PhiGARo: Automatic Phishing Detection and Incident Response Framework

Martin Husák and Jakub Čegan
Institute of Computer Science
Masaryk University
Brno, Czech Republic
Email: husakm@ics.muni.cz, cegan@ics.muni.cz

***Abstract*—We present a comprehensive framework for automatic phishing incident processing and work in progress concerning automatic phishing detection and reporting. Our work is based upon the automatic phishing incident processing tool PhiGARo which locates users responding to phishing attack attempts and prevents access to phishing sites from the protected network. Although PhiGARo processes the phishing incidents automatically, it depends on reports of phishing incidents from users. We propose a framework which introduces honeypots into the process in order to eliminate the reliance on user input. The honeypots are used to capture e-mails, automatically detect messages containing phishing and immediately transfer them to PhiGARo. There is a need to propagate e-mail addresses of a honeypot to attract phishers. We discuss approaches to the honeypot e-mail propagation and propose a further enhancement to using honeypots in response to phishing incidents. We propose providing phishers with false credentials, accounts and documents that will grant them access to other honeypot services. Tracing these honeytokens may lead us to the originators of the phishing attacks and help investigations into phishing incidents.**

***Keywords*-phishing; PhiGARo; honeypot; CSIRT; IPFIX;**

I. Introduction

No matter what technical means we use to protect IT infrastructure, a user will always be a weak point in computer security. Many forms of attacks generally named social engineering depend on this assumption. Phishing, probably the most well-known social engineering form of attack, has become one of the most popular practices among computer criminals. With publicly available tools such as Simple Phishing Toolkit [1], phishing is easy to perform and growing in numbers. One common perception is that phishing tries to gain access to the victim's banking account. In past, phishing has targeted not only clients of financial institutions, but also company employees, social networks users and even student accounts at universities. The phisher does not need to be motivated by instant financial gain, but can also gain access to mailboxes or social network accounts to harvest personal information and contacts or to send spam.

We, as a university network security department, have to handle phishing incidents. With thousands of users in the network distributed among many faculties, we do not have access to every mailserver, so we cannot guarantee the filtering of phishing messages. On the other hand, we run a network monitoring infrastructure and have access to system logs from critical servers, e.g., mail relays. The manual handling of an incident takes a significant amount of time as we have to search for victims of the incident and prevent any harm or information leak. We rely on reports by users, which is a highly unreliable source of information as not everyone is able to recognize phishing or know how to report it to a security incident response team. We need to automate the process of incident handling to reduce the time dedicated to incident handling and find a method to learn about a phishing incident that does not rely on user reports.

To formalize the scope of our work, we state three research questions which we shall answer:

(i) How can we effectively handle a phishing incident to protect common users?
(ii) Can we automate the phishing detection and incident handling?
(iii) How to attract phishers to phishing detectors?

A cornerstone of our work is PhiGARo, the automatic phishing incident processing tool. PhiGARo (Phishing: Gather, Analyze, React, and Distribute) looks up victims of phishing and prevents further harm related to the incident. A phishing incident is processed automatically, but it relies on the phishing being reported. We have to set up an automatic phishing detection method to reduce the reliance on reports from users. We decided to use a honeypot as a phishing detector. A honeypot is independent of a network setup so we do not have to modify mailservers in the network. E-mail messages incoming to a honeypot will be automatically evaluated and if a phishing is detected, the incident will be reported to PhiGARo. The automation of the detection process will allow us to handle a phishing incident right after the phishing e-mail is captured by the honeypot. We will no longer depend on reports from users or local administrators. The time window in which the phishing can cause any harm will also be significantly reduced. To provide a honeypot with incoming e-mail traffic we have to first propagate e-mail addresses of the honeypot to attract phishers. We are going to discuss approaches to the propagation of honeytokens and preliminary results.

This paper is organized into eight sections. Section II provides a survey of related work. A phishing incident response framework PhiGARo is presented in Section III. The automatic detection of phishing incidents based on honeypots is proposed

in Section IV. Attracting phishers to a phishing detector is discussed in Section V. Preliminary results are presented in Section VI. Section VII contains proposals for further work to extend the framework. Finally, section VIII concludes the paper.

II. Related work

We present a short survey of state-of-the-art phishing detection techniques and other related work. Anti-phishing techniques cover a wide area of research, we shall focus on phishing detection and the utilization of honeypots. We cover approaches and strategies for dealing with phishing incidents and their investigation. A deeper understanding of phishing is also discussed. The utilization of honeypots covers the deployment of honeypots to capture phishing as well as the propagation of honeytokens, e.g., e-mail addresses of the honeypot.

