Designing Ethical Phishing Experiments: A study of (ROT13) rOnl query features

Markus Jakobsson
School of Informatics
markus@indiana.edu

Jacob Ratkiewicz
Dept. of Computer Science
jpr@indiana.edu

Indiana University
Bloomington, IN 47406, USA

ABSTRACT

We study how to design experiments to measure the success rates of phishing attacks that are *ethical* and *accurate*, which are two requirements of contradictory forces. Namely, an *ethical* experiment must not expose the participants to any risk; it should be possible to locally verify by the participants or representatives thereof that this was the case. At the same time, an experiment is *accurate* if it is possible to argue why its success rate is not an upper or lower bound of that of a real attack – this may be difficult if the ethics considerations make the user perception of the *experiment* different from the user perception of the *attack*. We introduce several experimental techniques allowing us to achieve a balance between these two requirements, and demonstrate how to apply these, using a context aware phishing experiment on a popular online auction site which we call "rOnl". Our experiments exhibit a measured average yield of 11% per collection of unique users. This study was authorized by the Human Subjects Committee at Indiana University (Study #05-10306).

Categories and Subject Descriptors

K.4.4 [**Electronic Commerce**]: Security; K.4.1 [**Public Policy**]**:** Ethics;

General Terms

Experimentation, Security, Human Factors, Legal Aspects

Keywords

Accurate, Ethical, Experiment, Phishing, Security

1. INTRODUCTION

While it is of importance to understand what makes phishing attacks successful, there is to date very little work done in this area. Dominating the efforts are surveys, such as those performed by the Gartner Group in 2004 [6]; these studies put a cost of phishing attacks around $2.4 billion per year in the US alone, and report that around 5% of adult American Internet users are successfully targeted by phishing attacks each year. (Here, a successful phishing attack is one which persuades a user to release sensitive personal or financial information, such as login credentials or credit card numbers). However, we believe that this is a lower bound: the statistics may severely underestimate the real costs and number of victims, both due to the stigma associated with being tricked (causing people to under-report such events), and due to the fact that many victims may not be aware yet of the fact that they were successfully targeted. It is even conceivable that this estimate is an upper bound on the true success rate of phishing attacks, as some users may not understand what enables a phisher to gain access to their confidential information (e.g. they may believe that a phisher can compromise their identity simply by sending them a phishing email).

Mailfrontier [1] released in March '05 a report claiming (among other things) that people identified phishing emails correctly 83% of the time, and legitimate emails 52% of the time. Their conclusion is that when in doubt, people assume an email is fake. We believe that this conclusion is wrong — their study only shows that when users know they are being tested on their ability to identify a phishing email, they are suspicious.

A second technique of assessing the success rates is by monitoring of ongoing attacks, for instance, by monitoring honeypots. The recent efforts by The Honeypot Project [2] suggest that this approach may be very promising; however, it comes at a price: either the administrators of the honeypot elect to not interfere with an attack in progress (which may put them in a morally difficult situation, as more users may be victimized by their refusal to act) *or* they opt to protect users, thereby risking detection of the honeypot effort by the phisher, and in turn *affecting* the phenomenon they try to measure — again, causing a lower estimate of the real numbers.

A third and final approach is to perform experiments on real user populations. The main drawback of this is clearly that the experiments have to be ethical, i.e., not harm the participants. Unless particular care is taken, this restriction may make the experiment sufficiently different from reality that its findings do not properly represent reality or give appropriate predictive power. We are aware of only two studies of this type. The first study, by Garfinkel and Miller [3] indicates the (high) degree to which users are willing to ignore the presence or absence of the SSL lock icon when making a security-related decision; and how the name and context of the sender of an email in many cases matter more (to a recipient determining its validity) than the email address of

WWW 2006, May 23–26, 2006, Edinburgh, Scotland.
ACM 1-59593-323-9/06/0005.

the sender. While not immediately quantified in the context of phishing attacks, this gives indications that the current user interface may not communicate phishy behavior well to users. A second experimental study of relevance is that performed by Jagatic et al. [8]), in which a social network was used for extracting information about social relationships, after which users were sent email appearing to come from a close friend of theirs. This study showed that more than 80% of recipients followed a URL pointer that they believed a friend sent them, and over 70% of the recipients continued to enter credentials at the corresponding site. This is a strong indication of the relevance and effectiveness of context in phishing attacks. However, the study also showed that 15% of the users in a control group entered valid credentials on the site they were pointed to by an unknown (and fictitious) person within the same domain as themselves. This can be interpreted in two ways: either the similarity in domain of the apparent sender gave these user confidence that the site would be safe to visit, or the numbers by Gartner are severe underestimates of reality.

We believe it is important not only to assess the danger of *existing* types of phishing attacks, as can be done by all the three techniques described above, but also of *not yet* existing types — e.g., various types of context-aware [4] attacks. We are of the opinion that one can only assess the risk of attacks that do not yet exist in the wild by performing experiments. Moreover, we do not think it is possible to argue about the exact benefits of various countermeasures without actually performing studies of them. This, again, comes down to the need to be able to perform experiments. These need to be *ethical* as well as *accurate* — a very difficult balance to strike, as deviating from an actual attack that one wishes to study in order not to abuse the subjects may introduce a bias in the measurements. Further complicating the above dilemma, the participants in the studies need to be unaware of the existence of the study, or at the very least, of their own participation — at least until the study has completed. Otherwise, they might be placed at heightened awareness and respond differently than they would normally, which would also bias the experiment.

