expected 0.01% return and came from a variety of sources. The pop-up window appeared to almost all the users; and the responses came predominantly from Windows users reflecting the dominance of the Windows platform when the experiment was run.

6. Data Analysis

A full copy of the raw data can be found in Appendix 4 : Raw Data & Source Code (CD-Rom).

Once the data was collected, it needed to be filtered and analysed. As such, of the 310 responses obtained, 2 responses had to be excluded from the analysis, leaving 308 responses usable. Of the 308 responses, the majority of respondents who gave their credentials had been desensitized to security popup windows.

6.1 Data Filtering

Before the data could be analysed, the results which were irrelevant (due to technical issues) needed to be removed from the 310 results obtained. Thus, results where the respondent had JavaScript disabled, or null responses – typically from web-crawlers – were removed. There were two results obtained where the user did not have JavaScript enabled – these were not included in the analysis. Similarly, null responses, where a "bot[8]" had crawled the site but not answered any questions, or left any time-footprint on the various pages, were automatically excluded. In this case, no entry was made in the SQL answers table, and as such, these instances did not impact the 310 responses figure.

As a result of the filtering, 308 results from the survey were analysable. Of these, 48 responses were from cases where a user exited before the survey was complete. AS during the design phase it was decided that any respondent who saw the popup would constitute a response, three-exit time frames were devised. There were three such "time-frames" that the user could have exited the survey, before the pop-up, during or after. Considerable thought went into whether the data these provided was statistically valid.

Clearly, as per the design, any responses which passed the pop-up window or exited at the pop-up window were valid. These would represent users that closed the survey due to the popup, closed the popup, simply ignored it, or answered to the phishing attack.

At first it was thought that a user exiting the survey before the popup even showed rendered the response statistically invalid as the actual attack/test had not been delivered. This could have happened in a circumstance where the user got bored and closed the window. That said, one of the comments received highlighted that this was not the case, and that these answers should be included. One respondent stated that that when he opened the first question to the survey, a tool he has running on his browser warned him that the word "phish" appeared in the JavaScript source code. As such, he felt it was safest to close his browser and not proceed. As such, the user had security concerns with the phishing attack, even before the attack was carried out.

Similarly, users exiting the survey after the popup was displayed have the same scenario, but with the exception that their response to the challenge (i.e. the popup) has already

[8] A "bot", from robot, is an automated tool or script that requests all pages in a website. This is commonly used by search engines to index websites, but also by malicious programs.

been recorded. One user commented that he did enter his password, but didn't think it was a security issue until the survey "guessed" his password's first character. Then he realised that there was an issue and decided to close the window and use the "contact us" feature. The interesting part is that although he did not trust the website, he still felt fairly confident voicing his concerns on that site, from within the site, to an unknown third party.

Finally – the case where the users closes their browser when the popup is displayed. These make up the bulk of "browser closes". In fact, by the time the question after the popup was displayed, only 237 candidates still had their browser open. That means that 71 respondents, or just over 23%, did not close the browser until after the popup. Of those, only 16 had given their username and password. In essence, this meant that 53 candidates, i.e. over three-quarters (76%), either realised that there was a phishing attack in progress or that keychain password request window could not be trusted – yet they still continued using the site and gave their answers to the survey.

6.2 Data Analysis: Desensitization Categorisation

By using the sliding scale, as designed in section 5, it was possible to gauge if user has been exposed to repeated keychain password request windows. To summarise the sliding scale, the responses need to satisfy two or more of the following criteria for the respondent to be classified as desensitised:

- Browser reporting OsX as the OS to complete the survey
- The answer to "What Operating System do you predominantly use?" is Apple OsX
- The answer to "How would you access your home folder?" is one of the following:
    - Click on Go>Home
    - CMD + Shift + H

For each of the above answers, a "point" is added to the tally, while a negative response would subtract a point from the tally. A user choosing to give no response did not affect the tally. This created a -3 to +3 scale; as seen below in Table 6-1: Scale vs Reported Operating System.

Table 6-1: Scale vs Reported Operating System

		Password		Operating System Reported								
		FALSE	TRUE	Macintosh			Linux			Windows		
		Freq	Freq	Freq	Col %	Row %	Freq	Col %	Row %	Freq	Col %	Row %
OS Sum	-3	100	2	0	0.00 %	0.00%	4	100.00%	3.90%	98	41.40%	96.10%
	-2	61	0	0	0.00 %	0.00%	0	0.00%	0.00%	61	25.70%	100.00%
	-1	79	2	6	9.50 %	7.40%	0	0.00%	0.00%	75	31.60%	92.60%
	0	2	0	2	3.20 %	100.00%	0	0.00%	0.00%	0	0.00%	0.00%
	1	28	7	32	50.80 %	91.40%	0	0.00%	0.00%	3	1.30%	8.60%
	2	9	3	12	19.00 %	100.00%	0	0.00%	0.00%	0	0.00%	0.00%
	3	9	2	11	17.50 %	100.00%	0	0.00%	0.00%	0	0.00%	0.00%

6.3 Data Analysis: Positive/Negative Response Based

From a generic phishing perspective, 16 respondents gave a username/password, equivalent to 5.19% of respondents - very much in line with a report by Sophos, a leading IT software security firm, that states that “Phishers are able to convince up to 5% of recipients to respond” (Sophos 2005).

Of the answers of those that did give a username/password, 10 of the users were on a Macintosh machine (65%), and a further two responded that their “predominant” operating system was OsX, and bringing the total to 12 of the 16 (75%). Interestingly, of the 4 non-OsX users left, 2 demonstrated knowledge of OsX in how they got to their home folder, thus bringing the number of “exposed” (and hence desensitised) users to 14 (87.5%). This really leaves only 2 of the 16 (12.5%) users who initially gave their password that haven’t been exposed to the desensitizing effect of the keychain reminders.

Figures as biased as 87.5% of users who have been exposed versus 12.5% of those who haven’t (in the context of the positive respondents) cannot be ignored, but the counter-test lies with checking if the opposite holds with negative responses. This can be seen in Figure 6-1: User Response by Exposure, Basic. That is to say, of those users who did not give a username/password, is there a trend as to exposure?

Of the 308 respondents, the majority - 292 did not submit a username/password. As displayed in Figure 6-1: User Response by Exposure, Basic, these are classified as the

negative respondents. Only 20% (or 58) were on Apple OsX machines – and of these 58, 18 did not select OsX as their main choice of operating system; bringing the figure down to 14% (40 users). The remaining users that did not give away their password are mainly windows users (79%).

