Faheem: Explaining URLs to people using a Slack bot

Faheem: Explaining URLs to people using a Slack bot

Kholoud Althobaiti[§,†], **Kami Vaniea**[§], and **Serena Zheng**[‡]

k.althobaiti@sms.ed.ac.uk, kvaniea@inf.ed.ac.uk, serenaz@princeton.edu

[§]University of Edinburgh, Edinburgh, UK

[†]Taif University, Taif, KSA

[‡]Princeton University, Princeton, New Jersey, USA

ABSTRACT

Online safety regularly depends on users' ability to know either where a URL is likely to lead or identify when they are on a site other than they expect. Unfortunately, the combination of low URL reading ability in the general population and the use of hard-to-detect approaches like look-alike letters makes the reading of URLs quite challenging for people. We design a Slack bot, named Faheem, which assists users in identifying potentially fraudulent URLs while also teaching them about URL reading and common malicious tactics. In this work, we describe the design of the bot and provide an initial evaluation. We find that Faheem does a good job of interactively helping users identify issues with URLs, but Faheem users show minimal retention of knowledge when they lose access to the tool.

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous; K.6.5. Management of Computing and Information Systems: : Security and Protection

Author Keywords

Phishing; usable privacy and security; real-time learning; security education

INTRODUCTION

Uniform Resource Locators (URLs) are how the majority of internet citizens find information on the world wide web. "Linking" between web pages, chat messages, social media, or even emails is a common method of telling someone else how to find a piece of content. When asked to visit a physical space in the real world using a provided address, most people are able to pull up a map in advance which allows them to answer important questions like: "How far away is it?" or "Does Google Maps think that there really is an Office Depot there?" But with an online URL, people seem to have more difficulty asking and answering basic questions about the location they are visiting, for example: "Is this really the website for Office Depot?" or "Will my password be sent to the website safely so no one else can read it?"

AISB'18, April 4–6, 2018, Liverpool, UK
ACM xxx-x-xxxx-xxxx-x/xx/xx…$15.00
DOI: http://dx.doi.org/xx.xxxx/xxxxxxx.xxxxxxx

The goal of Faheem is to help people understand the content of URLs so that they can ask and answer questions about the URL, in particular, where it leads.

There are various reasons why understanding URLs can be useful, ranging from avoiding being Rickrolled to being able to identify when personal information is being sent in the URL. Phishing is likely the most financially impactful use case. Phishing attacks involve scammers attempting to obtain users' sensitive information for malicious reasons, with the individuals behind such attacks seeking to deceive users into visiting websites that impersonate legitimate ones [17]. One of the many reasons phishing works is that users cannot accurately read a URL to determine if it really is associated with an organization they interact with or not [8, 25].

Phishing is also quite expensive, costing the United Kingdom (UK) economy as much as £280 million a year [6]. Only about 72% of consumers in the UK even know what "phishing" is even though 92% of organizations report training users to identify and avoid phishing attacks [3]. Which is wise, since 98% of attacks involving a social element use phishing [2].

With the evolution of social media, instant messaging services, such as Slack and WhatsApp messengers, have become the main communication means between friends, relatives and colleagues [13]. These services allow end users to share links and files. However, on the heels of the adoption of such features, phishing on these new channels has become a threat [26]. More specifically, the manipulation of URLs is a popular phishing approach [11] which takes advantages of people's vulnerabilities when interacting with technology, and the characteristics of URLs, which makes it difficult for users to interpret them correctly in order to distinguish legitimate websites from those that are spoofed [21]. For example, URLs are read both left to right (path) and right to left (domain), URLs can be shortened, or URLs can be represented by an IP address, all of which is confusing for users [27].

We present here a new Slack chatbot called Faheem which helps users by parsing URLs for them and explaining the URL elements in a user-friendly way. The goal of Faheem is to both assist a user during a regular chat communication, and help them learn useful URL reading techniques. Ideally, a more polished version of Faheem could be installed on a company public Slack channel to provide contextual information about

the various URLs being shared and protect employees from erroneously visiting problematic URLs.

We test Faheem against a simplistic URL explanation web page – called URL Explainer – and find that Faheem does a better job of supporting people interactively as well as helping them to retain the knowledge.

BACKGROUND AND RELATED WORK

Uniform Resource Locators (URL)

A URL is a structured description of the location of a digital resource [5] as pictured in Figure 1. Note that URLs do not state where the resource is actually located, merely how to go about locating it, the difference is subtle but a key to understanding some of the design decisions. For example, URLs can contain usernames and passwords, this information is necessary to locate resources behind a login prompt, but strictly speaking, has nothing to do with the actual location of the resource. Similarly, the query string exists so that the requester can pass strings to the host computer and get back the desired resource. Again, query strings help locate the resource in places like databases but do not strictly describe the actual resource location. When reading URLs the distinction becomes important because some URLs are actually the location of a second URL (redirection), such that a basic reading is insufficient to learn the final destination.

