Spamalot: A Toolkit for Consuming Spammers' Resources

Peter C. Nelson, Kenneth P. Dallmeyer, Lukasz M. Szybalski, Tom P. Palarz, Michael Wieher
Artificial Intelligence Laboratory
Department of Computer Science
University of Illinois at Chicago
nelson@uic.edu, kdallmey@cs.uic.edu, szybalski@gmail.com, tpalar1@uic.edu, michael.wieher@gmail.com

ABSTRACT
The Spamalot system uses intelligent agents to interact with spam messages and systems referenced in spam. The goal of Spamalot is to consume spam senders' resources by engaging the spammer in an unproductive conversation or information exchange. To date two Spamalot agents have been implemented: *Arthur* which handles Nigerian spam and *Patsy* which processes spam requesting information via web forms.

1. INTRODUCTION
The primary reason why spam is profitable is that spammers send can messages for very little cost, with respect to both computing and human labor. The Spamalot project began as a technique, which we refer to as *duping*, which attacks the senders of spam. The basic idea of this technique is to pose as a *dupe* by responding to spam, forcing spammers to spend time pursuing a false lead or *dupe*. As a test of duping, in 2004 we started responding to Nigerian spam of the form "Dear Sir, Please help me transfer millions from my third world country and you will receive 25% of the proceeds." Instead of spending 2-3 seconds to delete the messages that made it through our various spam filters, we spent a few additional seconds sending a reply such as, "I am very interested, please send details." Spammers excitedly responded to such messages attempting to perpetrate their fraud. With each reply from the spammer we responded with another short message of the sort, "Yes I am very interested, please call me" along with an office telephone number or a fax number with a suggested calling time outside of working hours. The chain of communications is easily continued and at times generated more than 50 message exchanges and numerous telephone calls with a single spammer. While slightly more time consuming, duping also works for other types of spam. For example we have followed spam links to mortgage refinance web sites and entered data (e.g., excellent credit and looking to refinance a 9% mortgage) that has resulted in approximately 25 mortgage broker return calls over a few days.

The original idea of this project is that spam could be greatly reduced if we could encourage the public to have a different sociological response to spam. Spamming schemes such as the Nigerian 419 bank scam and even phishing become ineffective if spammers are flooded with dupes. After some exploration, we have decided that it is not realistic to rely on a change in human behavior to bring this idea forward. The project has now shifted to creating an artificial intelligence toolkit Spamalot to carry out this behavior. Spamalot is an intelligent agent paradigm whose sole purpose is to consume as much of the spammers' resources as possible. For spam that makes it through a spam filter, rather than clicking a *junk button*, a mail user can click a *Spamalot button* on his/her mail toolbar to have the message handled by Spamalot's agents. To date we have built two Spamalot agents: *Arthur*, the agent for handling Nigerian spam, and *Patsy*, the agent for handling web form spam such as mortgage brokers.

Section 2 very briefly describes related work. Section 3 overviews the basic architecture of Spamalot and its two existing agents *Arthur* and *Patsy*. Preliminary Spamalot experimental results are presented in section 4. Section 5 presents conclusions and future work.

2. RELATED WORK
The Spamalot system builds upon work used to classify different types of emails. Much work has been done in classifying spam versus non-spam email. Trudeau et al. [8] overviews different techniques of classifying email. Carvalho et al. [1] classify emails as a speech act, such as a request or question. Another similar concept put forth by Martin et al. [5] is to classify emails based on behavioral features using statistical learning. Likewise Dredze et al. [2] uses an algorithm to classify emails into activities. Our Spamalot system relies on classification not only to determine spam versus non-spam, but also to classify the type of spam to select an appropriate agent.

Other researchers have developed approaches and proposals that require sender resources to be consumed making spamming less feasible. Oudot [6] suggests creating a "honeypot" of fake proxys as a way to detect, slow, and block spammers. Goodman et al. [3] advocate that the best way to reduce spam is to add a cost to sending emails. They believe that adding a Turing test every X amount of emails would require the spammer to spend time proving that it was human. This lessens the capability and feasibility of using an automated way to spam. Another aspect of the paper also advocates adding a cost to sending an email. There could be an actual monetary cost or there could be a computational cost, say factor two large prime numbers. Johansson et al. [4] have developed a system called CAMRAM to do exactly that. Blue Security [7] attempted to consume spamming resources by having all users of its software automatically and repeatedly send out messages to spammers and their ISPs. Our approach differs from most of these methods in that we are directly targeting the consumption of human spamming resources rather than machine resources, and our penalty occurs after the spam has been sent.

3. METHOD
The Spamalot architecture is shown in Figure 1. It is used in conjunction with a traditional spam filter. When an email comes in it is either classified as spam or as normal email by the email client's spam filter. The spam then can be deleted

from the system or be passed to Spamalot. Sometimes spam gets past the filter. In that case the user may pass the spam directly to Spamalot.