Figure 6-1: User Response by Exposure, Basic

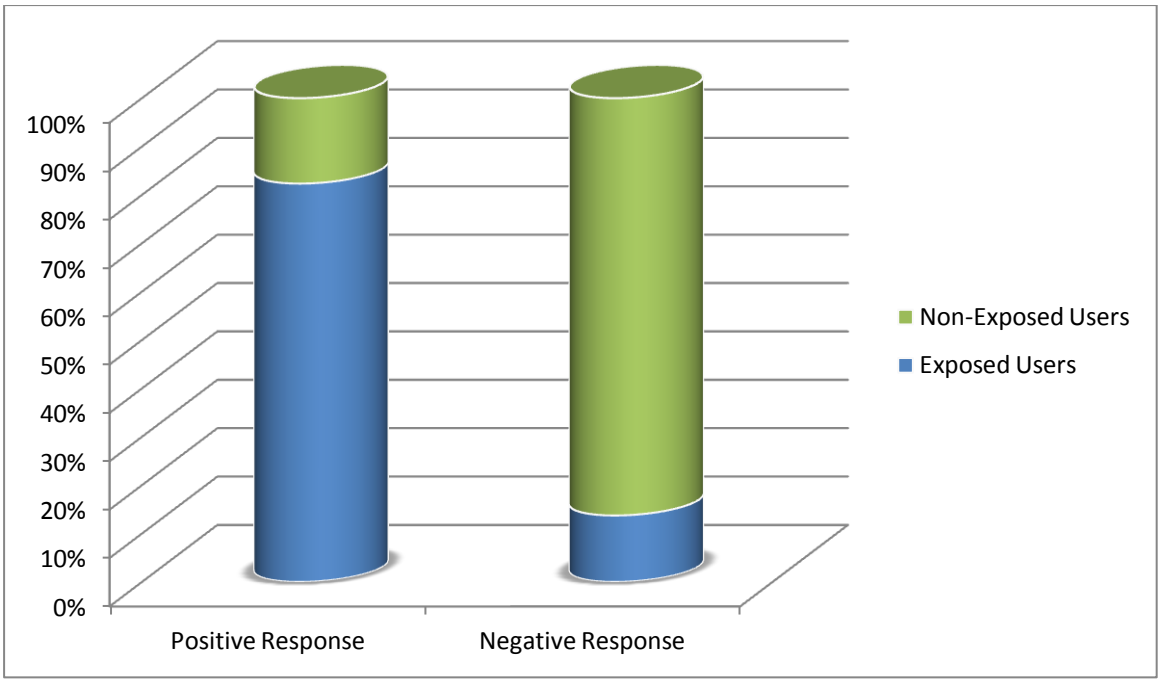


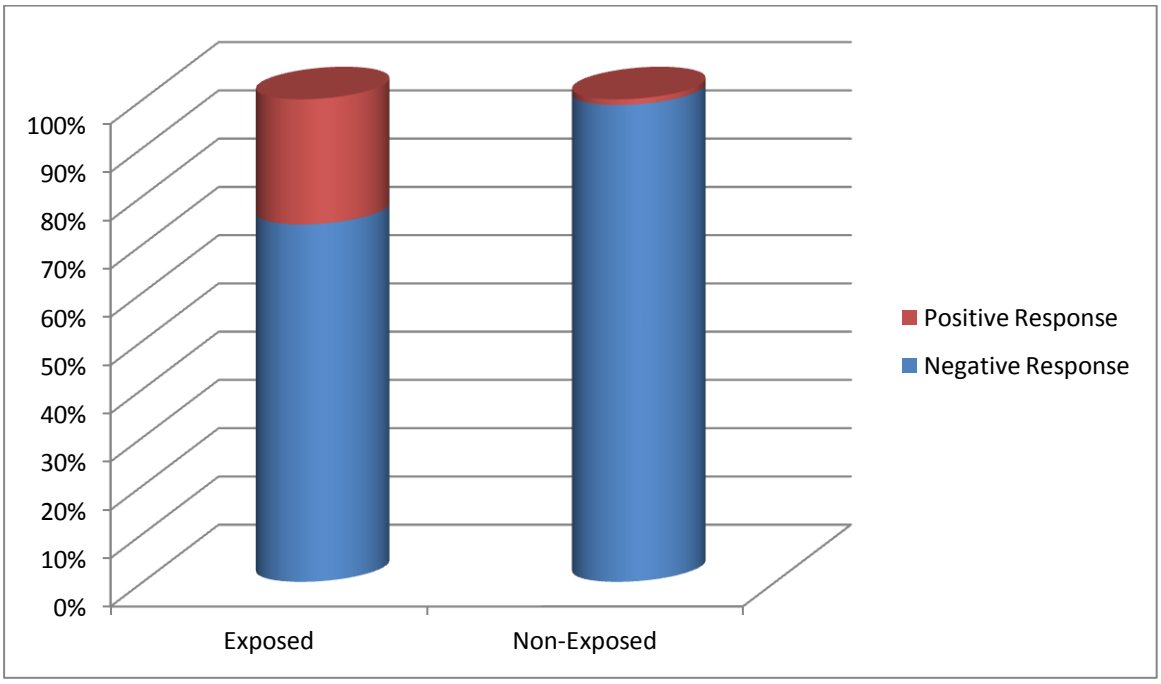
Figure 6-1: User Response by Exposure, Basic, above, shows figures with such a discrepancy which would appear to support the initial hypothesis. Not only have most of the positive responses come from the users exposed to the keychain window beforehand, but also most of the negative responses are composed of non desensitized users.

6.4 Data Analysis: Exposure Based

To ensure that the trend outlined above was statistically valid, further statistical analysis of the data was required. Instead of comparing the figures based on whether the responses were positive or negative to the challenge, the responses were compared based on "exposure". That is, what percentage of keychain exposed users gave a positive response versus a negative one? Additionally, what percentage of non-exposed users gave a positive response versus a negative one?

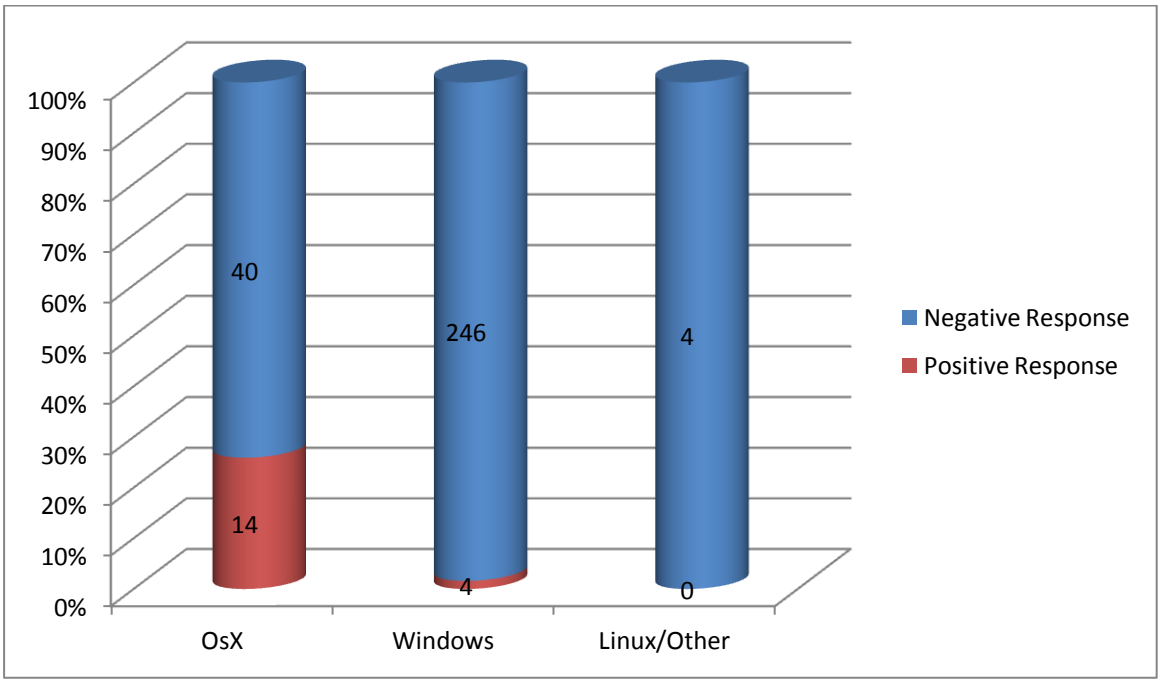
While only 1.6% of non-OsX users (i.e. non-desensitised) gave their credentials (well below the 5% from Sophos and APWG), 20% of OsX (i.e. desensitised) users gave away their credentials thinking that the popup window was authentic, as visible in Figure 6-2: User Response by Exposure, Advanced.

