DETECTING PHISHING ATTACKS IN PURCHASING PROCESS THROUGH PROACTIVE APPROACH

S.Arun, D.Anandan, T.Selvaprabhu, B.Sivakumar, P.Revathi, H.Shine
Department of Information Technology
Veltech Multitech Dr.Rangarajan Dr.Sakunthala Engineering College/Anna University
Chennai, India
Email: {arun14589, anandandk, sivablack, rockselva, revathipugazhe, shinehenry}@gmail.com

ABSTRACT

A Monitor is a software system that observes and analyzes the behavior of target system determining the quality of interest such as satisfaction of the target system. In the modern technology business processes are open and distributed which may lead to failure. Therefore monitoring is an important task for the services that comprise these processes. We are going to present a framework for multilevel monitoring of these service systems. The main objective of this project is monitoring the customer who purchases items from Merchant. Phishing is an online scam that attempts to defraud people of their personal information such as credit card or bank account information. We are going to detect, locate and remove the phishing E-mail. The customer details will be stored in web registry. We are going to demonstrate how the online business processes can be implemented with multiple scenarios that include monitoring open service policy commitments.

KEYWORDS

Monitoring, Phishing, Commitments, Scam, Processes.

1. INTRODUCTION

The Internet is now a popular means for providing entertainment, communicating with friends, conducting e-commerce, and delivering teaching materials. However, some people around the globe are taking advantage of the anonymity provided by the Internet to fool individuals with fake offers, or by misrepresenting themselves as legitimate companies. Phishing is the online scam that attempts to defraud people of their personal information such as credit card or bank account information, and username and password credentials. The online criminals are known as Phishers. Conventionally, mass E-mailing with a phishing link is the most popular way to lure the victims. However, SMS messages, chat rooms, fake add banners, fake job offers, and fake browser tools have emerged as a new platform among Phishers. Researchers have proposed techniques to prevent phishing attacks, Phishers are becoming increasingly sophisticated in their approaches. Phishing attacks often involve rigorous planning and incorporate strategies to bypass existing anti-phishing tools. The shear volume of phishing attacks suggests that existing anti-phishing tools are insufficient. This is primarily due to fact that they only take a reactive or passive approach to stemming the problem. That is, they only filter suspect emails, but don't actually do anything to shut down the problem at its source. This

DOI : 10.5121/acij.2012.3309

paper proposes a proactive approach to remove a phishing page from the host server. Rather than just filtering email and flagging suspect messages as 'spam', our approach actively seeks out Phishers in an attempt to disconnect them at the source.

The presence of phishing page is alerted initially upon receiving the Phisher's solicitation e-mail. Then the IP address, contact information of the host server is retrieved by the system using a tracking program. Next, the system sends notification about the phishing page to the administrator of that server. Finally, it's the responsibility of the administrator to remove the phishing page from its server, else the administrator have to face the possibility of criminals continuing to use its site. This approach acts as the basis for further development into proactively (or aggressively) attacking Phishers back, rather than being a reactionary approach that is common to most email filters and anti-virus software. Service-oriented architectures and associated interoperability standards provide key enablers for these service systems. As the business processes are open and distributed processes, the tasks that are performed by service were not centrally controlled, and hence the result is unpredictable. As a result, service outcomes themselves tend to be uncertain. Service monitoring, therefore, remains a significant challenge. The main goal of this research is to develop the detecting methods for monitoring of purchasing process. The key contribution of this paper is the introduction of ontology of communicative acts into these abstraction layers to enhance policy specification and monitoring of service systems. We finally develop this contribution in monitoring of service systems, establishing its feasibility, and going to demonstrate the online purchasing process with multiple scenarios.

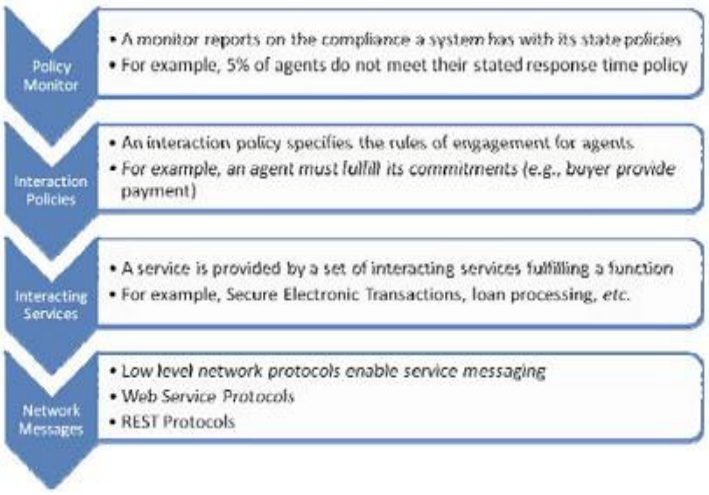

Figure 1. Abstraction layers for monitoring service systems.