Phishing detection and recognition is a common ground for research that can be used in many cases. The common goal of this work is to find an automatic technique to detect or filter phishing e-mails. These techniques can be utilized by mailservers or e-mail clients in a similar manner to well-known spam filtering techniques. Phishing is characterized by certain common signs, a phishing message typically contains a link to a phishing website or keywords such as *password*, *login*, etc. Phishing is also often written in poor language due to mechanic translation of the message into another language. Common techniques usually involve the detection of these characteristics.

A literature survey of phishing detection was presented by Khonji et al. [2]. Almomami et al. [3] present a survey of phishing email filtering techniques. A comparison of machine learning techniques for phishing detection was presented by Abu-Nimeh et al. [4]. Pandey and Ravi [5] present phishing detection based on text and data mining. Chandrasekaran et al. [6] mimicked user responses to detect phishing. Their work places the response before detection to provide the adversary with fake responses.

Understanding phishing attacks in depth is another welcomed topic of research. A framework for the detection and measurement of the phishing attacks is proposed by Garera et al. [7]. A forensic framework for tracing phishers is proposed by Gajek [8]. McRae and Vaughn [9] present the use of so-called web bugs and honeytokens to trace the sources of phishing attacks.

In our work we propose using honeypots, a well-known tools among the IT security community. The general idea behind honeypots is that they have no production value, therefore any access to them is by nature suspicious. In the case of e-mail traffic, any message targeting a honeypot mailserver is suspicious, preferably spam or phishing. Recent advances and future trends in honeypot research are outlined in a survey by Bringer et al. [10]. The survey concisely deals with honeypots used against spammers and phishers.

Honeypot mailservers are popular tools among the honeypot community. They are mainly focused on capturing spam, which is where the alternative term spamtrap comes from. Spamtraps have helped to analyze large volumes of spam, with the addition that many phishing messages were captured as well. The implementations and field results of using e-mail honeypot (spamtrap) are presented by Rathgeb and Hoffstadt [11].

False accounts, credentials or documents, generally referred as honeytokens, are another interesting form of honeypots in phishing detection. As Spitzer states [12], the honeytokens are as old as security itself, although the term first appeared in 2003. Using honeypot accounts to lure phishers and to test phishing detection tool is presented by Yu et al. [13].

Li and Schmitz [14] present an anti-phishing framework based on honeypots. They propose transforming real e-banking system into a honeypot equipped with honeytokens. The novelty of this approach is in automatically detecting the theft of money from accounts and asking for confirmation from the victim.

We learned from the survey of related work that there is no general framework for a phishing response. The anti-phishing solutions found aim to protect specific service or resource while we aim to provide incident response and protect users. On the other hand, we can take advantage of work dedicated to phishing detection and utilizing honeypots. Many techniques of phishing detection were proposed including data mining or machine learning approaches. Honeypots were successfully used for understanding and investigating phishing incidents. Specifically, we may utilize the experiences with honeytoken propagation.

III. Phishing Incident Response

We propose a general framework for phishing detection and automatic incident handling. Our solution is split into two parts, phishing incident processing and phishing detection. The core of our framework is the phishing incident processing part presented in this section. Automatic phishing detection is proposed in the next section.

We look at the problem from the point of view of a Computer Security Incident Response Team (CSIRT) of an organization. We do not create a solution to protect a specific service or resource against phishers but we are trying to protect users in our network from responding to phishing e-mails and accessing phishing websites. We are also working in an environment with limited access to production resources as the network is distributed among many faculties and administrators. On the other hand we use network monitoring based on network flows to gather information about network traffic.

The phishing incident response framework has its roots in the manual handling of phishing incidents. We have identified a common phishing incident handling process during the manual handling. Although we have standardized the process to some extent, it still took several hours to handle an incident manually. The phishing incident processing tool, PhiGARo [15], was developed to automate the process. This tool has become the core of our proposed framework.