Figure 6-2: User Response by Exposure, Advanced



In order to see if there was a trend between operating system (rather than desensitization) and the user's response, the figures with the Windows users separated from the others (i.e. Linux users and those whose browser did not report the operating system) were drawn in Figure 6-3: User Response by OS.

Figure 6-3: User Response by OS



While Linux/other users would appear to be immune, it must be noted that only 4 answers from "Linux/other" were obtained. When compared to 250 Windows responses, 0-out-of-4 can be considered statistically similar to 4-out-of-250. As such, there did not appear to be a distinguishable difference between responses of Windows users and Linux/other users. The responses of the 4 users are shown in Table 6-2: Non-Macintosh Positive Respondents.

Table 6-2: Non-Macintosh Positive Respondents

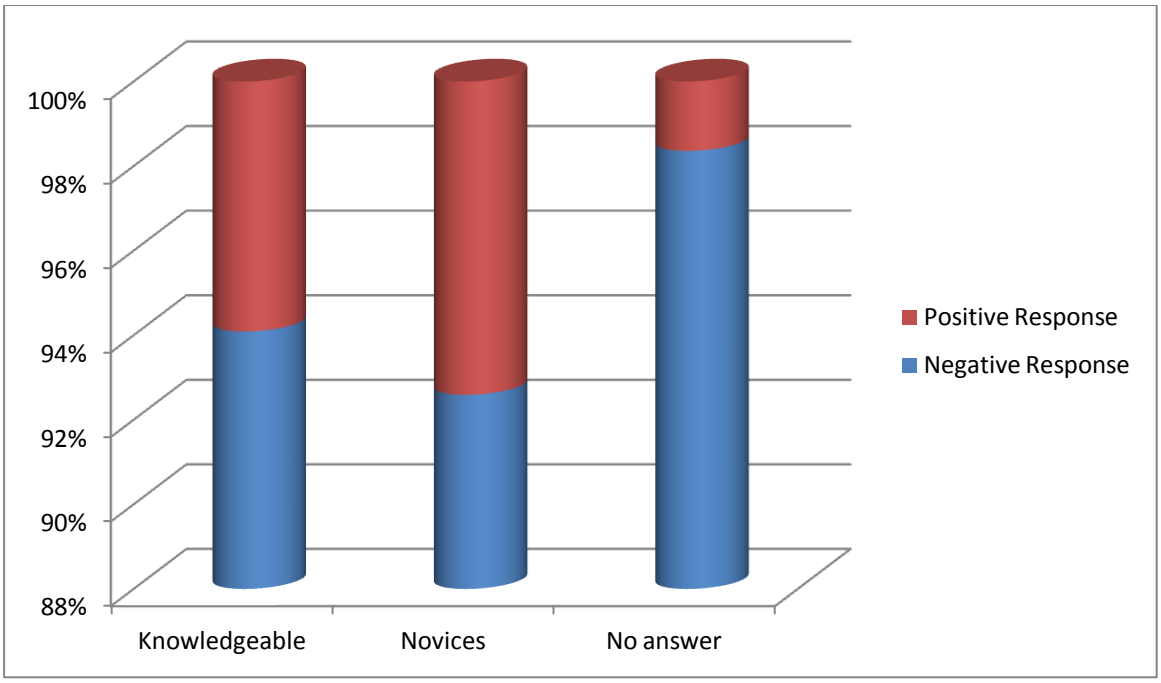
Age Group	Sex	Education	Children	IT Literacy	Predominant OS	OS Length	Home Folder	Keychain	Password	Phish Time	Operating System	OS Score
26-35	Female	N/A	0	Expert	Windows	Over a year	Windows	IDK	TRUE	16.718	Windows	-3
56-65	Female	Postgrad	2	Intermediate	Windows	Over a year	Windows	N/A	TRUE	35.468	Windows	-3
N/A	N/A	N/A	N/A	N/A	N/A	IDK	N/A	N/A	TRUE	87.828	Windows	-1
56-65	Female	High School	1	Intermediate	Windows	Over a year	Apple	Never	TRUE	28	Windows	-1

Interestingly, one user gave no answers other than answering the phishing popup. This may be a user that was merely attempting to see the phishing window, but this cannot be ascertained. Other than the one almost blank response, all the other non-desensitised positive responses came from female windows users with over a year of experience who felt that their experience was intermediate or above. That said, again, given the small sample size of non-desensitized positive responses, no trends can be drawn from this.

6.5 Data Analysis: IT Literacy Based

To further ensure that IT literacy was not a contributing factor to trending positive/negative answers, the users were split into three groups. The first group comprised of those that reported being at the intermediate level or above in IT literacy (knowledgeable), the second those that were novices and the third group those that either refused to answer, didn't know or left the answer blank.

Table 6-3: Response by Reported IT Literacy



To make the data more legible, the above table has been slightly rescaled, that is to say, instead of a scale from 0% to 100%, the scale is 88% - 100%. The IT savvy group and the novices group both gave similar ratios of positive responses (6.4% and 7.4% respectively). That said, the no-data group, had a predominantly negative response. Due to the nature of the response (i.e. no data), no trends could be drawn from this.

One of the respondents refused to release fairly general personal information (i.e. technical knowledge), yet they fell for a phishing attack. This may be due to lack of technical knowledge, or potentially distrust of the survey, but with only a single such response, no trends could be drawn.

Hence, with such a small difference between novices and knowledgeable (as per respondent definition) responses, knowledge was discounted as the main influencing factor.

6.6 Data Analysis: Chi Square (Desensitised vs. Password Given)

To further ascertain the results obtained, a Pearson Chi Square test was run, to test whether there was an association between the respondent giving their password and their predominant operating system (and hence whether they were desensitised).

The results of Pearson's Chi Square test can be seen in Table 6-4: Results of Pearson's Chi Square Test of Association Between "Password Given" and "Predominant Os" below.

Table 6-4: Results of Pearson's Chi Square Test of Association Between "Password Given" and "Predominant Os"

		Predominant OS					
		Apple		Not-Apple		TOTAL	
		Observed	Expected	Observed	Expected	Observed	Expected
Password Given	False	48	56.8	240	231.2	288	288.0
	True	12	3.2	4	12.8	16	16.0
	TOTAL	60	60.0	244	244.0	304	304.0

Minimum expected cell count: 3.158
% cells with expected count < 5: 25.0
p value: 0.000
Pearson's Chi Square statistic: 32.56
Degrees of Freedom (df): 1

Based on Table 6-4: Results of Pearson's Chi Square Test of Association Between "Password Given" and "Predominant Os" it become quite obvious that for the desensitized apple users the observed false responses (i.e. password not given) was significantly lower than the expected number of these. Similarly, the number on non-apple, non-desensitized used observed not to give a password was less than a third of the expected value. Further supporting the original hypothesis, the number of desensitized users observed as having given a password was almost four times that expected; while the non-desensitised users observed as not giving a password was above that expected.