In this study, we make an effort to develop an ethical experiment to measure the success rate of one particular type of attack. Namely, we design and perform an experiment to determine the success rates of a particular type of "content injection" attack. (A content injection attack is one which works by inserting malicious content in an otherwise-innocuous communication that appears to come from a trusted sender). As a vehicle for performing our study we use a popular online auction site which we call rOnl (and pronounce "ROW-null")[1]. We base our study on the current rOnl user interface, but want to emphasize that the results generalize to many other types of online transactions. Features of the rOnl communication system make possible our ethical construction (as will be discussed later); this construction is orthogonal with the success rate of the actual attack. Our work is therefore contributing both to the area of designing phishing experiments, and the more pragmatic area of assessing risks of various forms of online behavior.

[1]This name is of course an obfuscation of the real name of our subject site; this was done at the advice of our legal representative.

Overview of Paper. The next few sections (§ 2, § 3) introduce in detail some phishing attacks that may take place in the context of user-to-user communication. In particular, we describe several scenarios involving the specific phishing attacks that we would like to study. We then describe our experiment in § 4, and show that at while it is ethical and safe to perform, it simulates a real phishing attack.

Finally, we outline the implementation of the experiment in section § 5. We discuss findings in § 6, including the interesting conclusion that *each attack* will have a 11% success rate, and that users ignore the presence (or absence) of their username in a message (which rOnl uses to certify that a message is genuine).

Overview of Techniques. The following are some of the major techniques we develop and use in the process of crafting our experiment. We hope that their description will prove useful to others performing similar studies.

- Obfuscation of valid material, making the material (such as URLs) appear phishy. Thus, users who would have spotted a corresponding phishing attack will reject this (valid) URL, but naïve users who would have fallen victim to a real attack will accept it. This is covered in section § 4.2.
- Use of query forwarding and spoofing to simulate content injection. We do this by forwarding modified rOnl queries using spoofing, making them appear as though they still come from rOnl. This technique, combined with the one above, allows us to measure the success of the simulated attack without gaining access to credentials. Section § 4.1.
- Use of degradation of context information to mimic the lack of, or incomplete, context information in potential phishing attacks. For instance, this might be accomplished by leaving a subject's name out of an query (when a legitimate query would include it). This allows us to measure the degree to which these clues are observed by the recipient. This is discussed in the description of our experiments in section § 4, particularly Experiments 2 and 4.
- Creation of control groups that receive unaffected material — for example, unmodified rOnl queries containing valid links. This is performed by spoofing of emails to get the same risk of capture by spam filters as material that is degraded to signal a phishing attack. This too is described in § 4, as Experiments 1 and 2.

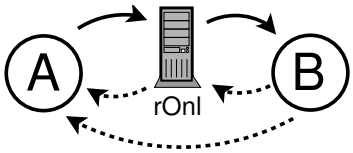
2. USER-TO-USER PHISHING ON RONL

We contrast a user-to-user phishing attempt with an attempt that is purported to come from an authority figure, such as rOnl itself. Before discussing what types of user-to-user phishing attacks are possible, it is useful to describe the facilities rOnl provides for its users to communicate.

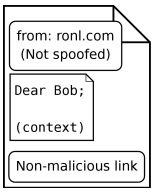
2.1 rOnl User-to-User Communication

rOnl enables user-to-user communication through an internal messaging system similar to internet email. Messages are delivered both to the recipient's email account, and their rOnl message box (similar to an email inbox, but can only receive messages from other rOnl users). The sender of a

message has the option to reveal her email address. If she does, the recipient (Bob in Figure 1) may simply press 'Reply' in his email client to reply to the message (though doing this will reveal his email address as well). He may also reply through rOnl's internal message system, which does not reveal his email address unless he chooses. See Figure 1 for an illustration of this scenario.



(a) Communication path



(b) Features of email

Figure 1: Normal use of the rOnl message system. In (a), Alice sends a message to Bob through rOnl. If she chooses to reveal her email address, Bob has the option to respond directly to Alice through email; in either case, he can also respond through the rOnl message system. If Bob responds through email, he reveals his email address; he also has the option to reveal it while responding through rOnl. The option to reveal one's email address implies that this system could be exploited to harvest email addresses, as will be discussed later. Figure (b) illustrates the features of an email in a normal-use scenario. This will be contrasted to the features of various types of attacks (and attack simulations) which will arise later.

In messages sent to a user's email account by the rOnl message system, a 'Reply Now' button is included. When the user clicks this button, they are taken to their rOnl messages to compose a reply (they must first log in to rOnl). The associated reply is sent through the rOnl message system rather than regular email, and thus need not contain the user's email address when it is being composed. Rather, rOnl acts as a message forwarding proxy between the two communicating users, enabling each user to conceal their internet email address if they choose. An interesting artifact of this feature is that the reply to a message need not come from its original recipient; the recipient may forward it to a third party, who may then click the link in the message, log in, and answer. That is, a message sent through rOnl to an email account contains what is essentially a `reply-to` address encoded in its 'Reply Now' button — and rOnl does not check that the answer to a question comes from its original recipient. This feature will be important in the design of our experiment.

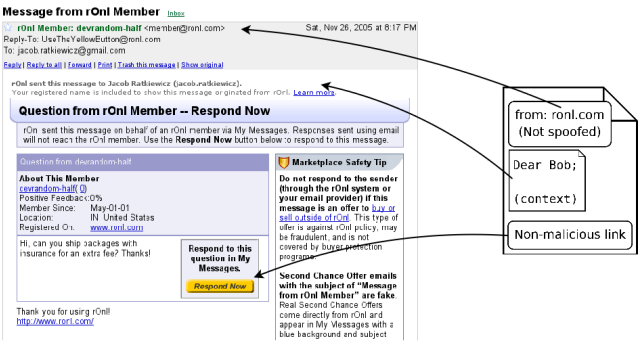


Figure 2: A message forwarded to an email account from the rOnl message system. Note the headers, greeting (including real name and username), and "Reply Now" button.