2. RELATED WORK

This section provides background on the phishing process, the various strategies employed by Phishers, and the style of phishing attack considered by this paper. It also presents the existing mechanisms that are currently being used to combat phishing. Generally, most phishing attacks begin with spam. Spam is mass unsolicited email. The email message typically contains some sort of socially engineered message enticing the recipient to venture to a web site or to reply to the message. It is usually at this point phishing attacks start to differ in their approach. In this paper, we will primarily be concentrating on phishing attacks that attempt to lure a recipient to a website by providing a link within an email. Upon reaching the website, the user is either

asked to enter personal details as they believe it to be a legitimate company (such as his/her bank), or the user is conned into believing that s/he must install a critical update for his/her computer (which is in fact a virus). A variation on this style of phishing attack is for the victim to reply directly to the Phisher's email address rather than following a link to a website. This style of attack will not be considered in this paper, but will be the focus of future work.

The majority of the anti-phishing tools use an email filtering process to separate legitimate emails from suspected spam in the inbox. It is then up to the individual to decide whether to discard the message. If an individual doesn't have the latest anti-phishing tools installed, or has failed to install the most recent update for his/her anti-phishing program, then they lose this layer of protection. We refer to this as a passive anti-phishing approach. This is because the approach only attempts to locally protect an individual from a phishing attack, but does not actively make any effort to remove or shut down the Phisher at the source. In effect, the Phisher is free to continue with his/her operation and can potentially accrue further victims. There are several spam filters, browser tools, anti-spyware and anti-virus software available to protect online computers from various attacks. However, there were very few research efforts have been entirely focused to protect online users from phishing attacks in the past. Existing anti-phishing and anti-spam techniques suffer from one or more limitations and they are not 100% effective at stopping all spam and phishing attacks. Phishers are able to find ways to bypass existing rule-based and statistical-based filters without much difficulty. Major e-mail service providers such as Yahoo, Hotmail, Gmail, and AOL filter all incoming emails separating them into Inbox (legitimate email) and junk (illegitimate email) email folders. However, these e-mail service providers do not actually attempt to remove the phishing page associated with the illegitimate email. Furthermore, Phishers have readily available tools to bypass such spam filters.

There have been efforts made to compare performance of various machine learning techniques such as fuzzy logic and neural network theory to detect phishing emails. However, these attempts still require improvement to achieve a higher accuracy rate. Many researchers have attempted to detect the structure, properties and technical subterfuge of the typical phishing emails in order to design more effective anti-phishing tools. The ultimate problem with only using detection as a defense is that the final decision rests with the user as to whether s/he should access a website or not. The extremely convincing nature of phishing emails makes this a dangerous approach for the occasional or non-technical Internet user. Other defensive techniques involve the use of Secure Sockets Layer (SSL), digital signatures, and digital certificates. The security of information is very important where the confidential data transferred on public Internet such as online shopping, banking transactions, government and corporate email communications, etc. SSL, digital certificates and digital signatures provide a level of information security while data travels across the public Internet. While such cryptographic techniques are quitereliable and robust in mathematical terms, however they suffer from weak implementation or incorrect use of technique.

People are familiar with the SSL icon and padlock on the browser and they believe that the communication is secure. However, phishers can exploit this perceived protection by using fake SSL padlock images on their phishing pages to create confidence and lure Internet users. Some Phishers also use a low-quality certificate or trial certificate on their phishing pages. Furthermore, there are technical subterfuges mechanisms in which malware can suppress any security errors and create false security indicators. Another popular approach for anti-phishing techniques is flooding the

Phisher's database with fake information. This approach significantly minimizes the probability of distinguishing the correct data from the flooded database in order to protect people who have submitted their personal credentials already. However, this approach does not prevent other Internet users from supplying their personal credentials to the phishing website. Raising awareness through training and enforcing policies for suspect emails is also popular approach among corporations and institutions for preventing the damage caused by phishers. However, researchers have found that best phishing sites have fooled 90% of the people during their experiment on various groups of people including academic staff and students at a prestigious American university.