Thus, all four possible cases (desensitized vs. non-desensitized and password given vs. not given) support the original hypothesis that the desensitization of users is the influencing factor; and that a desensitized user is more likely to give their password away during a phishing attack than one that is not.

Table 6-5: Password and Predominant OS Proportion

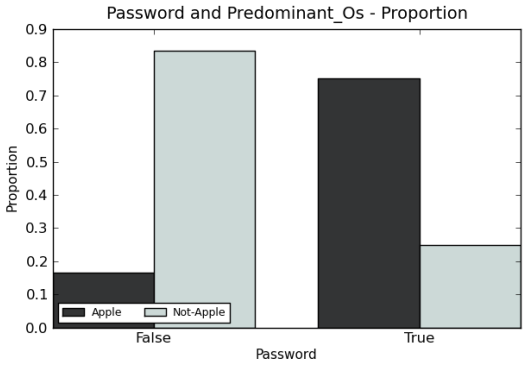


Table 6-6: Password and Predominant OS Frequency

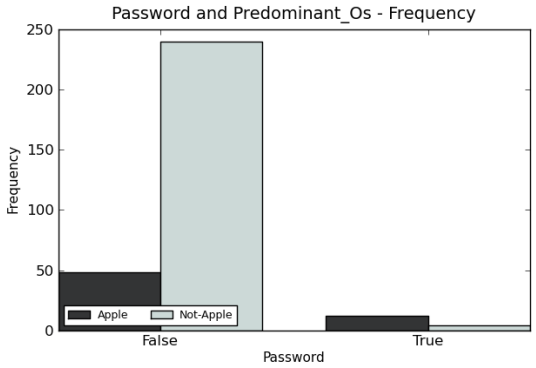


Table 6-5: Password and Predominant OS Proportion and Table 6-6: Password and Predominant OS Frequency show the proportion and frequency of response versus predominant operating system. Again, the portion of desensitised users giving away the password is larger than those not desensitised.

6.7 Data Analysis: Anova (Phishtime vs Operating System)

The Anova test allows a comparison of the means of the desensitised and non-desensitised users. No significant variance in the mean time of desensitised users signifies that there is no significant response time difference between the desensitised users and the non-desensitised users.

Table 6-7: Analysis of Variance, below, shows a relatively large p-value. Typically, a p-value below 0.01 would lead to the result being statistically significant, i.e. there is a difference in times. Similarly, O'Brien's test for homogeneity returns 0.975, which leads to the assumption that there is no significant difference in variance.

Table 6-7: Analysis of Variance - Anova (Phishtime vs Operating System)

Source	Sum of Squares	Df	Mean Sum of Squares	F	p[9]
Between	3153.4	1	3153.4	3.336	0.069
Within	231586.748	245	945.252		

O'Brien's test for homogeneity of variance: 0.975[10]

[9] If p is small, e.g. less than 0.01, or 0.001, you can assume the result is statistically significant i.e. there is a difference.

[10] If the value is small, e.g. less than 0.01, or 0.001, you can assume there is a difference in variance.

Table 6-8: Group Summary Details (Anova Phishtime)

Group	N	Mean	Standard Deviation[11]	Min	Max	Kurtosis[12]	Skew[13]	p abnormal[14]
APPLE	57	15.414	13.674	0.0	66.013	6.483	1.8	0.0
NOT-APPLE	190	23.895	34.204	0.0	410.062	87.262	7.935	0.0

Table 6-8: Group Summary Details (Anova Phishtime) further demonstrates that the values are not distributed in a normal distribution. The p-abnormality number is 0.0 (very low), and both the kurtosis and skew are vastly superior to the accepted limits.

Therefore, the operating system used by the respondent does not appear to have a significant impact on the time taken to respond to the phishing attack. Figure 6-4: Apple Phishtime Distribution and Figure 6-5: Non-Apple Phishtime Distribution below plot the time taken to respond to the phishing attack on a normal bell distribution curve. A bell curve is apparent in both cases.

Figure 6-4: Apple Phishtime Distribution

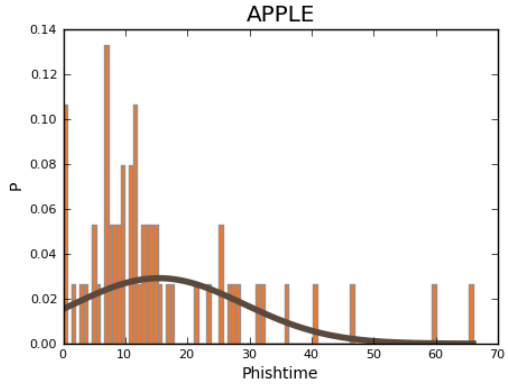
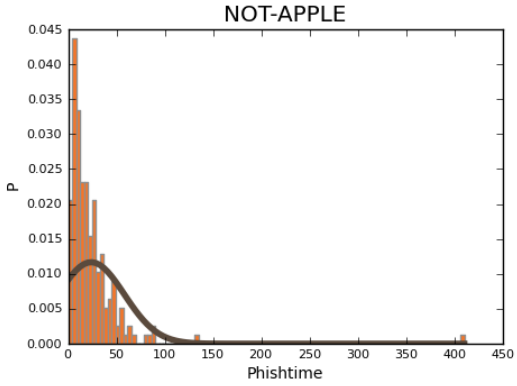


Figure 6-5: Non-Apple Phishtime Distribution



6.8 Data Analysis: Anova (Phishtime vs Password Given)

To further verify if the time taken to respond to the phishing attack influenced the respondents' decision, another Anova test was performed. The respondents were grouped depending on whether the password was given or not.

[11] Standard Deviation measures the spread of values.

[12] Kurtosis measures the peakedness or flatness of values. Between -1 and 1 is probably great. Between -2 and 2 is probably good.

[13] Skew measures the lopsidedness of values. Between -1 and 1 is probably great. Between -2 and 2 is probably good.

[14] This provides a single measure of normality. If p is small, e.g. less than 0.01, or 0.001, you can assume the distribution is not strictly normal. Note - it may be normal enough though.

"Phishtime" related to the time that the respondents took to respond to the question when the phishing window was presented, while password given related to whether the respondent gave data to the phishing attack.

Table 6-9: Analysis of Variance Table - ANOVA Phishtime vs. Password Given

Source	Sum of Squares	df	Mean Sum of Squares	F	p[15]
Between	530.982	1	530.982	0.555	0.457
Within	234209.166	245	955.956		

O'Brien's test for homogeneity of variance: 0.827[16] [2]

Table 6-10: ANOVA Phishtime vs. Password Given Group summary details

Group	N	Mean	Standard Deviation[17]	Min	Max	Kurtosis[18]	Skew[19]	p abnormal[20]
FALSE	231	22.323	31.441	0.0	410.062	101.987	8.508	0.0
TRUE	16	16.367	21.361	0.0	87.828	9.046	2.517	0.0