2.2 Abusing User-to-User Communication

A user-to-user phishing attempt would typically contain some type of deceptive question or request, as well as a malicious link that the user is encouraged to click. Since rOnl does not publish the actual internet email addresses of its users on their profiles, it is in general non-trivial to determine the email address of a given rOnl user. This means that a phisher wishing to attack an rOnl user in a context-aware way must do one of the following:

1. Send the attack message through the rOnl messaging system. This does not require the phisher to know the potential victim's internet email address, but limits the content of the message that may be sent. This is a type of *content injection* attack — the malicious information is inserted in an otherwise-innocuous message that really does come from rOnl.

2. Determine a particular rOnl user's email address, and send a message to that user's internet email. Disguise the message to make it appear as though it was sent through the rOnl internal message. This is a *spoofing* attack.

3. Spam users with messages appearing to have been sent via the rOnl message system, without regard to what their rOnl user identity is[2]. This may also use spoofing, but does not require knowledge of pairs of rOnl user names and email addresses; the resulting message will therefore not be of the proper form in that the user name cannot be included. Since rOnl tells its users that the presence of their username in a message is evidence that the message is genuine, this may make users more likely to reject the message.

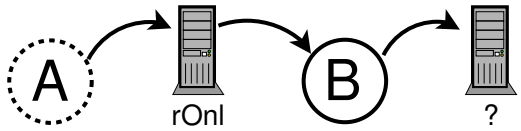
A more detailed discussion of each of these types of attacks follows.

Content Injection Attacks. rOnl's implementation of its internal message system makes content injection attacks impossible, but many sites have not yet taken this precaution.

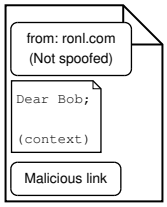
[2]We note that an attacker may still choose only to target actual rOnl users if he first manages to extract information from the users' browser caches indicating that they use rOnl. See [7] for details on such browser attacks.

A content injection attack against an rOnl user would proceed as follows. The phisher (Alice in Figure 3(a)) composes a question to the victim (Bob) using rOnl's message sending interface. Alice inserts some HTML code representing a malicious link in the question (Figure 3(b)) — this is what makes this attack a content injection attack. Since rOnl also includes HTML in their messages in order to use different fonts and graphics, the malicious link may be carefully constructed to blend in with the message.

This attack is particularly dangerous because the malicious email in fact does come from a trusted party (rOnl in this case), and thus generally will not be stopped by automatic spam filters. This attack is easy to prevent, however; rOnl could simply not allow users to enter HTML into their question interface. When a question is submitted, the text could be scanned for HTML tags, and rejected if it contained any. Doing so would prevent phishers from using the rOnl interface to create questions with malicious links. This is in fact what rOnl has implemented; thus an attack of this type is not possible. Figure 3 illustrates a content injection attack.



(a) Communication path



(b) Features of email. Important context information includes Bob's rOnl username.

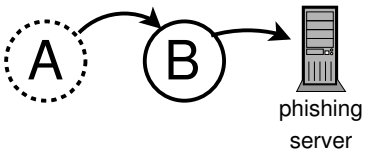
Figure 3: Content injection attack. The malicious communication originates from an rOnl server, but its link leads to a third-party site. Figure (b) shows the features of this email; note that the email would also contain a non-malicious link (since the malicious link is inserted, it does not replace the normal contents of the message).

Email Spoofing. Another common phishing tool is email spoofing. Spoofing an email refers to the act of forging an email message to make it appear to be from a sender it is not. Spoofing is remarkably easy to accomplish. The most widely used protocol for transmitting Internet email is SMTP, the Simple Mail Transfer Protocol. An email is typically relayed through several SMTP servers, each hop bringing it closer to its destination, before finally arriving. Because SMTP servers lack authentication, a malicious user may connect to an SMTP server and issue commands to transmit a message with an arbitrary sender and recipient. To the SMTP server, the malicious user may appear no different than another mail server transmitting a legitimate email message intended for relay.

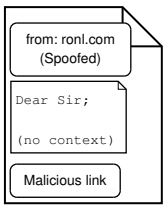
Spoofed emails can be identified by a close inspection of the header of the email (which contains, among other things, a list of all the mail servers that handled the email and the times at which they did so). For instance, if the true mail server of the supposed sender does not appear in the list of servers which handled the message, the message cannot be legitimate. In many cases, this inspection can be done automatically. This is important, for it implies that spoofed emails can frequently be caught and discarded by automatic spam filters.

Spoofing and Phishing. In a spoofing phishing attack, a phisher may forge an email from a trustworthy party, containing one or more malicious links. A common attack is an email from "rOnl" claiming that the user's account information is out of date and must be updated. When the user clicks the link to update his or her account information, they are taken to the phisher's site.

Since a spoofed message is created entirely by the phisher, the spoofed message can be made to look exactly like a message created by content injection. However, the spoofed message will still bear the marks of having been spoofed in its headers, which makes it more susceptible to detection by spam filters. Figure 4 illustrates a spoofing attack.



(a) Communication path – A sends a message to B which appears to come from the trusted site.



(b) Features of email. Often spoofing attacks contain no context information and are sent to many users.