At a high level, end-user issues around URL readability focus on: 1) who or what is being communicated with (host, port), 2) what is being said (username, password, path, query string), and 3) how it is being said (protocol). Who is being communicated with is typically a fraudulent communication (phishing) issue where the malicious actor is trying to trick the user into going to the wrong host. What is being said is typically a privacy issue where more information is communicated than the user would like, such as communicating unique marketing IDs via query strings which can include anything the URL creator wants. For example, the HealthCare.gov site which is used by United States citizens up for health care allows users to click on links which take them to different private insurer websites. Those links were found to include information like pregnancy status in the URL query string, effectively sending sensitive data from inside of HealthCare.gov to a private insurer the user had no current relationship with [22]. This behaviour is insecure as the query could be saved in server logs and the browser's history log, which is a potential confidentiality breach [28]. The last issue is about how the information is said which is typically an issue of encryption (http vs. https). In this paper, the primary focus of Faheem is to raise users awareness of the who issue, notably, the phishing techniques.

URL Manipulation Tricks

Phishers will often use URL manipulation approaches to make the URLs they send people look legitimate and deceive the victim into believing they are visiting a trusted website [11]. The following are a set of common tactics used to hide the malicious destination of a URL [14, 27]:

- **Obfuscate:** The company name is not visible in the URL, which could be owing to the use of the IP address in a hostname part, or shortened or redirected links.
- **Mislead:** The expected company name is embedded somewhere in the URL where the user can see it – possibly in the subdomain, pathname or credentials – but that company is not the destination of the URL.
- **Mangle:** The company name has letter substitution, misspelling or non-ASCII characters (similar to English ones), resulting in visually identical web addresses, known as a Homograph attack [16].
- **Camouflage:** The company name contains an extension in the domain name, such as a different top-level domain or deliminator-looking character other than the normal period; this is usually done with the addition of a hyphen. For example, the use of *home-depot.com* instead of *homedepot.com*.

Detecting Malicious URLs

The work related to detecting malicious URLs falls into two main approaches: automated detection and user training.

Automated Phishing Detection

Automated phishing detection uses a combination of many factors to detect phish, which includes the URLs in the communication. These detection tools are used by various groups. Large organizations will use them to scan all incoming communication such as email and proactively remove communications that are known to be fraudulent. Individual users can also download tools for their browsers and other communication clients that will identify fraudulent communication and either remove or warn about it [17]. There are also bots, such as MetaCert, which scans communications in Slack channels [1].

Most phishing identification procedures depend on Blacklists, meaning a list of phishing URLs [20]; however, these tools do not prevent zero-hour attacks, which is the attack before the malicious URL is discovered [17]. The Anti-Phishing Working Group revealed that the normal time taken to discover a phishing URL is 28.75 hours, during which time users are unprotected [9]. These tools can sometimes give false warnings that decrease users' trust in the results and cause them to ignore future warnings; consequently, the effectiveness of these tools relies on users' behaviour [12].

Training Users

While automation is a good idea, and effective, it is currently impossible to completely remove the user from the loop. Communication is an important part of business operations and overly aggressive automatic filters are likely to cost organizations in lost productivity. As a result, some phishing attempts will get through the automated filters, necessitating the training of users as a second complementary line of defence. There are two common types of training: upfront and embedded.

In upfront training, a user will go through a training session where they will learn about phishing in a condensed format. Examples include the Anti-Phishing Phil game [25] and NoPhish app [7] both of which train users to read URLs using concentrated engagement, such as a game. The upfront approach effectiveness relies on the user being able to understand the materials, retain them, and be able to apply them to daily activity. Prior work demonstrates that upfront training is effective when it comes to enhancing users' capability to

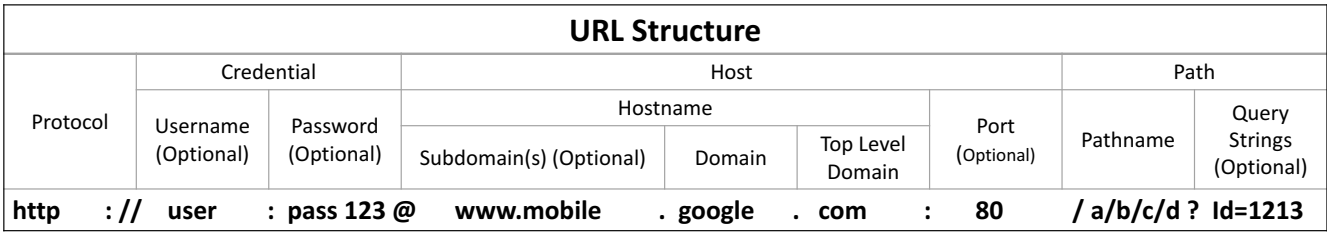


Figure 1: URL structure and example.

identify phishing URLs; however, the long-term benefit of this approach is uncertain [17]. Importantly, this approach can fail to produce a long-term advantage [18] because of the nature of forgetting [17]; along these lines, Volkamer et. al [27] recommends that users need to integrate training into their daily life. Another issue is that people are unwilling to invest energy in online instructional exercises, particularly given the perceived low risk of being exposed to real danger [10, 15, 27].