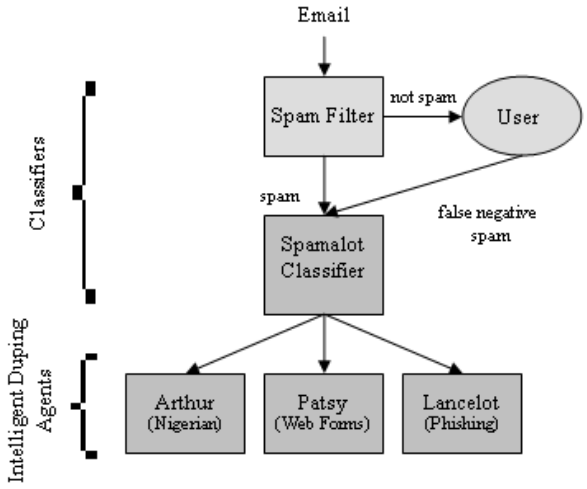

Figure 1 – Spamalot Architecture

The next step is for the Spamalot classifier to decide which intelligent duping agent to have handle the spam. If the spam is classified as a Nigerian 419 spam, then it is passed along to *Arthur*. If it is a web form or a link to a web form, then the spam is passed to *Patsy*. We have prototype versions of both *Arthur* and *Patsy*. A third agent under development is *Lancelot* which will handle phishing spams. In the future we will develop additional duping agents as necessary.

The *Arthur* agent is designed to interact with the Nigerian spammer as if a real person is responding to the spam. In order to do so, the agent first parses the email. The initial spam email has most of the information needed to get a conversation started. To identify the spammer the agent will use its email address. Next an initial *Arthur* response is sent to the spammer. *Arthur* responses are stored in a database and are stored by categories such as: *who are you*, *am interested*, *money issue*, *phone me*, etc. There are multiple messages per category type. The agent tracks what type of response categories it has sent out to limit duplicate messages. *Arthur* will continue to return messages to the spammer as long as the spammer keeps replying. In practice we have found even random messages (i.e., "How is the weather in your city?" or "Do you have a family?") work well in prolonging the dialog. We have set up email addresses, voicemail boxes, and a logging database to collect data.

The *Patsy* agent is designed to fill out spam web forms of the sort received from mortgage brokers and online universities. These forms request information to pursue a future sale. As with *Arthur,* the purpose of *Patsy* is to receive a response from the spammer. After following the links to a form, *Patsy* will parse the form to generate inputs. The form will be filled out to maximize interest from the spammers (e.g., data suggesting an excellent credit rating). For *Patsy* we have also set up email accounts and voicemail to track spammer interactions.

Lancelot is currently under development and is targeted to phishing spammers. The initial *Lancelot* strategy will be to flood the phishing site with long and complex user names and passwords. The result will be something close to a denial of service and will fill the phishing database with massive amounts of false usernames and passwords. There are additional agents that will also need to be developed for sites such as pharmaceutical spams.

4. PRELIMINARY RESULTS

Prior to testing we were required to receive approval from our Institutional Research Board for human subject experimentation. IRB approval was recently received and we have performed limited experimentation with both *Arthur* and *Patsy*.

Arthur's performance is more easily measured than *Patsy's* performance. To date we have seeded *Arthur* with twelve Nigerian spam messages, which have resulted in seven threads of communication. The average length of conversation with the spammers was six messages, with the longest being fifteen. The early results suggest Arthur is effective in generating a continuing stream of dialog with spammers.

Results for the *Patsy* agent are more difficult to quantify as the typical response from spammers is a telephone call. *Patsy* has successfully filled out eight unique forms, with each form being filled out multiple times. These actions have generated telephone calls and messages to our automated voicemail boxes. We also know from experience that callers often do not leave messages and instead will just call again. Our current setup does not allow us to track calls when no messages are left. Over the past two months we have received over 100 calls from Patsy's efforts.

Additionally since beginning our experiments a few months ago we have already received over seventy-five additional spam emails to our email accounts, suggesting that our email has been passed around in spam circles.

More detailed experimental results can be found at http://www.rites.uic.edu/projects.html.

5. CONCLUSIONS

We believe the Spamalot approach is novel and shows promise. We view the Spamalot approach as complementary to existing filtering techniques, giving mail users the option of employing Spamalot on spam messages that make it through their server or client filters. A more aggressive strategy would be to also apply Spamalot agents to automatically classified spam. In such cases the confidence level that the target is spam would need to be high.

Though we would like to eventually develop commercial grade tools that can be mass distributed, there are other potential uses for Spamalot in combating spam. Spamalot dialog could be posted online with password limited access to allow ISP and email service providers to shut down websites and email addresses. Similarly Spamalot can provide logs of its exchanges with phishers to financial institutions which can then take appropriate action when they detect fraudulent login attempts with Spamalot generated login data.

6. REFERENCES

[1] Carvalho, Vitor R. and Cohen, William W. On The Collective Classification of Email "Speech Acts". *Special Interest Group on Information Retrieval*. 2005

[2] Drdze, Mark, Lau, Tessa, and Kushmerick, Nicholas. Automatically Classifying Emails into Activities. *International Conference On Intelligent User Interfaces*. 2006.

[3] Goodman, Joshua and Rounthwaite, Robert. Stopping Outgoing Spam. *ACM Conference on Electronic Commerce '04*. 2004.

[4] Johansson, E. S. CAMRAM. Available at http://www.camram.org.

[5] Martin, Steve, Sewani, Anil, Nelson, Blaine, Chen, Karl, and Joseph, Anthony D. Analyzing Behavior Features for Email Classification. *Conference on Email and Anti-Spam*. 2005

[6] Oudet, Laurent. Fighting Spam With Honeypots. November 26, 2003. http://www.securityfocus.com/infocus/1747.

[7] Spring, Tom. Bringing Spammers to Their Knees. *PC World*. July 18, 2005. http://www.pcworld.com/news/article/0,aid,121841,00.asp

[8] Trudeau, Paris, Cullen, Richard, and Zwieback, Dave. Major Techniques for Classifying Spam. *SurfControl*, 2003.