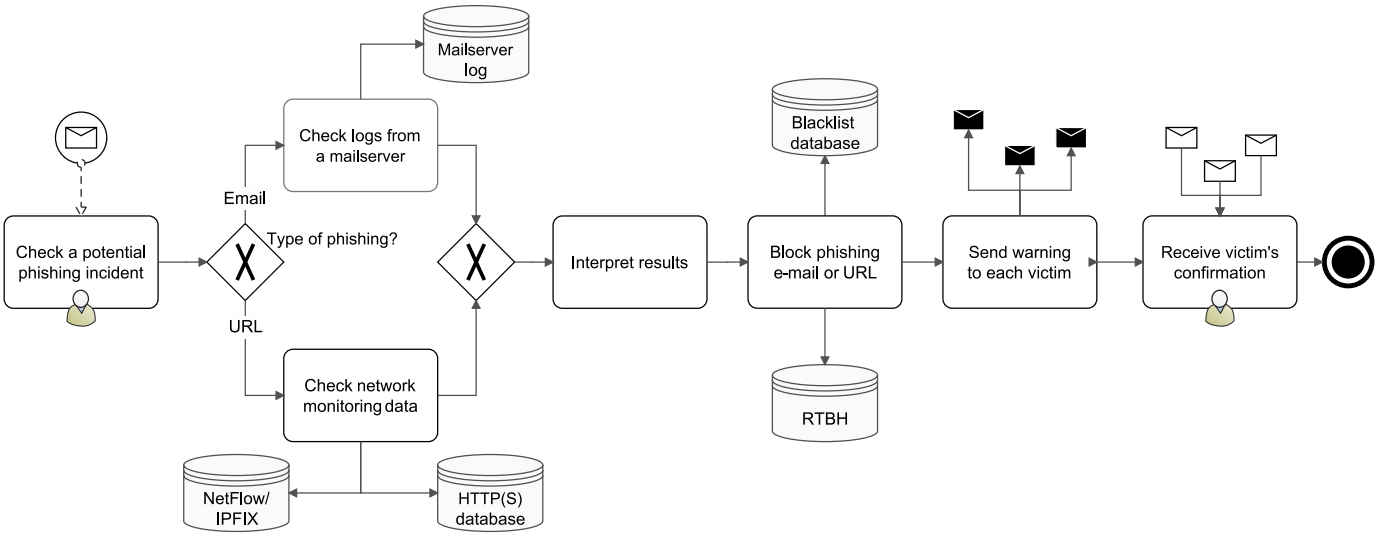

Fig. 1. Handling of a phishing incident

Phishing incidents are reported by users who identify a phishing message in their mailboxes. The report is sent via e-mail or web form and accepted by the Request Tracker [16]. A command line API is available for users who do not use the Request Tracker. PhiGARo is executed by the incident handler and executes all the steps in and incident process that used to be done manually. System logs and network monitoring data are checked, the results are interpreted, the phishing e-mail or URL is blocked, and the victims are notified. PhiGARo may be executed automatically, but the incident is first checked by an incident handler for safety reasons.

A workflow of PhiGARo is presented in Figure 1. PhiGARo starts with a phishing type evaluation to determine which method to use to find victims. If the phishing contains an URL, network monitoring is used, system logs from e-mail server are used otherwise. The reported URL is examined to uncloak the real URL hidden behind URL shorteners and redirections. A screenshot of the phishing website is obtained for later use.

NetFlow [17], a well-known network monitoring technology, is used to locate victims who accessed the phishing URL. NetFlow processes the network traffic on a third layer of the OSI model. The IP address of a phishing website is resolved and PhiGARo searches for network traffic between the IP address and the monitored network. The IP addresses, from which the phishing website was accessed, are marked as potential victims of phishing.

Due to the fact that many phishing sites are located on legitimate webservers or share an IP address with them, the extended network flow monitoring based on IPFIX [17] is used. IPFIX is extensible with the parsing of HTTP(S) requests [18]. This network monitoring technique provides a database of requested URLs in addition to L3 network traffic statistics. If the database is available, PhiGARo will search the HTTP(S) requests for the phishing URL. This approach provides significantly less false positive detections in comparison to using NetFlow or bare IPFIX. We are able to capture all URL requests in the network traffic of our university, i.e., around 15,000 machines and 40,000 users per day. We have to consider the privacy of our users due to the nature of the observed data.

If there is no URL in the phishing message, the system logs of e-mail servers are parsed. PhiGARo searches for e-mails sent in response to the phishing message. The e-mail responses to a phishing message are recognized by timestamps and the e-mail addresses of the sender and the recipient. The victim is instantly identified by their e-mail address.