3. SERVICE INTERACTION PROTOCOL

The researchers suggested the idea of conversations in the form of service interactions. An interaction protocol may be defined as the rules of engagement among the participants who are interacted. The service interaction protocol includes the possible set of actions that each participant may perform and the order in which these must be performed. Concerns that need resolution for the design of interaction protocols, therefore, include: 1) Specificity; 2) Semantic content; and 3)Composability. The first concern deals with specificity-abstraction dimension. For example, an abstract specification may represent interactions among services) that is domain-independent. On the other hand, a specific protocol, e.g., for shipment, may be domain-dependent. The second concern deals with semantic content that includes the nature of the interactions. For example, a protocol specification may rely on message content (e.g., destination, payment info) by using the request-reply mechanism to capture content. Whereas the message itself may capture the semantics to distinguish actions that include direct, inform, cancel, and others. The protocol specification can be illustrated on a common business process for purchasing, drawn from the Secure Electronic Transactions (SET) Standard. Four main roles collaborate to carry out this process:

1. the Customer, who wants to purchase items,
2. the Merchant, who sells items,
3. the Shipper, who transports the physical items to the Customer, and
4. the Payment Gateway, who authorizes payments.

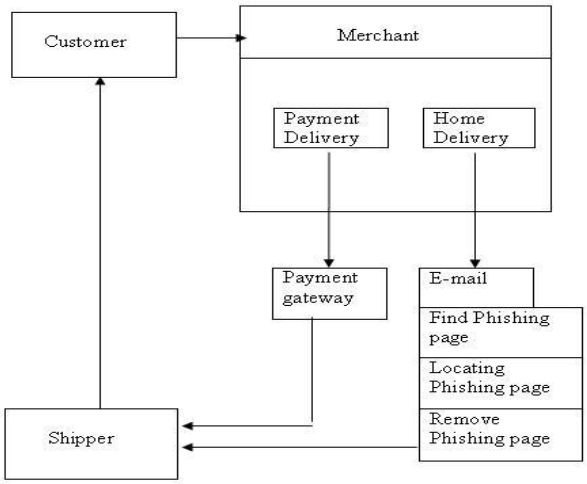

Figure 2. A purchasing process

4. PROTOCOL POLICIES AND COMMITMENTS

A protocol policy specifies outcomes that satisfy the protocol goals. For example, a policy for the Payment protocol may be: "after Payer sends paymentInfo, eventually Payer receives receipt". Protocol policies can also be specified in non-protocol terms. For example, one can conceptualize protocols as manipulation of commitments, e.g.,creation, cancellation, fulfillment,etc. A commitment based Payment protocol policy may be: "after Payer commits to Order, Payer pays for Order" (thereby, discharging the commitment). A commitment captures a contractual relationship that enables "manipulations, such as delegation and assignment, which are essential for open systems". In the seller process, the seller receives itemID, then consults its policy for quoting and sends the quote price i.e., itemPrice and receives either accept-Quote or reject-Quote. Their approach to specifying protocol policies has these characteristics:

1. Commitments are associated with protocol specifications.
2. Commitment ordering is not directly specified. For example, the commitment on the quote operation simply indicates that the buyer and seller are mutually committed based on two dependent conditions (pay and deliver).
3. Commitments are referenced only via the concrete operation with which they are associated.
4. Reasoning is supported only at design time.