Embedded training involves integrating the training into the daily life of the users. The most classic example of which is sending out fake phishing emails to employees and providing contextual training for those who click on the links [19]. Unlike upfront training, embedded training is fairly lite, requiring small amounts of time for most users and more time only for users who click the malicious links. However, due to its lite touch, users may not get the opportunity to build a strong conceptual model of how phishing works; making the lessons harder to apply in different contexts. Because this kind of training is embedded in routine, it is challenging to create consistent security training messages across an organization or worse, between organizations, potentially leaving users with conflicting advice [15, 23].

FAHEEM BOT DESIGN

The objective of this work is to develop and test Faheem: a Slack bot with the capacity to parse URLs posted in a Slack channel and clarify their components. It also warns users about suspicious patterns using friendly explanatory language that users can understand.

Our primary design objective is to create an interactive chat bot which helps average internet users correctly read URLs and identify phishing URLs. In order to accomplish this goal we focus on two features of the bot:

1. Parsing the URLs and identifying common malicious behaviours focusing primarily on the domain issues.

2. Presenting the results to the user in a clear and easy to understand manner.

Platform

We selected Slack as the platform for the bot because Slack is a commonly used communication platform with good support for custom bots. Slack bots can join any group, read and post messages and also contact members in direct messages.

URL Parser

The URL parser uses the Node.js programming language. The detailed processes for the URL analysis is as follows:

1. Listens to all Slack chats in the forum and extracts URLs using *url-regex* package.

2. Identifies and resolves IP address. The *ip-regex* package was used to detect IP address while the *dns* constructor package was used to reverse it to obtain the registered hostname.

3. Checks and resolves redirects and shortened links using the *unfurl-url* package to obtain the destination URL.

4. Parses the final URL into its component parts as shown in Figure 1 by using the built-in *URL* constructor provided by *node.js*.

5. Checks the domain for similarity with domains of the top 500 websites on Alexa Global Sites. Using the Levenshtein distance metric from *clj-fuzzy* package.

6. Checks for non-ASCII characters using the *non-ascii* and *langdetect* packages.

Walkthrough Example

For clarity, we detail here a sample interaction between the Faheem bot and a user Alex also pictured in Figure 2.

The user Alex starts the interaction by posting a URL into a Slack group the bot is listening to, which Faheem then detects. Faheem parses the URL and presents the most important information to Alex first with an offer of further details on request. In this case, Faheem detected that the subdomain is similar to popular domain google and warns the user that this URL will not go to Google. It also detects a small edit distance between the domain 'instaran' and the popular domain 'instagram', which it points out to Alex along with actionable advice on what to do if she is unsure. Finally, it provides positive feedback that the URL uses HTTPS and is therefore encrypted in transit.

Alex wants more details so she replies with "details". Faheem expands each of the previously presented sections and provided general advice for users, such as: "To clarify, the hostname is similar to reading the home address, etc" in order to help them develop conceptual understanding to deal with security risks.

Alex is confused about the Protocol section and asked Faheem about it by typing: "protocol". Faheem responds by explaining what a protocol is, particularly clarifying about HTTP.