Figure 4: A spoofing attack. Note that the legitimate site is never involved, in contrast to a content injection attack. A spoofing attack may include contextual information about its recipient, but current attacks in the wild usually do not. Important to note in Figure (b) is that the message only pretends to come from a trusted site, unlike messages in content injection attacks.

A spoofed message may also simulate a user-to-user communication. Spoofing used in this manner can not be used to deliver the phishing attempt to the user's internal rOnl message inbox — only a content injection attack could do that. It can only deliver the message to the user's standard email account. If a user does not check to ensure that the spoofed message appears in both inboxes, however, this shortcoming does not matter.

Since a spoofing attack must target a particular email address, including context information about an rOnl user would require knowing the correspondence between an rOnl username and an email account. This is in general non-trivial to acquire.

Context-Aware Attacks. A context-aware [4] phishing attack is one in which the phisher obtains some contextual information about the victim's situation, and uses it to make the attack more believable. We believe that in general context-aware attacks pose a higher risk to users, because they may believe that no one but a trusted party would have access to the personal information included in such an attack. There are several ways that publicly available information on rOnl can be used to construct context-aware attacks.

- *Purchase History:* rOnl includes a reputation system that allows buyers and sellers to rate each other once a transaction is completed. Each rating includes the item number of the item involved, and each user's collection of ratings is public information.

  A phisher could mine this information to determine which items a particular user has bought and sold, and could then use this information to make her attack more believable to a potential victim.

- *Username / Email Correspondence:* Even though rOnl attempts to preserve the anonymity of its users by hiding their email addresses and acting as a proxy in email communications, a phisher can still determine the correspondence between email addresses and usernames. One of the simplest methods is for the phisher to send some message through rOnl to each of a number of different usernames, choosing to reveal her own email address. A previous study [4] has suggested that 50% of users will reply directly to the message (i.e., by email) instead of replying through rOnl's message system or not at all. Thus, about half of the users so contacted will reveal their own email addresses. These numbers are supported by our study, in which we obtain an approximate "direct" response rate of 47%.

  rOnl places a limit on the number of messages any user may send through its interface; this limit is based on several factors, including the age of the user's account and the amount of feedback the user has received. (A post on the rOnl message boards stated that the allowed number never exceeded 10 messages in a 24-hour period; however, we have been able to send more than twice this many in experiments.) In any event, there are several ways that a phisher might circumvent this restriction:

  – the phisher could send messages from the account of every user whose credentials he gained (though previous attacks), or

  – the phisher could continue to register and curry new accounts, to be used for the express purpose of sending messages.

  Of course, each phishing attack the phisher sends may cause a user to compromise their account, with a given probability; and with each compromised account, the phisher may send more messages. Figure 5 shows the number of messages a phisher may send grows exponentially, with exponent determined by the success rate of the attack. In the figure, a unit of time is the time it takes the phisher to send a number of messages comprising phishing attacks from all the accounts he owns, gain control of any compromised accounts, and add these compromised accounts to his collection (assuming that this time is constant no matter how many accounts the phisher has). For a given success rate s, and assuming that an account may send a number of messages c on average, the number of messages a phisher may send after t time steps is given by the exponential function $m(t) = \lfloor c \cdot (1+s)^t \rfloor$, which is what is plotted for the given values of s.

  This type of context information — a pairing between a login name for a particular site, and a user's email address — is called *identity linkage* [4]. In rOnl's case this linkage is especially powerful, as rOnl tells its users that the presence of their username in an email to them is evidence that the email is genuine.

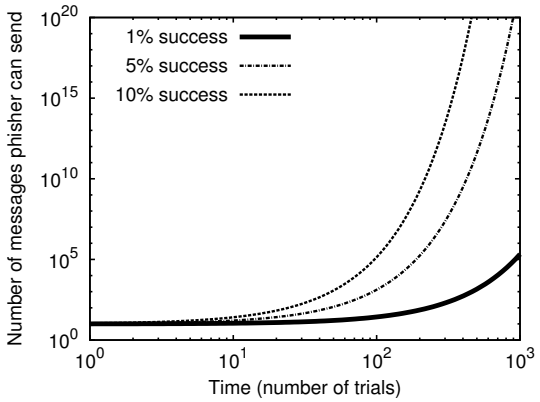


Figure 5: The number of messages a phisher may send at any time is an exponential function of the success rate of their attack; each new compromised account means gives them the ability to send more messages, thus potentially compromising more accounts. Here we show the growth in the number of messages that may be sent assuming a success rate of 1%, 5%, or 10%. As may be seen, this number quickly becomes very large, even for small success rates.

It should be noted that "context-aware" refers to the presence of certain meaningful information in the attack, not to the mechanism by which the attack is performed.

3. RONL PHISHING SCENARIOS

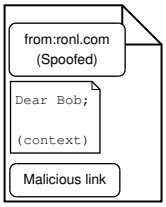
In considering the phishing attempts we discuss, it is useful to contrast them with the normal use scenario:

- *Normal use* – Alice sends a message to Bob through rOnl, Bob answers. If Bob does not reply directly through email, he must supply his credentials to rOnl in order to answer. This situation occurs regularly in typical rOnl use, and corresponds to Figure 1). Important for later is the fact that when a user logs in to answer a question through the rOnl message system, he is reminded of the original text of the question by the rOnl web site.

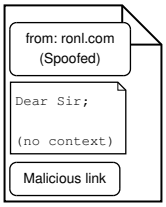
The following are some scenarios that involve the rOnl messaging interface. In each, a user (or phisher) Alice asks another user (or potential victim) Bob a question. In order to answer Alice's question, Bob must click a link in the email sent by Alice; if Bob clicks a link in an email that is actually a phishing attack, his identity may be compromised.