When the victims are identified, i.e., by e-mail or IP address, PhiGARo continues with an interpretation of results and false positive detections are filtered. False positive detection occurs when the victim accesses the phishing website, but does not send anything. False positives are identified by a small number of packets sent from a victim to a phishing website. HTTP(S) module is able to distinguish between a request that just visits a site and a request which sends data. The detection of e-mail responses in the mailserver logs is generally free of false positives.

The preventive counter-measures are provided by blocking and reporting modules. PhiGARo is connected to a blocking mechanism used in the protected network. RTBH (Remotely-Triggered Black Hole) [19] is used to block any traffic between the protected network and an IP addresses where the phishing website is hosted. E-mail addresses are forwarded to mailservers which then filter messages containing these addresses. Reporting modules are responsible for forwarding the reports to third parties. For example, a report is sent to Google if the phishing website is hosted on Google Docs sites.

The final step of phishing incident processing by PhiGARo is notifying the victims. PhiGARo sends e-mail warning to all identified victims. The warning contains an explanation of the incident, sample of a phishing message, evidence of victim's activity, and a screenshot o the phishing website, if available. The incident is then marked as resolved in the

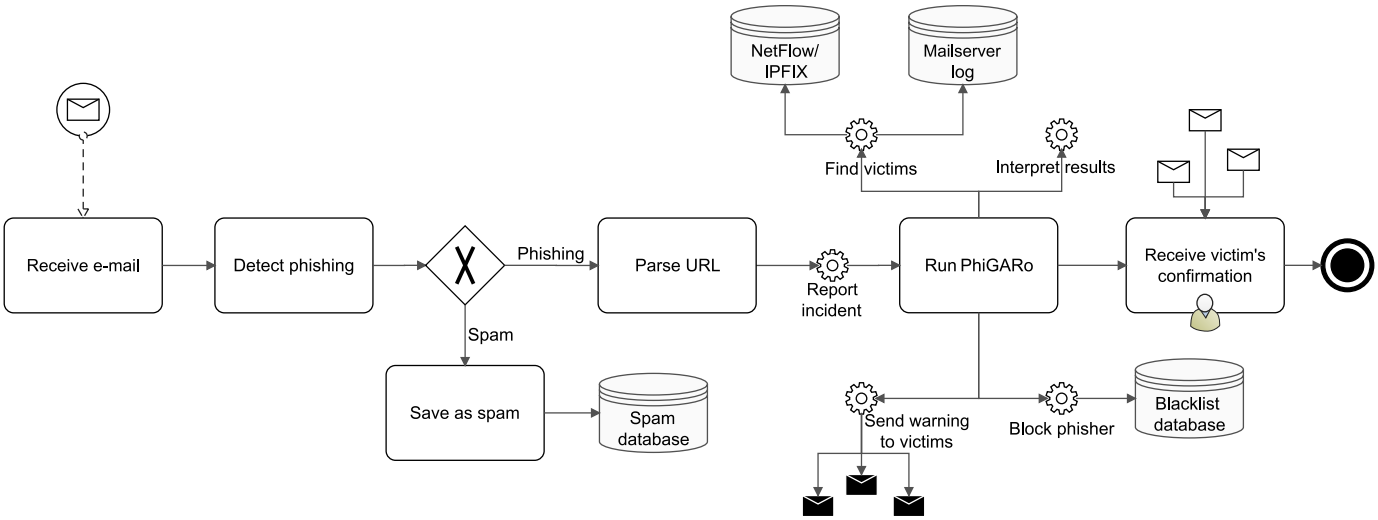

Fig. 2. Phishing detection and incident response framework

Request Tracker. Confirmations and questions from victims are received by Request Tracker and resolved by an incident handler.

PhiGARo was implemented and deployed in our network, the results are presented in Section VI. PhiGARo is easily extensible due to its modular architecture. A software package with PhiGARo is publicly available [15] and additional modules were added recently [20].

IV. Phishing detection using honeypots

In this section, we propose the second part of our phishing detection and incident response framework. We need to overcome the reliance on user reports and provide PhiGARo with an automatic phishing incident reporting tool. We propose using honeypots to capture, detect, and report phishing. Honeypots do not need to be tied to the production network so the phishing can be detected without access to mailservers in the network. Separate from PhiGARo, the development of an automatic phishing detection and reporting tool is in progress.