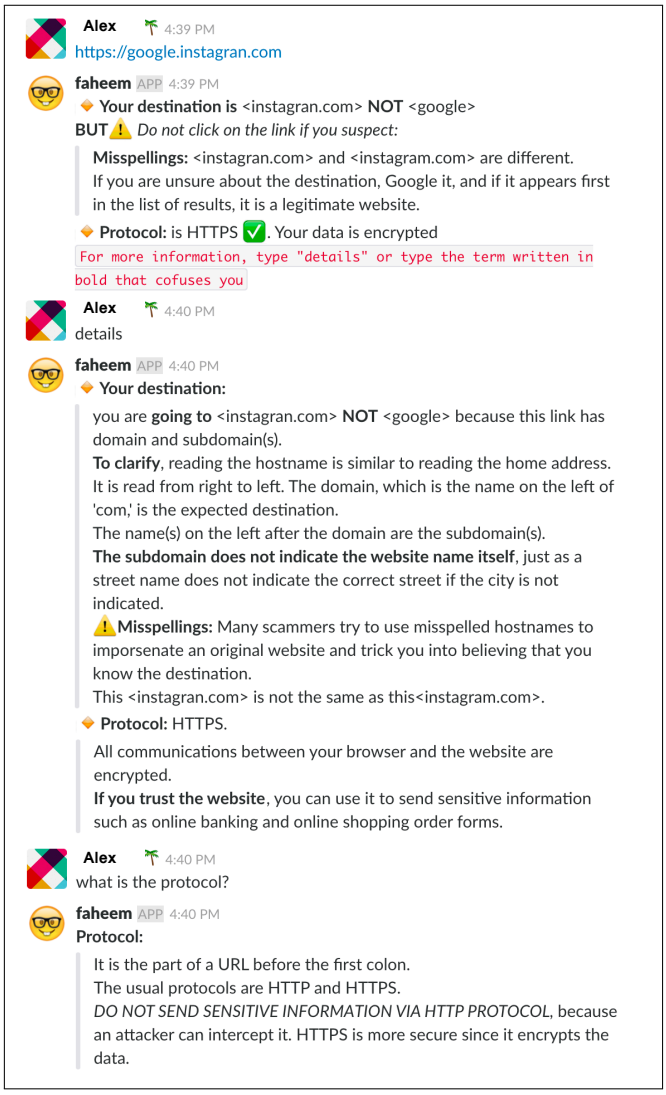


Figure 2: An example interaction with the Faheem bot.

Explanation of Design Choices

Simple initial information Users need to receive information just at the time they need it, especially since people may struggle to remember information that they have received out of context [4]; therefore, on detection of URLs, Faheem presents a concise summary format of the key information and replacing the technical concepts with terms which can be understood by average users. The key information and format of the summary were agreed by a focus group of security and HCI experts. Domain issues are quite serious security wise as the user may be communicating with someone other than they intend [27]. Hence, Faheem focuses initial information on issues around the domain including warnings for known malicious URL tricks, as shown in Figure 3.

User-lead interaction Faheem gives the users a chance to ask for details in general or for specific information. The goal is to make the interaction user-lead where the user can decide what they are most interested in seeing rather than provide piles of information up-front. Conceptual explanations are also provided to help people build relations between concepts and assist with applying to learned lessons to new situations [24].

Highlight most problematic elements with evidence Where possible, Faheem uses evidence from the URL itself to demonstrate potential issues to users in such a way that they can understand the issue and bring their own expertise to bear. For example, Faheem checks for non-ASCII characters and when found it points out to the user that there are, say, Cyrillic letters in a mostly ASCII URL and shows them which letter is non-ASCII. Other problems like potential misspellings of a common domain are also contextualized by stating both the domain in the URL and the common one so that the user can compare them by themselves. Moreover, where best practices exist, Faheem provides expert advice and positive reinforcement of certain actions. For example. HTTPS is almost always a better choice than HTTP so Faheem puts a green check mark to indicate that having HTTPS is a good feature of the URL.

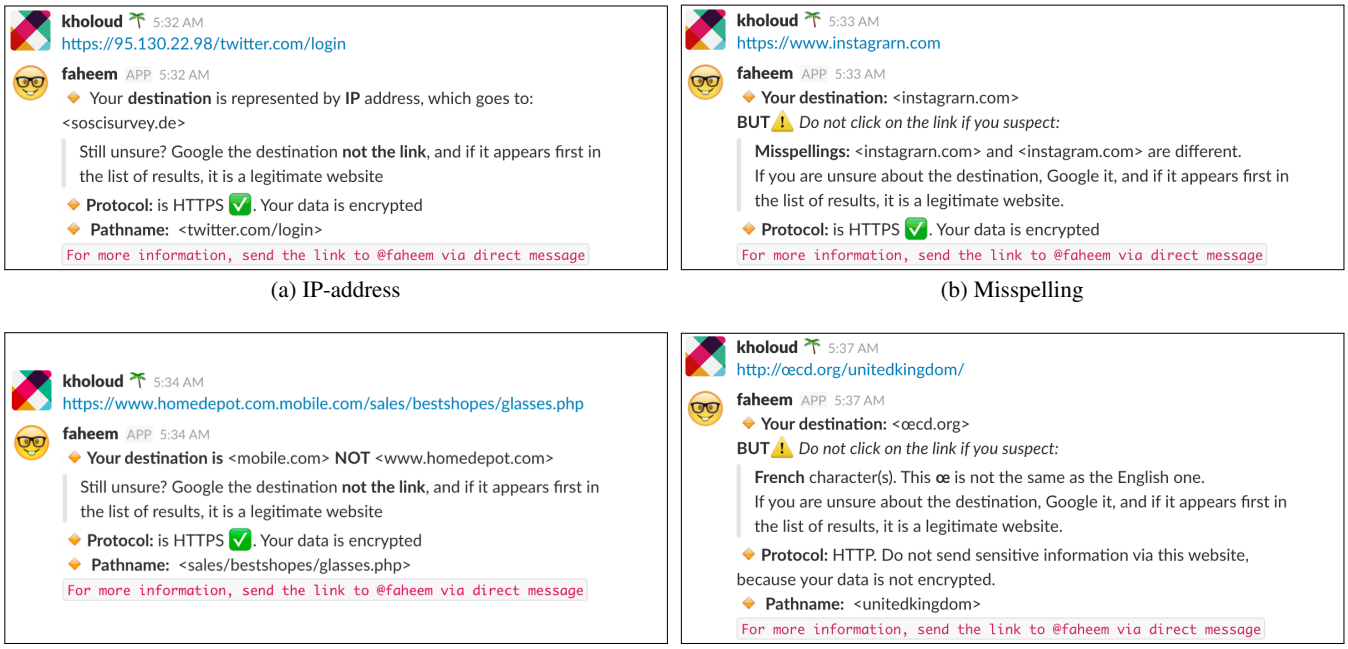
Explanation with advice Faheem provides general and actionable advice. The general advice follows the clarifications to help users to deal with URLs. For example, not to send sensitive data through a HTTP connection. Users are advised to take an action when they doubt a URL (procedural knowledge). This provides them with clear choices and potentially increases their ability to differentiate the original from the spoof URL. For example, Faheem advises them to Google the domain if they are unsure about its safety.