Figure 6-6: Password Not Given Phishtime Distribution

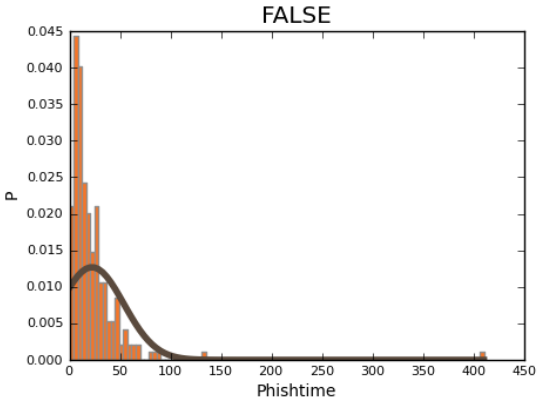
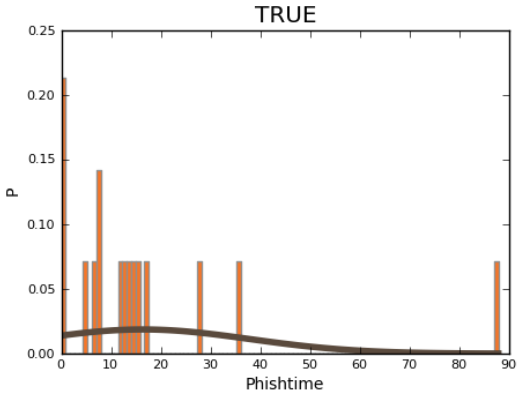


Figure 6-7: Password Given Phishtime Distribution



[15] If p is small, e.g. less than 0.01, or 0.001, you can assume the result is statistically significant i.e. there is a difference.
[16] If the value is small, e.g. less than 0.01, or 0.001, you can assume there is a difference in variance.

[17] Standard Deviation measures the spread of values.
[18] Kurtosis measures the peakedness or flatness of values. Between -1 and 1 is probably great. Between -2 and 2 is probably good.
[19] Skew measures the lopsidedness of values. Between -1 and 1 is probably great. Between -2 and 2 is probably good.
[20] This provides a single measure of normality. If p is small, e.g. less than 0.01, or 0.001, you can assume the distribution is not strictly normal. Note - it may be normal enough though.

As with the ANOVA tests between the phishtime and the operating system (and thus the user desensitisation), there would not appear to be a standard distribution or correlation between the phishtime and whether the password was given or not.

6.9 Data Analysis Conclusion

From the data collected over the course of the experiment, 17% of the users' that had been previously exposed to the keychain password request window fell for the phishing attack, while only 1% of those that did not see the window routinely fell for the attack.

Additionally, the time taken to respond to the phishing attack would appear to bear no relation to whether the user was desensitised or not; nor to whether the user responded to the phshing attack.

As a result of these pretty clear-cut statistics, it can be affirmed that exposure to security pop-ups can desensitize the user to such pop-ups, to the point that the user pays less attention and care to said pop-ups – thus negating, at least partially, the effectiveness of the security messages.

7. Mitigation

No golden bullet exists against phishing. There were automated tools, for example Microsoft's IE8 browser bar, which turned red and put a gateway page whenever an end user tried to access a page that had been reported as fraudulent. Automated tools have come a long way, and are definitely part of the solution, but a dedicated scammer could get around these. A recent response to these tools from the blackhat community relies on JavaScript. By hiding the real "red" URL address bar, and then displaying a fake "green" one at the top of the page, the exploit makes the user think that the user is on a legitimate website.Another failure point of many tools is that they rely on other users to report the site as fraudulent. By the time someone has reported it, who knows how many users have already been scammed!

7.1 Legislative Solutions

Legislation can act as a deterrent; but the scammers are hard to catch. In the fractioned global community that IT security needs to operate in, it is even harder to bring them to justice given the multiplicity, overlaps and gaps in laws between countries. For example, in Argentina "hacking is legal". After a serious breach of a website, a Judge rules that "the (Argentinean) law covers crimes against people, things and animals, but not cyberspace" and as such "hacking into it (the website) cannot be illegal, he said, declaring those on trial innocent of the charges that they broke into the site." On the other hand, in the U.K. not only is breaking into a website deemed illegal (under the computer misuse act amongst others) – but recent changes to the legislation make the possession of many hacking tools illegal (HM Crown 1990).

In a more applied example, in 2001-2002, Gary McKinnon, a system administrator and alleged hacker, broke into almost 100 US military and NASA computers. He was caught in the UK in 2002. That said, at the time of writing, in 2010, Gary McKinnon still hasn't been extradited to the US to face charges due to the complexity of any international case. (McKinnon 2010) (Boyd 2008) (politics.co.uk 2008)

7.2 Technical Solutions

In addition to legislation, new technical measures are developed constantly. Recognisable features are a commonly proposed tactic, as seen in the "Trusted Sender Stamp" system. That said, many of them rely on the user clicking on an image to verify the authenticity... Am I the only person to see the flaw in that? The scammer can simply copy the image and get the user to be directed to a spoof site.

A recent technique, which is growing in popularity in mobile banking systems, is to take a picture of the user when they sign up for the service (i.e. in a physical bank). (Sentinel 2010) When the user subsequently logs in, this picture is presented to them as a "proof" that the site is actually their banks. While a good idea, this can be worked around by a man-in-the-middle type attack.

A man in the middle attack is an attack whereby the attacker relays messages between the server and the client, spoofing to be the other party to whichever party the attacker is responding to. That is to say, the attacker pretends to be the bank to the client, collects the data, then presents the data to the real bank, pretending to be the client. The risk in this attack is that the bank is presented with the real credentials from the client and the client with the real credentials from the bank (though not the bank's security certificate).

For example, The South Florida bank includes "a personal photo on all screen interfaces" (Sentinel 2010). That said, if an attacker were to spoof the bank's website and perform a man-in-the-middle type attack, when the targeted client logged into the attacker's website, the server would then connect to the real bank's server and relay the picture of the user, thus making the user even more confident that the fake site is really the real site. A reasonably technologically savvy user should be able to deduce that the site is fake from the SSL certificate (which will not be the bank's real certificate, or will not match the server). Again though, usability comes into play, and the average non-IT professional user will have trouble verifying the certificate.

One step up from the personal photograph is a computer generated image, which takes as a seed the user's password (as it's typed). This way, as the user types in his/her password, the image changes. Thus, if a user intends to type "password", when a "p" is pressed the image could be blue, then "pa" would be blue with green checks and so on. This uses an algorithm that only the bank has, or combines the user's password with a secret key that only the bank has, to generate the image. As the image should be the same at every log in, the intention is that the user would notice a different image at login. Again, this protection is open to a man-in-the-middle type attack, and the user needs to be vigilant with the SSL certificates.While user education is often targeted as "the solution" to phishing attacks, that "all you need to do is train the users against phishers" – I feel that this isn't an effective solution. Aside from the fact that the amount of technical knowledge required to detect some of the newer, more sophisticated phishing attacks eludes even many IT experts, there is also the issue that from the experiment's results, there was no tangible difference between the knowledgeable group and the novice group in response-rate to the phishing pop-up, if we ignore the operating system they use. This leads me to believe that user education can lead to a dangerous, false sense of security, one where the user feels that the site is secure because the URL "looks right".

7.3 The Human Component

Even though the problem does lie with the users; after all, the problem is caused by the human component, not the digital one – the hardware and software are operating "within operational parameters". The issue is, user apathy is a very hard beast to slay, coupled with the very high technical requirement to detect a sophisticated phishing attack make it a very large problem. It is possible that this is also down to users not giving digital data sufficient value, as the inherent value of data is often ignored.