- *Attack 1: Context-aware spoofing attack* – Alice spoofs a message to Bob, bypassing rOnl. If Bob chooses to respond by clicking the link in the message (which Alice has crafted to look exactly like the link in a genuine message), he must supply his credentials to a potentially malicious site. Alice controls the contents of the email, and thus may choose to have the link direct Bob to her own website, which may harvest Bob's credentials. Alice includes contextual information in her comment to make Bob more likely to respond. This corresponds to Figure 6(a).

- *Attack 2: Contextless spoofing attack* – This is a spoofing attack in which Alice makes certain mistakes - perhaps including incorrect context information, or no information at all. This corresponds to an attack in which a phisher does not attempt to determine associations between rOnl user names and email addresses, but simply sends spoofed emails to addresses he has access to, hoping the corresponding users will respond. The degree to which Bob is less likely to click a link in a message that is part of this attack (with respect to a message in the context-aware attack above) measures the impact that contextual information has on Bob, which is an important variable we wish to measure. This corresponds to Figure 6(b).



(a) Attack 1 – Includes context information



(b) Attack 2 – Incorrect, or missing, context information

Figure 6: Two possible spoofing attacks. Attacks currently in the wild, at the time of writing, are closer to (b), as they do not often contain contextual information.

4. EXPERIMENT DESIGN

In our study we wished to determine the success rates of Attacks 1 and 2 as described in the previous section, but we cannot ethically or legally perform either — indeed, performing one of these attacks would make us vulnerable to lawsuits from rOnl, and rightly so. Thus one of our goals must be to develop an experiment whose success rate is strongly correlated with the success rate of Attacks 1 and 2, but which we can perform without risk to our subjects (or ourselves).

To this end we must carefully consider the features of the attacks above that make them different from a normal, innocuous message (from the recipient's point of view):

1. Spoofing is used (and hence, a message constituting one of these attacks may be caught by a spam filter).

2. An attack message contains a malicious link rather than a link to `rOnl.com`.

More carefully restated, our goals are as follows: we wish to create an experiment in which we send a message with both of the above characteristics to our experimental subjects. This message must thus look exactly like a phishing attack, and must ask for the type of information that a phishing attack would (login credentials). We want to make sure that while we have a way of knowing that the credentials are correct, we never have access to them. We believe that a well-constructed phishing experiment will not give researchers access to credentials, because this makes it possible to prove to subjects after the fact that their identities were not compromised[3].

Let us consider how we may simulate each of the features in a phishing attack — spoofing and a malicious link — while maintaining the ability to tell if the recipient was willing to enter his or her credentials.

4.1 Experimenting with Spoofing

The difficulty in simulating this feature is not the spoofing itself, as spoofing is not necessarily unethical. Rather, the challenge is to make it possible for us to receive a response from our subject even though she does not have our real return address. Fortunately, rOnl's message system includes a feature that makes this possible.

Recall that when one rOnl user sends a message to another, the reply to that question need not come from the original recipient. That is, if some rOnl user Alice sends a message to another user Cindy, Cindy may choose to forward the email containing the question to a third party, Bob. Bob may click the 'Respond Now' button in the body of the email, log in to rOnl, and compose a response; the response will be delivered to the original sender, Alice.

Using this feature, consider the situation shown in Figure 7. Suppose that the researcher controls the nodes designated Alice and Cindy. The experiment — which we call *Experiment 1* — proceeds as follows:

1. Alice composes a message using the rOnl question interface. She writes it as though it is addressed to Bob,

[3]This is analogous to the experiment by Jagatic et al. [8] in which an authenticator for the domain users were requested credentials for was used to verify these; in our setting though, it is less straightforward, given the lack of collaboration with rOnl.

including context information about Bob, but sends it instead to the other node under our control (Cindy).

2. Cindy receives the question and forwards to Bob, hiding the fact that she has handled it (e.g., through spoofing). The apparent sender of the message is still `member@ronl.com`.

   Note that at this point, Cindy also has the option of making other changes to the body of the email. This fact will be important in duplicating the other feature of a phishing attack — the malicious link. For now, assume that Cindy leaves the message text untouched except for changing recipient information in the text of the message (to make it appear as though it was always addressed to Bob).

3. If Bob chooses to respond, the response will come to Alice.

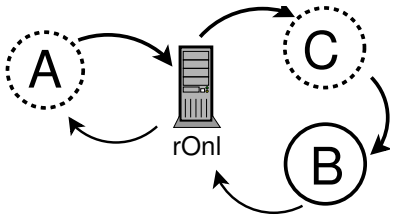
We measure the success rate of this experiment by considering the ratio of responses received to messages we send. Notice that our experiment sends messages using spoofing, making them just as likely to be caught in a spam filter as a message that is a component of a spoofing attack (such as the attacks described above). However, our message does not contain a malicious link (Figure 8(a)) — thus it simulates only one of the features of a real phishing attack.

It's important to note that spam filters may attempt to detect spoofed or malicious messages in many different ways. For the purposes of our experiments we make the simplifying assumption that the decision (whether or not the message is spam) is made without regard to any of the links in the message; however, in practice this may not be the case. We make this assumption to allow us to measure the impact that a (seemingly) malicious link has on the user's likelihood to respond.

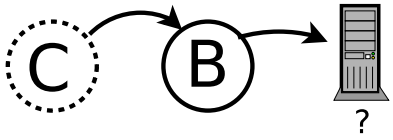
Note that in order to respond, Bob must click the 'Respond Now' button in our email and enter his credentials. Simply pressing "reply" in his email client will compose a message to `UseTheYellowButton@ronl.com`, which is the `reply-to` address rOnl uses to remind people not to try to reply to anonymized messages.