URL EXPLAINER DESIGN

URL Explainer is a website created by one of the authors as a class project when studying abroad. It takes in a URL, feeds it into the URL.js parser, and presents the results on a webpage. Each presented element of the URL is pulled out separately onto different lines where the components are highlighted and a generic explanation provided. URL Explainer also attempts to fetch the URL server-side to get its title and preview. An example can be seen in Figure 4.

A small pilot was run with 14 university students to see if URL Explainer could be used to improve URL reading skills. The study had a simple three-part format, with a pre-test, a test where they could use URL Explainer, and a post-test. We found, unsurprisingly, that participants are bad at identifying the destination of a URL; participants had an average accuracy of just over 50% in the pre-test. When using URL Explainer, participants jumped to 100% accuracy while the control condition which had no assistance stayed at 50% accuracy. Unfortunately, when URL Explainer was taken away, experimental participants dropped to an accuracy of 54% compared to the control which had a post-test accuracy of 34%. The overall take away from the study was that URL Explainer did help people correctly identify the end destination of the URL, but using it did not lead to skill building or retention.

In this paper, we will be using URL Explainer as a control condition to compare Faheem with. We selected URL Explainer as a control condition because it is comprehensive, simplistic, and shown to be effective at helping a user read a



(a) IP-address

(b) Misspelling

(c) Multilevel domain (subdomains)

(d) Non-ASCII characters in the domain

Figure 3: Sample Faheem messages for different malicious URL patterns.

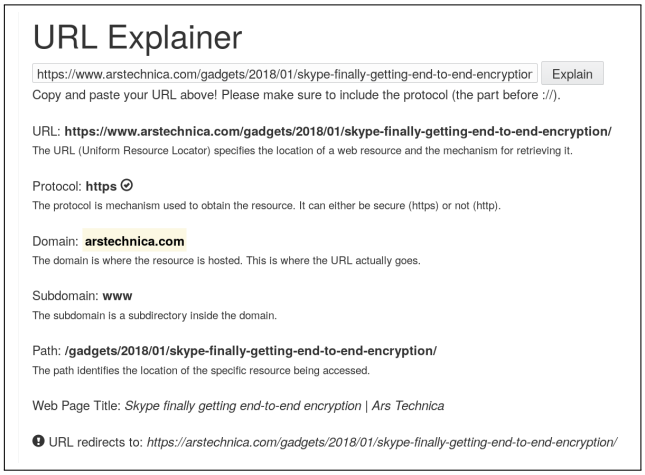


Figure 4: URL Explainer interface after the user has asked it to parse an Ars Technica news article URL where "www" has been added as a subdomain to show the URL redirect notice.

URL. Unlike Faheem, URL Explainer makes no attempt to identify Phishing indicators or provide contextual information, instead focusing solely on factual presentation of the contents of the URL itself. As such, it is a good choice for a control.

METHODS

An empirical lab study was conducted to investigate the overall effectiveness of Faheem's interactive explanations in raising users awareness of phishing URLs as compared to a basic presentation.

We hypothesise that Faheem users show a greater improvement, compared to URL Explainer users, in their ability to identify phishing URLs in the following two conditions: (i) With the support of the tool (Faheem or URL Explainer). (ii) When access to the tool has been removed.

Participant Recruitment

A request was posted by the lead researcher on their accounts for Twitter, WhatsApp, and Facebook. As a motivation, prospective participants were told that they would be awarded £10 for their participation. Only three people were located locally, the others were from a wide variety of countries including Saudi Arabia and parts of the European Union. The participants were from a variety of sectors including mathematics, business, and management. A total of 40 participants were recruited, 20 for each group, all of whom were aged between 20 and 58 years old with a mean of 28. 60% were female and 40% male.