That said, if instead of looking for a solution to phishing as a whole, one looks at ways to mitigate user desensitisation, a contributing cause to users falling for phishing attacks, there is hope. User apathy to security windows is quite widespread, thus complicating matters. It's not as simple as saying "get Apple to change the keychain password request window". While Microsoft's approach in Vista where no username/password is required for administrative tasks, would work from a password protection point of view (or an approach like banks use online where only parts of the password are requested, never the whole password) this does not stop the user from being desensitized, it merely helps user not give out their password.

At the same time, totally removing all security warning windows is not a solution either; an uninformed user makes uninformed decisions, which can be just as bad. The solution lies in a middle ground – the problem with the "middle ground" is that this can vary from one user to the next so the same frequency of pop-up can't work for all users. That is to say, while for a technical, IT savvy administrator having a pop-up every 5 minutes telling him that his machine was port-scanned is useful (i.e. he can act on this and maybe block the IP address that is scanning him on his firewall) for a non-technical user that doesn't know the meaning of the pop-up, or what to do about it, this would be a problem.

As such, one potential method would be to layer information presented to the user. That is, a first message shows the minimal amount of information. Then, if the user wants to know more, know what to do, or generally needs more information about it the can click-through. Vendors have started implementing this, like Windows' "Learn More" feature in internet explorer security warnings.

That brings another point up – security messages should not just warn a user. While a message telling a user that there is a virus on the machine can be useful to a system administrator – to the end user such a message would be akin to spam. If an antimalware solution has found a virus, there are two possible outcomes – the threat has been mitigated automatically or user interaction is needed to mitigate the threat.

If the threat has been mitigated, then an entry in the log may be enough. This can then be submitted to the administrator as part of a monthly report. If the threat could not be removed, then telling the user that there is a threat without telling them what to do about it isn't very helpful. Assuming the user is also the administrator; the user needs to know what to do next, i.e. how to remove the virus. If the user is not the administrator, the warning should go to the administrator and the system put in a safe state (e.g. an emergency shutdown).

From experience, including time working at a major anti-malware vendor, when an antivirus tells a user that the virus could not be removed; further steps to remove the virus are rarely given. When they are, they are beyond the technical expertise of most users, or are overly alarmist.

Which brings to the next point – overly alarmist security pop-up windows can be the worst culprit in user desensitisation. If a user sees a "WARNING: CRITICAL ERROR" message multiple time a day, only to find that it doesn't require action (possibly because it's a false positive) then when a real critical error shows up the user will treat it like one of the ones he's used to and potentially ignore it. As such, if security messages were to be "rated" into a hierarchy of "criticality" – one which was uniform over all applications a user sees, this would be a positive step forward. Hence, a password-request window would be ranked negligible, and have a green background. Alternatively, a blocked port-scan would be low and have a yellow background and not even display for most users. On the other hand, a worm found on the system, that requires the user to restart the machine to finish removal may be medium risk and marked orange (potentially even forcing the user to reboot). Finally a high risk issue (potentially something that requires the user to turn off their machine immediately and call tech-support, like a hardware keylogger being detected would be red. These ratings are purely an example, a rating system would need to be devised with consultation with hardware and software manufacturers, as well as some key security experts – and is out of the scope of this project.

Another similar mitigation attempt would revolve around reducing the number of security alerts users see to a minimum. That said, some users (e.g. corporate security administrators) would want to see all alerts – no matter how trivial. Hence, some degree of customisation would be ideal – something on the lines of Symantec's Endpoint Protection alert-level display, where a user can have low (only critical alerts displayed), medium (only alerts requiring user action) or high alerts (all alerts displayed). This a definitely a step in the right direction, but would need to be more ingrained in the operating system i.e. instead of seeing an alert telling the user that the form they are using is "submitting data" (as with older versions of IE), this can be "muted" unless a more problematic error needs to be displayed. Sadly – all these changes would need to be made by the developers of software (and potentially hardware); having a third party making these changes would be very difficult.

One final, slightly more technical idea, and one much more targeted at the keychain password request window in particular would be to include a reference to what program. As things stand, the user has no way of knowing what program requested elevation (the user's password) and for what reason. Even a single line that referenced the actual application name or window that requested it would go a long way. The classical "who, what, why, where" still apply. For example, in Apple's privilege escalation case, ideally, the window would have a "more information" section or button, which would inform the user that program XYZ requested elevation at a specific time, how long the elevation will last (expiry of privileges) and have a box with information provided by the program as to the justification that those privileges are required. Suddenly, instead of the user thinking that their operating system is asking for elevation, they would know it's an online survey. With a

bit of education (i.e. don't give your local password to websites) this could help mitigate the issue.

So in a nutshell, while there isn't a simple solution that fixes all problems, a combination of education (both of users and developers) along with a reduction (or at least a greater degree of customisation) in the number of spurious alerts displayed to the end user would be exposed to would be the best possible scenario.

8. Conclusion

Starting with the hypothesis that users that had seen the keychain reminder window would be more prone to giving their credentials away than a user that hadn't due to a desensitizing effect, an experiment was set up to test that. Following a period of testing the experiment, 308 useable responses were collected. From these, the split became apparent; almost 80% of positive responses were from users that had seen the keychain request window before. At the same time, almost 80% of negative responses were from users that had not been regularly exposed to the keychain password request window before. With figures as extreme as these, it's obvious that the familiarity or exposure to the pop-up has been affected the users.

Future work in the area would be quite interesting. Specifically, research that correlates whether there is a relation between preference in operating system (and thus security measures and posture) and the level of IT knowledge and/or skill levels. That is to say, do people with advanced knowledge of IT Security prefer Linux? Or do Apples appeal more to the less technical users? Additionally, what configurations do different skill levels use? Do less technical users stick with the default settings, or do they unlock everything to make it easier to use? Do more technical users choose to lock down the settings that they understand, or do they leave the defaults?

This would be particularly open to debate as there would be a myriad of influencing factors that would lead to both the operating system choice and the security options chosen; like the intended use of the system (e.g. a technical graphics expert may prefer OsX, while an assembly programmer wouldn't be able to survive without Linux), or the security stance may be mandated (e.g. a corporate machine may be locked down so tightly that it impedes the user's ability to work).

Refining the test carried out would also be of particular interest. In particular, carrying out the same test, but customising the elevation popup (i.e. the phishing window), so that there are Windows, Linux and Macintosh flavours.

It could be interpreted that the experiment was "designed to fool apple users", but in essence, the apple users merely served as a desensitized user-base. Had windows had a similar popup window requesting a password (or Linux for that matter), it is probable that those desensitized users would give their password in the same manner on a phishing attack crafted to target them. This would be a particularly interesting test, as it could help increase confidence of the findings.

Another test which would be good would be to re-run the experiment, but target a much larger user base. This could be problematic, as by targeting more people, test subjects may warn each other of the phishing attack, which in turn would skew the results.

Overall though, the results confirmed the hypothesis, and there does appear to be a correlation between user desensitisation and phishing response rates.

Appendix 1: False Survey Explanation

The information provided by you in this questionnaire will be used for research purposes. It will not be used in a manner which allows identification of your individual responses.

Anonymised research data will be archived and used as part of my Dissertation.