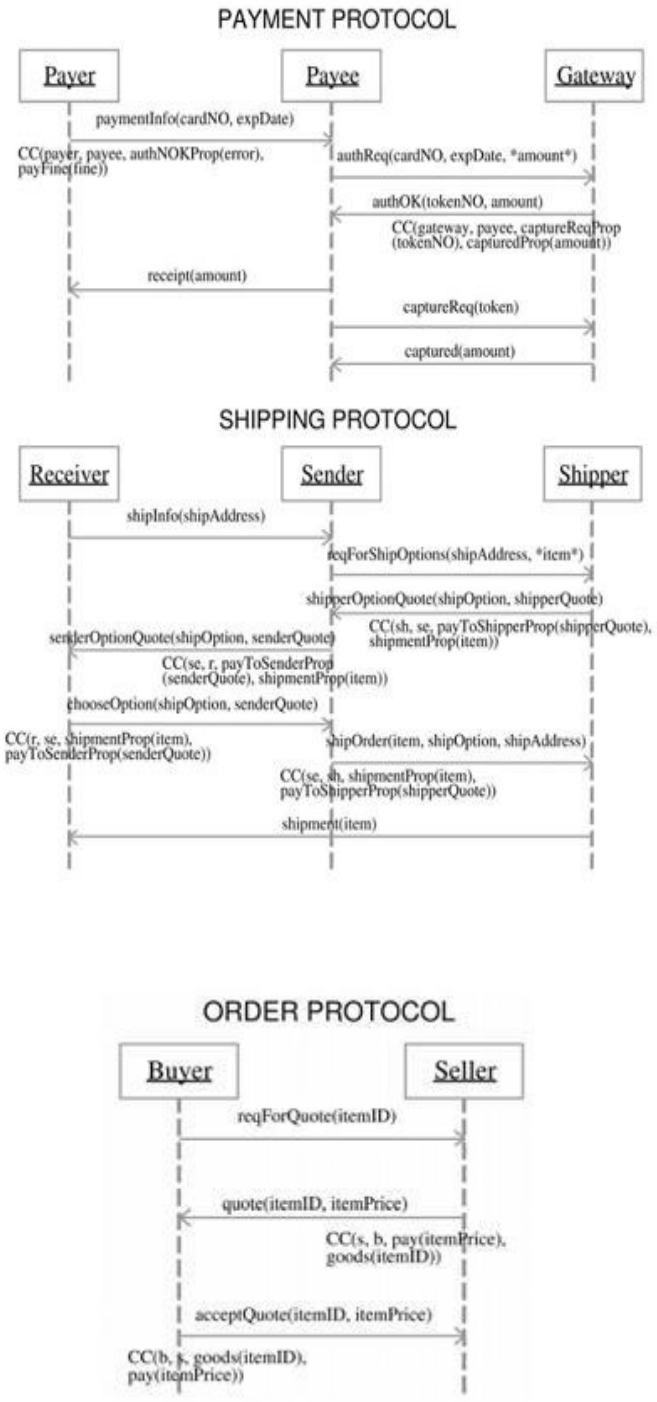

Figure 3. Protocols for the purchase process

5. METHODOLOGY AND ARCHITECTURE

The SERMON (A monitoring framework for service systems) methodology uses a tailored approach for specifying service interaction goals. These include: 1) agent service goals and 2) agent protocol goals. Agent service goals describe desired characteristics of an agent's behavior during the execution of a protocol, for example, "provide timely response to a request". Agent protocol goals describe desired characteristics of an agent's behavior as it manages multiple protocols simultaneously, for example, "abandon a protocol execution that may not result in a commitment". Based on the goal model, monitors are compiled and the monitoring system is deployed. The monitor updates the runtime status of the goals during the execution of target software system.

As events arrive, they are stored in the repository, which is analyzed for goal satisfaction. The results may be presented on a dash board or they may guide active responses, such as selecting alternative services. The service monitoring repository stores event data as well as analysis results. It consists of five layers.

- Level 0(events): The lowest layer stores raw data such as, "agent x receives message m from agent y".
- Level 1(ontology): The next layer stores an interpretation of the event data using terms form a selected ontology.
- Level 2(Transaction Properties): Properties of level 1 events are stored at layer 2, for example, the property "agent x never violates a commitment" is true.
- Level 3(Metaproperties): Properties about properties are stored at layer 3, for example, "property p1 has a 75 percent success to failure ratio over the last 72 hours".

The SERMON monitors both payment delivery and Home delivery of the purchasing process. In case of payment delivery, the customer selects an item and shipping option. The Merchant will check for the feasibility from shipper. Then the customer will select the payment options and he will be redirected to payment gateway. After the transaction is committed, the receipt will be displayed and E-mail will be sent to customer inbox. Then the merchant will request the shipper to deliver the product to the customer and finally the product will be delivered to the customer. In case of home delivery, the customer will select an item and shipping option. Then the details regarding delivery of the product will be sent to customer inbox through E-mail. Suppose if a Hacker sends an E-mail indicating the customer to enter the details of the credit card information or bank account information. The customer without knowledge can enter the details which will be stored in the Hacker's database. Therefore it is an important concern to detect the Phishing E-mail.