A high-interaction honeypot, i.e., physical or virtual machine with real operating system and applications, is configured as a mailserver and acts as an open relay. The honeypot accepts any incoming e-mail and does not send or forward anything. The whole server is configured as a spamtrap. Any incoming e-mail message is accepted and saved into a common mailbox, including messages for non-existent accounts. We were partially motivated by the assumption that the spammers (not necessarily phishers) may try to send messages not only to e-mail addresses known to them, but also to derived addresses. By derived addresses we mean common accounts existing in domains such as *abuse*, *admin*, etc. An adversary may find a new address from a previously unknown domain and insert it into the recipient list along with the derived addresses from the same domain. Since we are interested in the content of the message, not the recipient, we do not need to create an account for any potential recipient.

When the honeypot accepts an e-mail, we need to determine if it is phishing or common spam. Many techniques of phishing detection were proposed as presented in related work [2], [3]. A phishing message has to be detected automatically. The task is the same as in common mailservers or e-mail clients where phishing detection techniques may be used for filtering. Therefore we can apply any existing method of phishing detection [4], [5]. Only a slight modification is applied to pass the message to the reporting module. The reporting module will generate a phishing report for further processing by PhiGARo.

The techniques of phishing detection are evaluated and the most suitable will be selected for deployment in a framework. Simple methods based on URL detection or searching for keywords are evaluated first to provide the phishing detection framework with basic functioning modules. The simple approaches are easy to implement as we can use common e-mail antivirus software such as ClamAV [21] and specify custom filtering rules suited for phishing detection [22]. Data-mining [5] and machine learning [4] approaches are also an option. These techniques will be evaluated from the point of view automating the process and easy deploying it into the framework. A database of phishing incidents processed by PhiGARo will be made accessible for the methods that use machine learning or a history of detected phishing incidents.

A simple search of URLs in the messages was sufficient in a preliminary phase of development to detect phishing. An URL is often included in a phishing message, although a significant number of spam also contains URLs. Therefore, a simple URL lookup is not sufficient for automatic spam detection as it would cause false positive reports. On the other hand, an URL is one of the parameters of PhiGARo. We deploy an URL parsing module to to detect an URL in the phishing messages already recognized as phishing by any other method. A parsed URL is then passed to PhiGARo along with the original message and other parameters.

A workflow of automatic phishing detection and reporting is presented on Figure 2. The e-mail message is accepted and passed to a phishing detection module in the honeypot. If the message is identified as a phishing attempt, a URL detection module searches for any URL in the message and parses it. A standardized phishing report is generated and sent to the Request Tracker where PhiGARo takes charge of processing the incident. Reports from the honeypot will start PhiGARo automatically without approval by the operator. We suggest that these reports are considered valid in contrary to reports by users.

V. Attracting phishers

Although we have proposed the automatic framework incident response and enhanced it with an automatic phishing detection based on honeypots, we still have a problem to solve. The deployment of a honeypot itself is not sufficient, we have to attract the phishers to a phishing detector. The phisher needs to know the e-mail addresses of the honeypot to send phishing there. The e-mail addresses, generally named honeytokens, need to be propagated. In this section, we discuss the techniques to attract phisher to the honeypot.

The propagation of honeypot e-mail addresses or honeytokens in general can be active or passive. Passive propagation is making honeytokens publicly available, e.g., making them accessible from an organization's website where they may be accessed by web crawlers or e-mail harvesters [23]. Active propagation involves pushing the honeytokens to the phisher, either by publishing them on websites dedicated to phishing or offering them on a black market [24]. Approaches to honeytoken propagation are discussed and evaluated during the development of the framework. Suitable techniques will become a part of the framework.

The most common passive method of honeytoken propagation is publishing them on a regular website. The honeytokens, i.e., e-mail addresses, are instantly accessible by web crawlers. Honeytoken should be placed on a website that is frequently visited, e.g., main site of an organization, if applicable. Honeytokens should also be hidden to avoid access to them by legitimate users. HTML provides many ways to do that, e.g., by placing a honeytoken to an invisible frame or covering it with some other element. The automatic crawlers and harvesters cannot tell the difference between legitimate contacts on the sites and honeytokens. It may be advisable to precede the honeypot e-mail address with a *mailto:* keyword which may be searched for by some e-mail harvesters.