- The questionnaire should be completed only by adults aged 18 or over.
- Please read each question carefully and select the most appropriate answer.
- Once you have finished please take a minute to read the consent form.
- The survey should take no longer than 10 minutes to complete.
- If you have any queries about the questionnaire please do not hesitate to contact me using the ContactMe link

- By clicking on the Next Question button below, you confirm that you have read and understand the above information and agree to take part in this study.
- You will have an opportunity to remove your answers at the end
- You understand that your participation is voluntary and that you are free to withdraw at any time, without giving any reason, or legal rights being affected
- You understand that relevant questions, answers and data collected during the study (in anonymised form), may be looked at by responsible individuals as part of this study, where it is relevant to your taking part in this research. You give permission for these individuals to have access to this data.

Instructions:

- Please read each question carefully and select the most appropriate answer.
- In most cases, you will be presented with a set of buttons ( ); please select ONE answer
- You may also be presented with an open-ended answer ( ); please type the most appropriate answer
- At the bottom of each page there are 3 buttons
- Next Question submits the answer and moves you onto the next question
- Reset Question allows you to reset the answers for that particular question to all blank

Appendix 2 : Consent Form and Debriefing

Thank you for taking the time to complete the survey.

Research designs often require that the full intent of the study not be explained prior to participation. Although we have described the general nature of the tasks that you have been asked to perform, the full intent of the study was not explained in full. The survey included a research component on the user (your) interactions with security.

This survey is part of a research on User Desensitization, not about operating systems. That is to say, the use of recurrent security popup messages shown to a user. The hypothesis is that by presenting a user with an identical popup very frequently can have an adverse effect on the effectiveness of said popup. Hence, halfway through the survey you were presented with a popup similar to Apple OsX's Keychain password request window. The way you reacted to this window (time, data provided and buttons pressed) will then be used in the research.

Partial disclosure was required as to not affect your reaction to the security popup. Your username and password have NOT been saved and have already been removed from our system.

With the results of this study, I hope to help further usability in the field of IT security. Your responses are a valuable contribution to this. If you would like to receive a report of this study (i.e. a copy of my thesis) when it is completed, please use the Contact Me form and tick the "keep me updated" box.

If you have any questions about the study, about the limited disclosure involved or about the methods used, please feel free to ask the principal investigator using the Contact form. If you have concerns about this study or your rights as a participant in this study, you may also use the said form.

If you wish to have all data collected and answers removed from the study during this session, click the Withdraw from Survey button now. If you consent to your responses

Appendix 3 : Survey Questions & Structure

1. What is your age?
    a. Between 18-25
    b. Between 26-35
    c. Between 36-45
    d. Between 46-55
    e. Between 56-65
    f. Between 66-75
    g. Between 76-85
    h. Over 86
    i. Prefer Not Answer
2. What is your sex?
    a. Female
    b. Male
    c. Prefer Not Answer
3. Please indicate the highest level of formal education attained?
    a. No High School
    b. Some High School
    c. High School Graduate
    d. Some College/University
    e. College Graduate
    f. Postgraduate Degree
    g. Prefer Not Answer
4. How many children do you have?
    a. 0
    b. 1
    c. 2
    d. 3
    e. 4
    f. 5 or more
    g. Prefer Not Answer
5. How would you describe your computer comfort level (IT literacy level)?
    a. IT Native
    b. IT Expert
    c. Intermediate
    d. IT Novice
    e. Prefer Not Answer
6. What Operating System do you predominantly use?
    a. Apple OsX

b. Microsoft Windows
c. Linux
d. Other
e. Don't know
f. Prefer Not Answer

7. How long have you been using (OsX, Windows, Linux, Other) as your main operating system?
    a. Less than a month
    b. 1-6 Months
    c. 6-12 Months
    d. Over a year
    e. Prefer Not Answer
    f. Don't know
8. How would you access your home folder?
    <Show Phishing Popup Window Overlay>
    a. Click on Go>Home
    b. Click on Start>{username}
    c. Type cd ~
    d. CMD+Shift+H
    e. Other
    f. Don't know
    g. Prefer Not Answer
9. How would you access your home folder?
    a. Click on Go>Home
    b. Click on Start>{username}
    c. Type cd ~
    d. CMD+Shift+H
    e. Other
    f. Don't know
    g. Prefer Not Answer
10. How many times do you shop online?
    a. Never
    b. Monthly
    c. Weekly
    d. Daily
    e. Don't know
    f. Prefer Not Answer
11. What is your most visited website?
    a. <Open Ended Answer (text box)>
12. How many spam emails do you receive per day?

a. None
b. 1-5
c. 6-10
d. 10+
e. Don't know
f. Prefer Not Answer

13. How often do you see keychain reminders?
    a. Never
    b. Monthly
    c. Weekly
    d. Daily
    e. Don't know
    f. Prefer Not Answer
14. Does your password begin with <password from phish> ?
    a. Yes
    b. No
    c. Don’t Know
    d. Prefer Not Answer

Appendix 4 : Raw Data & Source Code (CD-Rom)

Appendix Folder Structure:

- Data
  - Raw Data.xls
    - Contains the raw data as gathered
  - Interim Data.xls
    - Contains the data in a human readable format
- Source Code
  - Original Survey
    - Contains the source code for the survey as presented to respondents
  - Closed Survey
    - Contains the source code for the survey after the survey closed

Bibliography

Addiction Science Research and Education Center, University of Texas. “Dopamine - A Sample Neurotransmitter.” *University of Texas.* 2009. http://www.utexas.edu/research/asrec/dopamine.html.

Anderson, R. *Security Engineering.* Danvers, MA: Wiley, 2001.

APWG. “Second Half 2008 Report.” *APWG.* 12 2008. http://www.antiphishing.org/reports/apwg_report_H2_2008.pdf.

Boyd, Clark. *Profile: Gary McKinnon .* 30 07 2008. http://news.bbc.co.uk/1/hi/technology/4715612.stm.

Carnagey, Nicholas. “ISU psychologists produce first study on violence desensitization from video games.” *Iowa State University.* 24 07 2006. http://www.public.iastate.edu/~nscentral/news/06/jul/desensitized.shtml.

Cheng, Jacqui. “Hundreds of MobileMe customers caught in phishing net.” *Ars Technica.* 15 08 2009. http://arstechnica.com/apple/news/2008/08/hundreds-of-mobileme-customers-caught-in-phishing-net.ars.

Cherecwich, Rich. “The Last Hope for Online Advertising.” *Imedia Connection.* 09 02 2009. http://www.imediaconnection.com/summits/coverage/21967.asp.

Connolly, P.M. Connolly and K.J. *Employee Surveys.* Old Saybrook, CT: Performance Programs, Inc, 2006.

Consumer Reports. “State of the net.” *http://www.consumerreports.org.* 12 2007. http://www.consumerreports.org/cro/electronics-computers/computers-internet/internet-and-other-services/net-threats-9-07/state-of-the-net/0709_state_net.htm.

Consumer.Gov. “Top 10 Online Fraud Methods.” *Consumer.Gov.* http://www.consumer.gov/sentinel/pubs/Top10Fraud2003.pdf.

Couper, M. P. *Designing Effective Web Surveys.* Cambridge: Cambridge University Press, 2008.

Creswell, J. W. *Research Design.* Thousand Oaks, CA: Sage, 2009.

D. Krane, S. Ford, J. Gilder, K. Inman, A. Jamieson, R. Koppl, I. Kornfield, D. Risinger, N. Rudin, M. Taylor, W.C. Thompson. “A means of minimizing observer effects in forensic DNA interpretation.” *Journal of Forensic Sciences*, 2008: 53(4):1006-7.

Dal Conte, Dr. Nadia, interview by Michele Daryanani. *Internet Security* (08 06 2010).