5.1 Detecting a Phishing Email:

This module finds the Phishing page from our inbox email or spam mail. We follow some terms to find the phishing pages, such as subject heading an unrealistically lucrative offer, a request for personal details, redirecting the URL. We also notice that phishers have spoofed sender's email address and ask the victims not to reply to the email. Hence we use the anatomy of phishing mail to find the phishing page. The majority of the phishing emails we found were targeted at leading financial institutions. We also noticed that phishers have spoofed sender's email address and

ask the victims not to reply to the email. Instead they have requested users to click on the link sent within the email. The Detection of Phishing E-mail is based on the following:

5.1.1 Collecting Information in the Email

To gather information the Phishing emails used HTML forms within the E-mail. This technique was being used in some of today's scams. Once the information has been entered by the customer, the email must provide a method of sending the information to the fraudster. Generally the "Submit" button that was present in the form causes the information to be sent to the hacker's specified location.

5.1.2 Links to Web Sites That Gather Information

Most of the phishing emails includes a link that sends the recipient to a Web site instead of using forms within the E-mail. Some hackers register domain names which are similar to those owned by a reputable company.

5.1.3 Using the @ Symbol to Confuse

When the @ symbol is used in an "http://" or "https://" URL, all text that was entered before the @ symbol is ignored and the browser references only the information that has been entered after the @ symbol. This trick was being used by hackers for fooling the person viewing the email code into thinking the link is going to the site listed before the @ symbol, while it actually links to the hackers site after the @ symbol. However, this text before the @ symbol is ignored and the link sends the user to "210.93.131.250/my/index.htm".

5.1.4 Redirecting the URL

A URL can be further obscured by using a redirection service. For example, cjb.net and tinyurl.com provide redirection services that assign the user an alias for the user's specified URL. For example, a URL such as "http://tinyurl.com/3" is provided by tinyurl.com when the user enters a URL into the site. When a redirection service is used, the provided link sends the user to the service site and the service site then forwards the user to the intended site. This service is useful for replacing long URLs, but unfortunately it can be abused by fraudsters because it hides the true destination of the link.

Some fraudsters have even gone to the effort to redirect their URL twice. The link "http://r.aol.com/cgi/redir?http://jne9rrfj4.CjB.neT/?uudzQYRgY1GNEn" was found in a fraudulent Iobbank email message which shows a double-redirect. First the browser is sent to "http://r.aol.com/cgi."

Then the browser is redirected to "http://jne9rrfj4.CjB.neT/?uudzQYRgY1GNEn," which an alias is provided by cjb.net. Finally cjb.net redirects the browser a second time to the intended Web page (the actual URL is stored at cjb.net and is accessed through the cjb.net alias).

5.1.5 Using Hexadecimal Character Codes

Hackers can also hide the URLs by using hexadecimal character codes to represent the numbers in the IP address. Each hexadecimal character code begins with "%." This next example combines a few of the fraud tricks mentioned above:

http://www.visa.com%00@%32%30%30%2E%38%38%2E%38%31%34%2E%32%31%33
The URL is put in <userinfo><null>@<host> format. On computers using Microsoft Internet Explorer that have not installed the patch, only the www.visa.com is displayed in the address bar but the browser window displays the site at "%32%32%30%2E%36 %38%2E%32%31%34 2E%32%31%33," which is the fraudulent Web site's IP address hidden in hexadecimal character code.

5.1.6 Switching Ports

Web pages are accessed on servers through ports. A port is entered by following the URL with a colon and the port number. If port is not specified, the browser uses port 80, the default port number for Web pages. Scammers occasionally use other ports to hide their location. In the following example, the IP address after the @ symbol is followed by ":8034", which represents the specified port of 8034.

http://www.citibankonline.com:ac-KTtF4BD6y4TZlcv6GT5D@64.29.173.91:8034/

Some scammers even hack into a legitimate company's server and host their Web site on the server using a higher numbered port. The legitimate company may be completely unaware of the fraudulent site.