Study Design

Because of the wide geographic locations of the participants, the study was conducted remotely with the researcher communicating with the participants via email and Slack using a pre-defined script which differed between conditions only in the explanation of the functionality of the systems.

Protocol

Setup: The study purpose was explained to the participants from each group, as well as what phishing is if they were not

already aware. They were explicitly told to not visit any of the links, only to read them. They were also asked at the end of the study if they had used any external resources. Participants in the Faheem condition were asked to join a Slack team before the study started in attempt to limit the communication means between researcher and participants and ensure smooth study flow. They were invited to an empty Slack channel to which Faheem was added later in the study.

Demographics: Participants in both groups started the session with a consent form and a demographics survey where they were asked for their Slack username (Faheem) or preferred first name for communication (control), age, gender and topics they have previously studied, with two additional questions incorporated for the experimental group, asking how frequently they use Slack and chatbots to ensure their understanding of Slack and chatbots would not influence the study results.

Pre-test: Participants were given a set of 14 URLs one at a time via a survey and instructed to imagine that they had received each URL during an instant messaging interaction with the text "You want to visit <website name>" associated with the URL. For each URL the participants answered the following questions:

1. Decide whether it is a phishing or an original website.
   Select one: phishing, original
2. Which part of the link does influence your answer?
   Multi-answer: all elements of URL, except the protocol, were provided as choices.
3. Why would you click / not click on the link?
   Free-text answer.

The goal of these questions was to determine their a-priory ability to determine if the URL went to the stated organization or not.

Supported reading: Participants were given access to either a live version of Faheem or screenshots of URL Explainer and asked to use them to answer the same set of questions shown in the pre-test, but with a different set of URLs. For the Faheem group, the participants were given access to Faheem and told that a link would be posed in the group, which Faheem automatically would parse, and questions were then sent in the group chat. For the control group, members were given a survey with screenshots of URL Explainer for the link in question; screenshots were to ensure that easy access to URL Explainer could be revoked during the post-test below. Since URL Explainer produces static output, there is no functional difference between the actual page and an image.

Post-test Similar to the pre-test, participants in both groups were given a new set of URLs and asked to answer the same questions from the pre-test without the support of the tool. Participants were again asked to not type in URLs or use other resources. Access to Faheem was revoked and URL Explainer participants were asked if they had searched for the site online.

Tested URLs

In each of the pre, supported, and post stages the participant is given a set of 14 URLs, which were selected to cover the following phishing techniques: shortened links, redirects, IP addresses, misspellings, multi-level domains, company name located somewhere other than the host position, and non-ASCII characters. The three sets of URLs were selected to be comparable in structure but not identical. Every participant saw the same URLs in the same order. One URL from each stage was excluded from analysis due to a technical issue during data collection. Results are drawn from 13 URLs per stage.

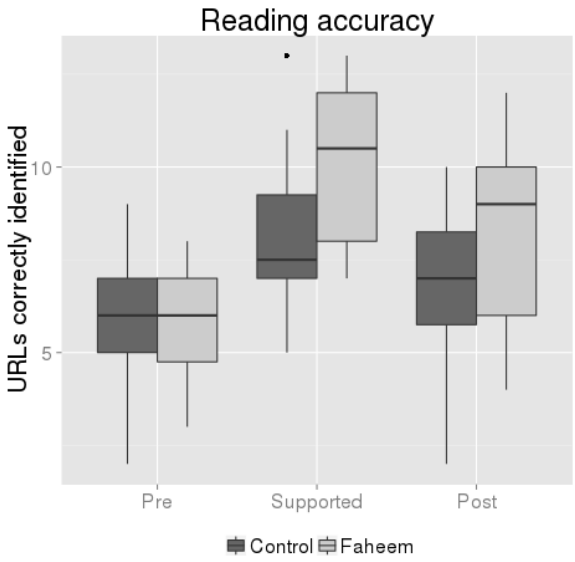


Figure 5: Boxplot of the number of URLs participants correctly identified as phishing for conditions in the pre test, when supported by the tool, and post test.

RESULTS

Our primary concern is if Faheem helped participants to accurately determine if a given URL lead to a particular company or not, both while using Faheem and after. As can be seen in Figure 5, participants were able to use both Faheem and URL Explainer to improve their ability to identify potential phishing URLs with more accuracy than they could without the tools. To determine if the Faheem group experienced a significant improvement compared to URL Explainer, we computed the per-participant change between supported and pre to account for initial skill variation. Then, we ran an independent t-test with an α of .05. We found that the Faheem group (M=4.55) showed statistically significantly more improvement than the control group (M=2.15), ($p<0.003$), and fairly large effect size (r=0.47).