Note that Experiment 1 is just a convoluted simulation of the normal use scenario, with the exception of the spoofed originating address (Figure 1). If Bob is careful, he will be suspicious of the message in Experiment 1 because he will see that it has been spoofed. However, the message will be completely legitimate and in all other ways indistinguishable from a normal message. Bob may simply delete the message at this point, but if he clicks the 'Respond Now' button in the message, he will be taken directly to rOnl. It is possible he will then choose to answer, despite his initial suspicion. Thus Experiment 1 gives us an upper bound on the percentage of users who would click a link in a message in a context-aware attack. This is the percentage of users who either do not notice the message is spoofed, or overcome their initial suspicion when they see that the link is not malicious.

To measure the effect of the context information in Experiment 1, we construct a second experiment by removing it. We call this Experiment 2; it is analogous to the non-context-aware attack (Figure 8(b)). In this experiment, we omit the rOnl username and registered real-life name of the recipient, Bob. Thus, the number of responses in this experiment is an upper bound on the number of users who would be victimized by a non-context-aware phishing attack.



(a) Our experiment's communication flow



(b) C spoofs a return address when sending to B, so B should perceive the message as a spoofing attack.

Figure 7: Experimental setup for Experiments 1 and 2. Nodes A and C are experimenters; node B is the subject. A sends a message to C through rOnl in the normal way; C spoofs it to B. The involvement of node C is hidden, making node B perceive the situation as the spoofing attack in (b); but if B answers anyway, the response will come to A.

4.2 Experimenting with a Malicious Link

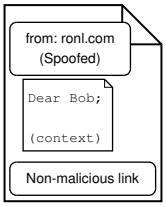
Here, our challenge is to simulate a malicious link in the email — but in such a way that the following are true:

1. The site linked from the malicious link asks for the user's authentication information

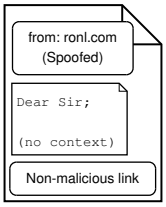
2. We have a way of knowing if the user actually entered their authentication information, but,

3. The entering of this authentication information does not compromise the user's identity in any way — in particular, we must never have the chance to view or record it.

Recall that Cindy in Experiment 1 had the chance to modify the message before spoofing it to Bob. Suppose that she takes advantage of this chance in the following way: instead of the link to rOnl (attached to the 'Respond Now' button) that would allow Bob to answer the original question, Cindy inserts a link that still leads to rOnl but *appears* not to. One way that Cindy may do this is to replace `signin.ronl.com` in the link with the IP address of the server that `signin.ronl.com` refers to; another way is to replace `signin.ronl.com` by a domain that Cindy has



(a) Experiment 1 - Spoofed originating address, but real link



(b) Experiment 2 - Spoofed originating address, real link, but poorly written message text

Figure 8: Spoofed messages without malicious links. These messages have a strong chance of being caught in a spam filter, but may appear innocuous even to a careful human user.

registered as a synonym; that is, a domain that looks different, but resolves to the same IP.

This link then fulfills the three requirements above — not only does it certainly appear untrustworthy, but it requests that the user log in to rOnl. We can tell if the user actually did, for we will get a response to our question if they do — but since the credentials really are submitted directly to rOnl, the user's identity is safe.

4.3 Simulating a Real Attack

Combining the two techniques above, then, let us simulate a real phishing attack. The experiment performing this simulation would proceed as follows:

1. Alice composes a message as in Experiment 1.
2. Cindy receives the question and forwards to Bob, hiding the fact that she has handled it (e.g., through spoofing). Before forwarding the message, Cindy replaces the 'Respond Now' link with the simulated malicious link.
3. If Bob chooses to respond, the response will come to Alice.

Call this experiment *Experiment 3*. See Figure 9, the setup of this experiment, and Figure 10(a) for a summary of the features of this experiment email.

Note that the message that Bob receives in this experiment is principally no different (in appearance) than the common message Bob would receive as part of a spoofing attack; it has a false sender and a (seemingly) malicious link. Thus, it is almost certain that Bob will react to the message exactly as he would if the message really was a spoofing attack.

We also define a contextless version, *Experiment 4*, in which we omit personalized information about the recipient (just as in Experiment 2). Figure 10(b) illustrates the key distinction between Experiments 3 and 4. In Experiment 4, the number of responses gives an upper bound on the number of victims of a real phishing attack — anyone who responds to this experiment probably has ignored many cues that they should not trust it. Figure 11 summarizes our four experiments in contrast to real phishing attacks.

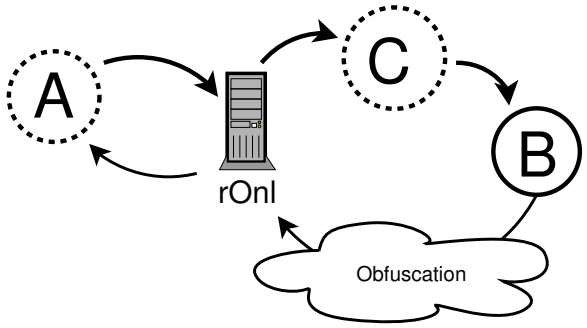
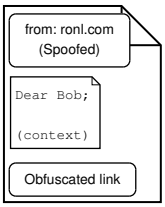
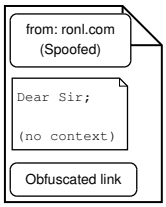


Figure 9: Communication flow for experiments 3 and 4. Node C uses spoofing to make the message to B appear to come from `member@ronl.com`, and obfuscates the link to `signin.ronl.com` to make it appear malicious. B should perceive the communication as a phishing attack.