5.1.7 Hiding the Host Information

Links in emails using the <userinfo>@<host> format discussed sometimes take the trick a step further by inserting a null or other unprintable character before the @ symbol, which prevents the host information from being displayed in the address bar of the browser. Web browsers generally display the URL information for the current Web page in the address bar.
However, if the <userinfo><null>@<host> format is used in the link in the email, some versions of Microsoft Internet Explorer will not display the host information. [12] For example, if a fraudster uses the format <userinfo><null>@<host>, the <userinfo> is displayed in the browser address bar in Microsoft Internet Explorer and the <host> information is concealed. Using the same example given above:

http://cgi1.ebay.com.aw-cgiebayISAPI.dll%00@210.93.131.250/index.htm

The character represented by "%00" causes only the userinfo "http://cgi1.ebay.com.aw-cgiebayISAPI.dll" to be displayed in the browser address bar, but the Web page is actually accessed by the host information, "210.93.131.250/my/index.htm."

5.1.8 Using the IP Address

This domain name clearly shows the true destination of the link: "www.membership.com." Hackers frequently attempt to conceal the destination Web site by obscuring the URL. One method to conceal the destination is to use the IP address rather than the using the hostname. An example of an IP address used in a hackers email message is "http://210.16.224.36/sr/." An IP address can also be used further by expressing it in Dword, Octal, or Hexadecimal format.

5.1.9 Link Text in Email Differs From Link Destination

In fraudulent email, the link that was present in the email is usually different than the actual destination. For example, the email looks as though it is going to send the user to "http://account-registration.com," but instead sends the user to "http://www.membership.com."

 http://account.earthlink.com

5.1.10 using onMouseOver to Hide the Link

Some hackers use the JavaScript handler "onMouseOver" to show a different URL in the status bar of the user's email application. The below code was taken from a fraudulent email. When the user clicks over the link, the status bar will show "https://www.amazon.com/cgi-bin/webscr?cmd=_login." However the link actually takes the user to http://greenland.com/snow/scr.dll.

5.1.11 SSL Certificates

A URL that starts with https:// (instead of http://) indicates that information entered by user is being transmitted over a secure connection and the company has been issued an SSL certificate. Some fraudulent sites use an https:// URL to appear as a legitimate site. The following is a link to a fraudulent PayPal site:

https://www.paypal.com%01[string of ~60"%01"elided]@207.173.185.20/f/

Clicking on this link brought the user to "https:// 207.173.185.20/f/" and opened a security alert, which warned the viewer the certificate had been issued by a company that the user had not chosen to trust and the name on the security certificate was invalid or did not match the name on the site. However, most users are unsure what this information may indicate and these warnings are not uncommon when trying to access legitimate sites. Even with this warning, an invalid or fake certificate may make the user feel more secure in the transaction.

5.1.12 Reply Address Differs From the Claimed Sender

In some fraudulent emails messages, the email claims to be from a credible reputable company, but the email is set to reply to a fraudulent reply address. The following are some examples from fraudulent emails:

From:	Greenland Security Dept.	From:	IobBank
Reply-To:	greenland80@1-base.com	Reply-To:	Iobbank41@collegeclub.com

5.1.13 Using Pop-Ups

Many fraudulent Web pages are opened as pop-ups. Fraudsters cause the email link to go to the fraudulent Web site, which generates the fraudulent pop-up, and then redirects the main browser window to the real company site. This transaction appears to the user as a pop-up over the real

company site. Fraudsters use this technique to make their information gathering appear more credible. Some fraudsters use JavaScript to reopen the fraudulent pop-ups if closed until the user fills out the requested information. Using a pop-up with the browser menu disabled discourages the viewer from saving the page. The viewer is limited to saving the source code by right-clicking on the pop-up, selecting View source, and saving the code.

5.2 Locating the Host Server of Phishing Page:

The Pguard technique locates the host server of a phishing page using a WHOIS query. WHOIS is a query or response protocol that is widely used for querying an official database. The WHOIS database consists of autonomous system numbers, IP addresses, organizations or customers that are associated with these resources. The Pguard technique runs the WHOIS query on the URL that is contained within the phishing email. While phishing emails may give erroneous FROM emails addresses, this type of attack requires that they provide a genuine/legitimate website address for the victim to interact with. This therefore is the vulnerability in a Phisher's attack which a Pguard can exploit. A WHOIS server listens on (Transmission Control Protocol) TCP port 43 for requests of the host server and related contact information sent through web-based referrals. Once the output is finished, the WHOIS server closes its connection. The TCP connection that was closed indicates the client that the response has been received.