We also looked at the difference between the pre and post tests. Similar to the prior analysis, we computed the change per participant and then compared using an independent t-test. Faheem (M=2.75) still showed a statistically significant improvement ($p< a0.044$) with (r=0.32) as compared to the control (M=1.05). Though the difference between the conditions narrowed after support was removed.

URL category	Pre		Supported		Post	
	Control	Faheem	Control	Faheem	Control	Faheem
Standard-URLs	43%	28%	83%	70%	58%	62%
IP-Based	80%	85%	70%	85%	85%	95%
Shortened links	15%	15%	30%	35%	_	_
Redirects	25%	60%	65%	100%	_	_
Misspelling	40%	48%	53%	100%	48%	88%
Multi-level domain	50%	40%	50%	80%	49%	50%
Company name not in host position	60%	58%	58%	75%	65%	70%
Non-ASCII characters	55%	55%	45%	100%	45%	90%

Table 1: URL identification accuracy for each condition, stage and type of URL issue presented. Participants were given an organization name and asked if the URL went to that organization or if it was likely phishing. So the top left value should be read as 43% of the standard URLs presented to the control group in the pre-test were correctly identified as the company or phishing.

As a reminder, participants were provided with a company name and asked if the URL lead to that company or if it was likely phishing. Table 1 shows the results of the question for the different conditions, stages, and types of URL manipulations. The pre-test results show that participants in both conditions achieved the lowest scores for standard, shortened, redirects and misspelt URLs.

For the supported stage, both groups scored lower for the shortened link *https://bit.ly/18AOiDE* which redirects to *https://www.facebook.com/unsupportedbrowser*. Participants' justifications were different, with one of them stating "*Bitly* always sends me to advertisement website", and others stating that the link goes to Facebook but 'unsupported browser' in the link is suspicious. Both URL Explainer and Faheem resolve shortened URLs, like the Bitly example above, and tell the user the ultimate destination of the URL. Participants in the supported stage clearly did not understand the feature or it failed to overcome their previous biases as they still do quite poorly at identifying phishing sites. One potential explanation is technical. Both Faheem and URL Explainer make a headless request to resolve the URL server-side. Doing so can trigger behaviours in the host server. In the above example, it caused Facebook to serve back its "unsupported browser" page rather than the actual content, which was then reflected in the two tools.

After using the tools, participants, in both groups, were seen to experience problems when the links containing top-level domains other than `.com`, such as *tagesschau.de*. The Faheem participants who answered this question correctly said that they Googled the domain, suggesting that the Faheem group did benefit from the provided advice. The top-level domain `.de` is the country code top-level domain for the Federal Republic of Germany. Another URL was *https://translate.google.co.uk/*. Participants who are not from the UK did not trust it with the justification provided was that they had never seen a Google website with these characters.

Moreover, the Faheem group of participants were confused between the URL and the recognized brand name for the organization, such as New York Times (*www.nytimes.com/*) whereas the other group's performance was found to be higher because the other tool provided a webpage title containing the full website name.

CONCLUSION AND FUTURE WORK

In this work, we have presented Faheem, a Slack bot which helps users learn about URLs in an interactive format. Faheem assists users who have no understanding of URLs in identifying common URL elements and well known malicious URL tricks. It also assists more experienced URL readers in identifying less user-visible tricks such as non-ASCII letters which are visibly identical to ASCII ones.

To test Faheem we compared it with URL Explainer, a simplistic web page which parses a URL for a user but focuses on a factual clear representation of the URL contents rather than helping the user identify common issues. We find that while using both tools, Faheem is better at helping a user identify URLs which have a destination other than where the user wishes to go. Additionally, we also saw some minimal learning effects with Faheem users showing an improved ability to identify phishing URLs after using the tool.

In conclusion, Faheem is a novel approach to helping users understand the contents of URLs. Our study shows that the approach has some promise, though more comprehensive studies are needed to conclusively determine the effectiveness of Faheem-type solutions.

REFERENCES

1. 2009-2018. MetaCert Security. (2009-2018). `https://slacksecurity.metacert.com/`
2. 2017. *2017 Data Breach Investigations Report*. Technical Report. Verizon.
3. 2017. *State of the Phish 2017*. Technical Report. Wombat security technologies.
4. Lawrence W Barsalou. 1999. Language comprehension: Archival memory or preparation for situated action? (1999).
5. T. Berners-Lee, L. Masinter, and M. McCahill. 1994. RFC1738: Uniform Resource Locators (URL). (December 1994). `https://www.w3.org/Addressing/rfc1738.txt`
6. Mark Button, David Shepherd, Dean Blackbourn, and Martin Tunley. 2016. *Annual Fraud Indicators 2016*. Technical Report. University of Portsmouth Center for Counter Fraud Studies.