Dorothy, C. Albert and A. *Managing Information Security Risks.* Boston: Addison Wesley, 2003.

DR-K. *The Complete H@ckers Handbook.* London: Carlton, 2000.

Dupre, L. *Bugs in Writing.* Hamilton, NY: Addison Wesley, 1998.

Fear Free Flying. "Systematic Desensitisation." *Feer Free Flying.* http://www.fearfreeflying.co.uk/SystematicDesensitisation.html.

Fielder, L.B. Bourque and E. P. *How to Conduct Self-Adminsitered and Mail Surveys 2nd ed.* Thousand Oaks, CA: Sage, 2003.

Fink, A. *How to ask survey questions.* Thousand Oaks, CA: SAGE Publications, 1995.

FireHaus Network. "iTunes/Apple Store E-mail Phishing Scam." *FireHaus Network.* 29 03 2010. http://www.firehaus.net/2010/03/29/itunesapple-store-e-mail-phishing-scam/.

Garfinkel, F. Cranor and S. *Security and Useability.* Sebastopol, CA: O'Rilley, 2005.

Goodin, Dan. "Apple faithful snared in phishing scam targeting Mac.com users." *The Register.* 13 08 2008. http://www.theregister.co.uk/2008/08/13/phishers_attack_mac_faithful/.

Grimes, R. "Is Windows Vista's user security elevation better than Mac OS X's?" *InfoWorld: Security Central.* 05 03 2006. http://www.infoworld.com/d/security-central/windows-vistas-user-security-elevation-better-mac-os-xs-196 (accessed 04 22, 2009).

HM Crown. "Computer Misuse Act 1990." *Legislation.gov.uk.* 1990. http://www.legislation.gov.uk/ukpga/1990/18/contents (accessed 2010).

Kuljis, A. J. DeWitt and J. "Aligning Usability and Security: A Usability Study of Polaris." Uxbridge: Brunel University, 2006.

L. Cranor, Y. Zhang, S. Egelman and J. Hong. "Phinding Phish: Evaluating Anti-Phishing Tools." San Diego, CA: 14th Annual Network & Distributed System Security Symposium, 2006.

Lee, Martin. *The Security of Password Reset Questions.* Oxford: Oxford Dissertation, 2008.

Lorrie Faith Cranor, Simson Garfinkel. "Secure or Useable?" (IEEE) 2, no. 5 (2004).

M.A. Sasse, R.J. Cunningham, and R.L. Winder. *People and Computers XI, Proceedings of HCI '96.* London: Springer, 1996.

MailOnline. “Violent video games 'desensitise' players.” *MailOnline.* 17 07 2006. http://www.dailymail.co.uk/news/article-401113/Violent-video-games-desensitise-players.html.

Markus Jakobsson, Alex Tsow, Ankur Shah, Eli Blevis, and Youn-Kyung Lim. “Useable Security.” *What Instills Trust? A Qualitative Study of Phishing.* http://usablesecurity.org/papers/jakobsson.pdf (accessed 04 21, 2009).

McKinnon, Gary. *Free Gary McKinnon.* 11 09 2010. http://freegary.org.uk/.

McLean, Prince. “New phishing scam targets MobileMe users.” *Apple Insider.* 29 02 2009. http://www.appleinsider.com/articles/09/02/26/new_phishing_scam_targets_mobileme_users.html.

Miles, Millers. “Scam Report.” *Millers Miles.* 01 04 2010. http://www.millersmiles.co.uk/report/18553.

Millers Miles. “HSBC Important Security Verification.” *MillersMiles.co.uk.* 25 04 2010. http://www.millersmiles.co.uk/report/18862.

Mitnick, K. D. *The Art of Deception.* Danvers, MA: Wiley, 2002.

Nicholas L. Carnageya, Craig A. Andersonb and Brad J. Bushmanc. “The effect of video game violence on physiological desensitization to real-life violence.” 2005.

Oxford Information Services Ltd. “Millers Miles.” *Millers Miles.* 04 2010. http://www.millersmiles.co.uk/search.php.

P. Kumaraguru, S. Sheng, R Acquisti, L. Faith Cranor, J. Hong. *Teaching Johnny Not to Fall for Phish.* Pittsburgh: Carnegie Mellon University, 2007.

P. Sebranek, V. Meyer, and D. Kemper. *Write for College.* Wilmington, MA: Writesource, 1997.

politics.co.uk. *British hacker loses extradition appeal .* 28 08 2008. http://www.politics.co.uk/news/legal-and-constitutional/british-hacker-loses-extradition-appeal-$1238262.htm.

R. Kohavi, R. M. Henne and D. Sommerfield. “Practical Guide to Controlled Experiments on the Web.” San Jose: Microsoft, KDD 2007, 2007.

Race, B. Allison and P. *The Student's Guide to Preparing Dissertations and Thesis.* London: Routledge-Falmer, 2004.

Ritter, V.M. Sue and L.A. *Conducting Online Surveys.* Thousand Oaks, CA: Sage, 2007.

Ronny Kohavi, Thomas Crook, and Roger Longbotham. “Online Experimentation at Microsoft.” *Microsoft's Experimentation Platform.* 09 09 2009. http://exp-platform.com/expMicrosoft.aspx.

Schneier, B. *Secrets and Lies.* Danvers, MA: Wiley, 2000.

Sentinel, Doreen Hemlock Sun. *Mobile Banking Takes Off, But Some Customers Worry About Security.* 15 09 2010. http://www.allbusiness.com/banking-finance/banking-finance-sector-performance/15087514-1.html.

Sophos. “Phishing and the threat to corporate networks.” *Sophos.* 08 2005. http://www.sophos.com/whitepapers/sophos-phishing-wpuk.pdf.

—. “Simple Steps to Avoid Being Phished.” *Sophos.* 2009. http://www.sophos.com/security/best-practice/phishing.html.

“'Spear Phishing' Increasing.” *redOrbit Technology News.* 7 12 2008. http://www.redorbit.com/news/technology/1612771/spear_phishing_increasing/index.html?source=r_technology.

Swetnam, D. *Writing your Dissertation.* Oxford: HowToBooks, 2004.

Symantec. “Symantec Global Internet Security Threat Report.” *Symantec.* 04 2010. http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_internet_security_threat_report_xv_04-2010.en-us.pdf.

—. *Symantec Intelligence Quarterly.* 04 2010. http://www.symantec.com/business/theme.jsp?themeid=threatreport.

—. “Symantec Internet Security Threat Report Execurity Summary.” *Symantec.com.* 04 2010. http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_exec_summary_internet_security_threat_report_xv_04-2010.en-us.pdf.

Trusteer. “Measuring the Effectiveness of In-the-Wild Phishing Attacks.” *Trusteer.* 02 12 2009. http://www.trusteer.com/sites/default/files/Phishing-Statistics-Dec-2009-FIN.pdf.

—. *Trusteer.* 04 2010. http://www.trusteer.com/research-0 (accessed 04 21, 2010).

Vertster. *How does Split Testing Work.* Vertster. 2009. http://www.vertster.com/howitworks (accessed 04 21, 2009).

www.data-archive.ac.uk. “Sample consent statement for a survey or questionnaire.” *UK Data Archive.* 24 06 2008. http://www.data-archive.ac.uk/news/eventsdocs/sample24jun08.doc.

Zobel, J. *Writing for Computer Science.* London: Springer, 2004.