Active methods of honeytoken propagation have a higher potential in attracting attackers compared to passive methods. Although the passive propagation is basically publishing the honeytokens, it has to be done manually. The same applies for many active methods, which overlap with marketing and social engineering [24]. Active methods may seem more difficult to proceed, but we propose there are active methods suitable for automation. The phishing campaigns detected earlier are used and the phishers are provided with fake responses to their phishing attempts. Inspired by the related work [6], [8], we propose an automatic response to phishing messages from a honeypot e-mail addresses so that they appear active and vulnerable to phishing. The phisher will add the responding addresses to the recipient list and include them in the next phishing campaign. The automatic e-mail response to phishing will be added as a module to PhiGARo.

In summary, two approaches to automatic honeytoken propagation were proposed, one of them is ready to use. Both methods do not solve initial honeytoken propagation. We either propagate the honeytokens in response to a manually reported phishing incident, or do the initial honeytoken propagation manually. Once the automatic propagation is included into the processing of manually reported phishing attempts, it will provide phishers with the e-mail addresses that will be used in the next phishing campaign. The next time, the incident will be processed automatically from phishing detection to incident response.

VI. Achieved results

PhiGARo, the automatic phishing incident response tool, was implemented and deployed in our network with promising results. The software package [15] was released for public use and additional modules were published later [20]. Although PhiGARo uses primarily the tools available in our network, it is modular and extensible. PhiGARo has become the core of our proposed network, however, the other parts of the framework are currently under development. In this section, we present the results achieved to the current time and lessons learned from deploying PhiGARo and implementing the rest of the framework.

We processed 79 phishing incidents in 2012 and 133 phishing incidents in 2013. The average time spent on phishing incident handling is compared in Figure 3. The machine time remains the same while the time spent by incident handlers is significantly reduced. We saved hundreds of hours of manual work by deploying PhiGARo in 2013. In comparison, the development of PhiGARo took about 0.2 FTE (Full-time equivalent) for one year, so payback occurred in the first year of using it. In addition, the incident handler does not need complete know-how to handle a phishing incident and junior handlers can process it independently.

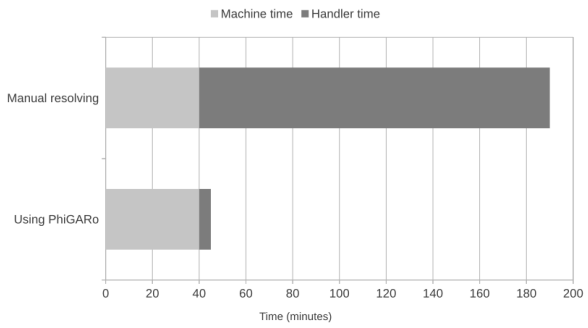

Fig. 3. Average time spent on phishing incident handling in 2013

We suppose that the number of phishing attacks against our network is significantly higher. Only the incidents reported by users were processed. We have tried to make reporting easier for users by providing a web form for phishing incident reports. The users are either not willing to report the incidents or they do not know about the option. On the other hand, there is a small group of active users who reported every phishing message they encountered. The majority of the reports are reported by these few users.

Despite the educational effort, we believe that most of the users are not aware of the threat until they are explicitly warned they fell for phishing. Units of victims are identified per month, although their number may also be higher. Not everyone is willing to admit their mistake and many users try to persuade us that their case is a false positive.

Therefore, we have confirmed our previous expectations that depending on user reports is unreliable. Although a small number of users are willing to participate in the network protection, there are thousands of users who may have become victims of phishing. We suggest that the real number of phishing attempts is higher, therefore we need an automatic phishing detection tool to reduce the reliance on users. Our effort is now directed towards deploying the honeypot and completing the implementation of the proposed framework.

We have deployed a honeypot in our network as proposed in Section IV. Our honeypot is configured as a spamtrap and accepts any incoming e-mail to any e-mail address. We have assigned a custom third-level domain and the e-mail traffic to the domain is routed to the honeypot. A short survey of automatic phishing detection techniques was presented in Section II and discussed in Section IV. The techniques of phishing detection are currently being evaluated and the most suitable one will be deployed to detect phishing which is incoming to the honeypot. Reports will be sent to the Request Tracker where PhiGARo takes charge of the incident processing.

Several unique e-mail addresses belonging to the domain were propagated as honeytokens to attract adversaries. One group of honeytokens were propagated passively by placing them on various websites of our university. The other group of honeytokens was used for active propagation. We have responded to several phishing messages from these addresses and filled several phishing web forms with false information and honeytokens. The honeytokens are active approximately three months and their efficiency is currently being evaluated.