7. Gamze Canova, Melanie Volkamer, Clemens Bergmann, and Roland Borza. 2014. NoPhish: an anti-phishing education app. In *International Workshop on Security and Trust Management*. Springer, 188–192.
8. Gamze Canova, Melanie Volkamer, Clemens Bergmann, and Benjamin Reinheimer. 2015. NoPhish app evaluation: lab and retention study. In *NDSS workshop on usable security*.
9. APWG Internet Policy Committee and others. 2013. Global phishing survey: Trends and domain name use in 2h2013. (2013).
10. Nicola Davinson and Elizabeth Sillence. 2010. It won't happen to me: Promoting secure behaviour among internet users. *Computers in Human Behavior* 26, 6 (2010), 1739–1747.
11. Rachna Dhamija, J Doug Tygar, and Marti Hearst. 2006. Why phishing works. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 581–590.
12. Serge Egelman, Lorrie Faith Cranor, and Jason Hong. 2008. You've been warned: an empirical study of the effectiveness of web browser phishing warnings. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1065–1074.
13. Nina Eyrich, Monica L Padman, and Kaye D Sweetser. 2008. PR practitioners' use of social media tools and communication technology. *Public relations review* 34, 4 (2008), 412–414.
14. Sujata Garera, Niels Provos, Monica Chew, and Aviel D. Rubin. 2007. A framework for detection and measurement of phishing attacks. In *Proceedings of the 2007 ACM workshop on Recurring malcode - WORM '07*. ACM Press, New York, New York, USA, 1. DOI: http://dx.doi.org/10.1145/1314389.1314391
15. Cormac Herley. 2009. So Long, And No Thanks for the Externalities: The rational rejection of security advice by users. In *Proceedings of NSPW'09*.
16. Oliver J Hunt and Ivan Krstic. 2017. Preventing URL confusion attacks. (March 21 2017). US Patent 9,602,520.
17. Mahmoud Khonji, Youssef Iraqi, and Andrew Jones. 2013. Phishing detection: a literature survey. *IEEE Communications Surveys & Tutorials* 15, 4 (2013), 2091–2121.
18. Iacovos Kirlappos and M Angela Sasse. 2012. Security education against phishing: A modest proposal for a major rethink. *IEEE Security & Privacy* 10, 2 (2012), 24–32.
19. Ponnurangam Kumaraguru, Steve Sheng, Alessandro Acquisti, Lorrie Faith Cranor, and Jason Hong. 2008. Lessons From a Real World Evaluation of Anti-Phishing Training. *e-Crime Researchers Summit, Anti-Phishing Working Group* (October 2008). http://precog.iiitd.edu.in/Publications_files/eCrime_APWG_08.pdf
20. Alexandra Kunz, Melanie Volkamer, Simon Stockhardt, Sven Palberg, Tessa Lottermann, and Eric Piegert. 2016. Nophish: evaluation of a web application that teaches people being aware of phishing attacks.. In *GI-Jahrestagung*. 509–518.
21. Eric Lin, Saul Greenberg, Eileah Trotter, David Ma, and John Aycock. 2011. Does domain highlighting help people identify phishing sites?. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2075–2084.
22. Cooper Quintin. 2015. HealthCare.gov Sends Personal Data to Dozens of Tracking Websites. (20 January 2015). https://www.eff.org/deeplinks/2015/01/healthcare.gov-sends-personal-data
23. R. Reeder, I. Ion, and S. Consolvo. 2017. 152 Simple Steps to Stay Safe Online: Security Advice for Non-tech-savvy Users. *IEEE Security Privacy* PP, 99 (2017), 1–1. DOI: http://dx.doi.org/10.1109/MSP.2017.265093101
24. Bethany Rittle-Johnson and Kenneth R Koedinger. 2002. Comparing Instructional Strategies for Integrating Conceptual and Procedural Knowledge. (2002).
25. Steve Sheng, Bryant Magnien, Ponnurangam Kumaraguru, Alessandro Acquisti, Lorrie Faith Cranor, Jason Hong, and Elizabeth Nunge. 2007. Anti-Phishing Phil: the design and evaluation of a game that teaches people not to fall for phish. In *Proceedings of the 3rd symposium on Usable privacy and security*. ACM, 88–99.
26. Kurt Thomas, Chris Grier, Justin Ma, Vern Paxson, and Dawn Song. 2011. Design and evaluation of a real-time URL spam filtering service. In *Security and Privacy (SP), 2011 IEEE Symposium on*. IEEE, 447–462.
27. Melanie Volkamer, Karen Renaud, Karen Renaud, Paul Gerber, and Paul Gerber. 2016. Spot the phish by checking the pruned URL. *Information & Computer Security* 24, 4 (2016), 372–385.
28. Andrew G West and Adam J Aviv. 2014. On the Privacy Concerns of URL Query Strings. (2014).