(a) Experiment 3 - Spoofed originating address and simulated malicious link



(b) Experiment 4 - Experiment 3 without context information

Figure 10: Spoofed messages with simulated malicious links. The message in (b) simulates a phishing attack currently in the wild; the message in (a) simulates a more dangerous *context-aware* phishing attack.

4.4 Experiment Design Analysis

In summary, we have constructed experiments that mirror the context-aware and non-context-aware attacks, but do so in a safe and ethical manner. The emails in our experiments are indistinguishable from the emails in the corresponding attacks (Figure 12). That is, if in Experiment 3 we receive (through Alice) an answer from Bob, we know that Bob has entered his credentials to a site he had no reason to trust — so we can consider the probability that we receive a response from Bob to be strongly indicative of the probability Bob would have compromised his credentials had he received a real phishing attack. Refer to Figure 11; our goal is to have each experiment model a real attack's *apparent* phishiness (that is, to a user, and to automated anti-phishing methods), while not actually being a phishing attempt.

In the above, we use the term *indistinguishable* in a different manner than what is traditionally done in computer security; we mean indistinguishable to a human user of the software used to communicate and display the associated information. While this makes the argument difficult to prove in a formal way, we can still make the argument that the claim holds, using assumptions on what humans can dis-

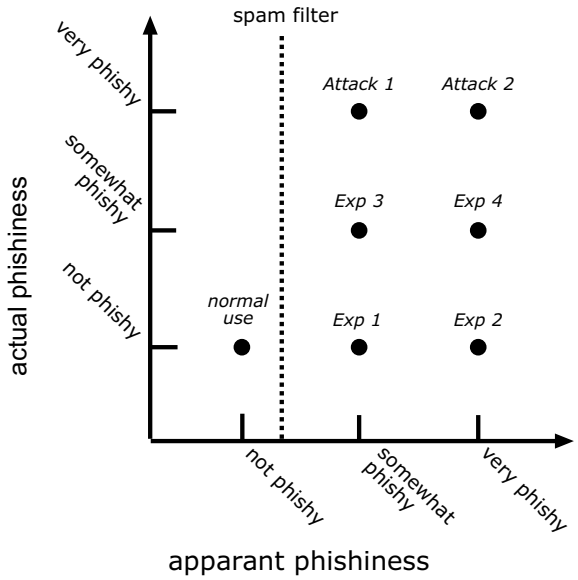


Figure 11: Our four experiments, contrasted with the phishing attacks they model, and the normal use scenarios which they imitate. Attacks that appear "somewhat phishy" are those that can be recognized by close scrutiny of the source of the message, but will look legitimate to casual investigation. "Very phishy" appearing attacks will be rejected by all but the most careless users. In the context of actual phishiness, "somewhat phishy" messages with deceptive (but not malicious) links, and "very phishy" messages are those which attempt to cause a user to compromise his identity. Any message to the right of the "spam filter" line may potentially be discarded by a spam filter.

tinguish. Thus, we see that Experiment 1 (normal use, but spoofed) is indistinguishable from Experiment 3 (obfuscated link and spoofed) for any user who does not scrutinize the URLs. This is analogous to how — in the eyes of the same user — an actual message from rOnl (which is simulated in Experiment 1) cannot be distinguished from a phishing email with a malicious link (simulated by Experiment 3). However, and as noted, we have that messages of both Experiments 1 and 3 suffer the risk of not being delivered to their intended recipients due to spam filtering. This is not affecting the comparison between Experiment 1 (resp. 3) and real use (resp. phishing attack).

More in detail, the following argument holds:

1. A real and valid message from rOnl cannot be distinguished from a delivered attack message, unless the recipient scrutinizes the path or the URL (which typical users do not know how to do.)

2. A delivered attack message cannot be distinguished from an experiment 3 message, under the assumption that a naïve recipient will not scrutinize path or URLs, and that a suspicious recipient will not accept an obfuscated link with a different probability than he will accept a malicious (and possibly also obfuscated) link.

Similarly, we have that a phishing attack message that has only partial context (e.g., does not include the recipient's rOnl user name, as is done in real communication from rOnl) cannot be distinguished from an experiment message with a similar degradation of context (as modeled by Experiment 4).

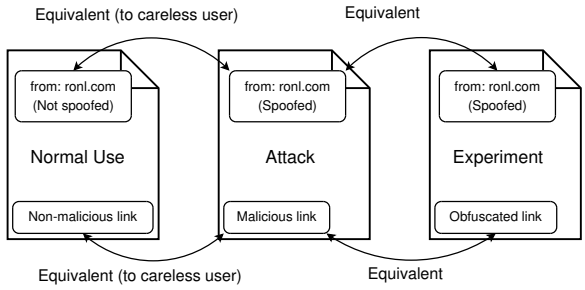


Figure 12: Our experimental email is indistinguishable from a phishing attack to the savvy user; to the careless user, it is also indistinguishable to normal use.

5. METHODOLOGY

5.1 Identity Linkage

The first step in performing our experiments was establishing a link between rOnl users' account names and their real email addresses. To gather this information, we sent 93 rOnl users a message through the rOnl interface. We selected the users by performing searches for the keywords 'baby clothes' and 'ipod' and gathering unique usernames from the auctions that were given in response.[4]

We chose not to anonymize ourselves, thus allowing these users to reply using their email client if they chose. A previous experiment by Jakobsson [4] had suggested that approximately 50% of users so contacted would reply from their email client rather than through rOnl, thus revealing their email address. In our experiment, 44 of the 93 users (47%) did so, and we recorded their email addresses and usernames.