5.3 Removing the Phishing Page:

Upon receiving the notification of the phishing page existence on the host server through the Pguard technique, the host Administrator confirms the phishing page by testing the legitimacy of the phishing link and its genuineness. Once the Administrator confirms the phishing page, the infected or hacked website is quickly shut down to protect Internet users from further phishing. The host Administrator then notifies the website owner about the existence of the phishing page within their website. Once the phishing page is removed, if no notification has been sent to the Pguard, the Pguard periodically checks to for evidence that it has been removed. This technique assumes that website owner and host Administrator are absolutely unaware of the presence of the phishing page within their website or server until our technique notifies them.

6. CONCLUSION

We have presented a framework and an approach for multilevel monitoring of service systems. The framework specified supports the following:

- Support for the specification of abstractions over agents and their operations, and decoupling operations from commitments via a mapping specification
- Service system specifications for an arbitrary number of services and processes.
- Specification of message semantics.
- Specification of local service behaviors that contribute to the participation in multiple conversations.

This paper presented a proactive method to shut down a Phisher's operation by using a Pguard. This effectively stops a phishing attack at its source thereby protecting a significant number of other innocent users from being duped in the future. This is in contrast to the existing passive approach that only attempts to filter suspect email and allows the Phisher to continue his/her operations. While this technique does not prevent an initial phishing email from being sent, once the phishing page has been removed, all future victims are essentially protected from the Phisher. Experimental results show that this approach can be an effective way to remove phishing pages hosted on servers around the world. Furthermore, there is scope to undertake development on more aggressive techniques to address the problem of a non-responsive host Administrator that fails to shut down a phishing site.

At present our proactive approach to shutting down a Phisher is performed manually in our laboratory. Future work involves automating this technique. This would involve firstly integrating our approach with an email filtering program to initially detect a potential phishing email. The next step would be to automate the tracing and web host email notification process. The final stage would be to devise a method to tangibly check to see whether a phishing web page has been removed, and if not, what means of action then must take place. Furthermore, we plan to significantly increase the number of phishing subjects used in the experimentation to test the Pguard technique effectiveness.

REFERENCES

[1] C. E. Drake, J. J. Oliver, and E. J. Koontz, "Anatomy of Phishing Email",MailFrontier Inc.,CA,USA.

[2] M.Chandrashekaran, K.Narayana, S.Upadhyaya, "Phishing Email Detection Based on Structural Properties",Symposium on Information Assurance: Intrusion Detection and Prevention, New York, 2006.

[3] Y. Zhang, S. Egelman, L. Cranor, J. Hong, "Phinding Phish: Evaluating Anti-phishing Tools", Annual Network and Distributed System Security Symposium, USA, February 2007.

[4] K.Umapathy and S.Purao, "A Theoretical Investigation of the Emerging Standards for Web Services," 2006.

[5] N. William Robinson and Sandeep Purao, "Monitoring Service Systems from a Language-Action Prespective(LAP), March 2011.

[6] A.Lazovik et al., "Planning and Monitoring the Execution of Web Service Requests," J.Digital Libraries, 2005.

[7] J.E. Hanson et al., "Conversation-Enabled Web Services for Agents and e-Business," Proc. Int'l Conf. Internet Computing (IC), 791-796, 2002.

[8] H.Roth et al., "Probing and Monitoring of WSBPEL Processes with Web Services," Proc. Eighth IEEE ubt'1 E-Commerce Technology, 2006.

[9] N. Desai et al., "Engineering Foreign Exchange Processes via Commitment Protocols," Proc. Fourth IEEE Int'1 Conf. Service Oriented Computing (SCC), 2007.

[10] W.N. Robinson, "Monitoring Web Service Requirements," Proc. 11th IEEE Int'l Conf. Requirements Eng., pp. 65-74, 2003.

[11] N.Desai et al., "Business Process Adaptations via Protocols," Proc. IEEE Int'1 Conf. Services Computing, pp.103-110, 2006.

[12] M. Chandrasekaran, R. Chinchani and S. Upadhyaya, PHONEY: Mimicking user response to detect phishing attacks, to appear at TSPUC 2005 Workshop affiliated with IEEE WoWMoM.

[13] X. Fan et al., "A Theoretical Framework for Proactive Information Exchange in Agent Teamwork,"
Artificial Intelligence, vol. 169, pp. 23-97, 2005.
[14] L. Baresi et al., "Smart Monitors for Composed Services," Proc. Second Int'l Conf. Service Oriented Computing, pp. 193-202, 2004.
[15] S.A. Moore, "A Foundation for Flexible Automated Electronic Communication," Information Systems Research, vol.12, 2001.