We also performed Google searches with several queries limited to `cgi.ronl.com`, which is where rOnl stores its auction listings. We designed these queries to find pages likely to include email addresses.[5]

We automated the process of performing these queries and scanning the returned pages for email addresses and rOnl usernames; by this means we collected 237 more email and username pairs. It's important to note that we cannot have complete confidence in the validity of these pairs without performing the collection by hand. We chose to do the collection automatically to simulate a phisher performing a large-scale attack.

5.2 Experimental Email

Our goal was to try each experiment on each user, rather than splitting the users into four groups and using each user as a subject only once. This gives us more statistical significance, under the assumption that each trial is independent

[4]Most automated data collection was done in the Perl programming language, using the `WWW::Mechanize` package [5].
[5]These queries were "`@ site:cgi.ronl.com`", "`@ ipod site:cgi.ronl.com`", and "`@ "baby clothes" site:cgi.ronl.com`"

— that is, the user will not become 'smarter,' or better able to identify a phishing attack, after the first messages. We believe this assumption is a fair one because users are exposed to many phishing attacks during normal internet use. If receiving a phishing attack modifies a user's susceptibility to later attempts, the user's probability to respond to our experiments has already been modified by the unknown number of phishing attacks he or she has already seen, and we can hope for no greater accuracy.

In order that the experimental messages appear disjoint from each other, we used several different accounts to send them over the course of several days. We created 4 different questions to be used in different rounds of experiments, as follows:

1. `Hi! How soon after payment do you ship? Thanks!`
2. `Hi, can you ship packages with insurance for an extra fee? Thanks.`
3. `HI CAN YOU DO OVERNIGHT SHIPPING??`
4. `Hi - could I still get delivery before Christmas to a US address? Thanks!!` (sent a few weeks before Christmas '05).

As previously mentioned, rOnl places a limit on the number of messages that any given account may send in one day; this limit is determined by several factors, including the age of the account and the number of feedback items the account has received.

Because of this, we only created one message for each experiment. We sent this message first to another account we owned, modified it to include an obfuscated link or other necessary information, and then forwarded it (using spoofing) to the experimental subjects.

As discussed earlier, a real phisher would not be effectively hampered by this limitation on the number of potential messages. They might use accounts which they have already taken over to send out messages; every account they took over would increase their attack potential. They might also spam attacks to many email addresses, without including a rOnl username at all.

6. RESULTS

The results of our experiments are summarized in Figure 13.

Experiment	Response Rate
No name, good link (Exp 2)	19% ± 5%
Good name, good link (Exp 1)	15% ± 4%
Good name, "evil" IP link (Exp 3)	7% ± 3%
Good name, "evil" Subdomain link (Exp 3)	11% ± 3%

Figure 13: Results from our experiments. It's interesting to note that the presence or absence of a greeting makes no significant difference in the user's acceptance of the message. The intervals given are for 95% confidence. Note that we did not attempt Experiment 4, opting for two trials of Experiment 3 with different parameters instead.

These results indicate that the absence of the greeting text at the top of each message has little to no effect on the user's chance to trust the contents of the message. This finding is significant, because rOnl states that the presence of a user's registered name in a message addressed to them signifies that the message is genuine. It seems that users ignore this text, and therefore its inclusion has no benefit; identity linkage grants no improvement in the success rate of an attack.

However, we observe a significant drop in the number of users who will follow a link that is designed to look malicious. Note that the success rate for the attack simulated by a subdomain link is significantly higher than that predicted by Gartner. Further, Gartner's survey was an estimation on the number of adult Americans who will be victimized by at least one of the (many) phishing attacks they receive over the course of a year. Our study finds that a single attack may have a success rate as high as $11 \pm 3\%$ realized in only 24 hours.

7. CONCLUSION

This paper has presented a set of techniques for the ethical and safe construction of experiments to measure the success rate of a real phishing attack. Our experiments can also be constructed to measure the impact of the inclusion of various types of context information in the phishing attacks. While we use rOnl as a case study because a feature of its design permits the construction of an ethical phishing simulation we believe our results (with respect to the success rate of attacks) are applicable to other comparable populations.

We also present the results of several phishing experiments constructed by our techniques. We find that identity linkage had little or no effect on the willingness of a given user to click a link in a message. We also find that even with the effects of modern anti-spoofing and anti-phishing efforts, more than 11% of rOnl users will read a spoofed message, click the link it contains, and enter their login information.

8. REFERENCES

[1] Mailfrontier phishing IQ test. `http://survey.mailfrontier.com/survey/quiztest.html`.

[2] Know your enemy : Phishing. behind the scenes of phishing attacks. `http://www.honeynet.org/papers/phishing/`, 2005.

[3] Garfinkel, S., and Miller, R. Johnny 2: A user test of key continuity management with S/MIME and Outlook Express. Symposium on Usable Privacy and Security.

[4] Jakobsson, M. Modeling and preventing phishing attacks. In *Financial Cryptography* (2005).

[5] Lester, A. WWW::Mechanize - handy web browsing in a perl object. `http://search.cpan.org/~petdance/WWW-Mechanize-1.16/lib/WWW/Mechanize.p%m`, 2005.

[6] Litan, A. Phishing attack victims likely targets for identity theft. *FT-22-8873, Gartner Research* (2004).

[7] M. Jakobsson, T. Jagatic, S. S. Phishing for clues. `www.browser-recon.info`.

[8] T. Jagatic, N. Johnson, M. J., and Menczer, F. Social phishing. 2006.