When we published the e-mail addresses of the honeypot we expected the prevalence of common spam and a smaller amount of phishing. We were not expecting to receive spam or phishing immediately, it took almost a week before the first spam was received and almost a month before the first phishing attempt was observed. Due to the small number of honeytokens and the short time of their propagation we received hundreds of spam e-mails and only a few phishing attempts. The most common type of received message was scam. A significant number of spam was related to academia and contained invitations to questionable scientific conferences. A bigger spam campaign that lasted for several weeks and had several language mutations was also observed.

Apart from phishing attempts, we paid attention to messages that may be mistaken for phishing by automatic tools. For example, a simple technique of phishing recognition is based on URL detection. We have observed many spam messages containing URLs that were not phishing attempts. First type of such messages contained subjects related to recent news to catch the reader's attention. The links then redirected user to a malicious websites. The second type of messages containing URLs were spam advertising a product and linking to its website. Apart from this, we observed a spam campaign where each e-mail contained link to a (most likely fake) LinkedIn profile of the sender to support the trustworthiness of the message. A whole network of mutually connected accounts was linked in the spam content.

In summary, we evaluated the efficiency of various approaches to honeytoken propagation. We confirmed that it may take weeks before the first e-mail is received by the propagated address [11]. The passive technique, i.e., publishing honeytoken on a website, was successful at attracting spammers, however, phishers were not attracted as much. Active techniques are also evaluated, we responded to phishing messages from the honeypot e-mail addresses and filled in web forms on phishing websites with honeytokens. In the next phase of honeytoken propagation we will scatter e-mail addresses of freemail accounts to compare their efficiency in attracting phishers with the e-mail addresses in our honeypot domain.

VII. Further work

We propose further work to extend the scope of phishing detection and incident response. We have identified three areas of further research and development which may increase the usability of the proposed framework. The investigation of a phishing incident takes place after the incident response to trace the adversary. In order to propagate honeytokens and attract phishers it may be necessary to support the trustworthiness of honeytokens and honeypots. Outsourcing the honeypots as phishing detection tools may be used to lowering the costs of phishing detection, but can also increase the chances of successful phishing detection.

An investigation into a phishing incident aims at locating the source of the attack and tracing the adversary. Therefore, the basic beginning of any phishing investigation is responding to it. It is quite common to respond to unsolicited e-mail on purpose. We were inspired by the scambaiters who are responding to scam messages, although mostly just for fun or to keep scammers busy with their demands [25]. We followed up on the work of Garera [7] in the field of phishing measurement and the work of Gajek [8], by proposing a forensic framework for tracing phishers. We are interested in the possibilities of automatic phishing investigation to enhance the capabilities of our framework.

We have already proposed an automatic response to phishing to propagate the e-mail addresses of the honeypot and a

phishing web form filling module was implemented in PhiGARo. We will use these capabilities to propagate honeypot e-mail addresses as well as honeypot accounts and credentials. Using honeytokens to trace the source of the phishing attacks is discussed by McRae and Vaughn [9]. We suggest creating an account with a unique password on the honeypot and then sending the credentials to a phisher. An analysis of the passwords used by attackers [26] will help us avoid common passwords. The honeypot will detect authentication attempts and detect the usage of the unique honeytoken password. The unique password will link the attacker with a phisher and we can observe the spread of the honeytoken. The honeypot may then provide the attacker with another honeytoken, e.g., a document generated by the tool HoneyDocs [27]. The document reports its status and allows us to trace the attacker.

We propose further support from honeypots to increase the trustworthiness of the honeytokens and support the phishing investigation. Spreading the honeypot accounts and documents requires the deployment of the honeypots to host them. A simple honeypot may be insufficient, therefore it is advisable to support it with an appropriate domain name, content, and provided services. We propose the concept of a virtual organization and deploying the honeypot within an organization. The honeypot provides content and services resembling an actual organization or its department. The provided content and services, e.g., fake website and mailserver, will prevent the attacker from recognizing a honeypot. Although providing content is laborious, we suggest it is worth the effort to support incident investigation.

The third area of further research concerns problems of outsourcing anti-phishing honeypots and is addressed by Li and Schmitz [14]. We suggest that outsourcing honeypots used in our proposed framework is possible and can be beneficial. The honeypot is logically separated from the production network and only relevant e-mail traffic is forwarded there. We propose providing honeypots as a service. The interested users may outsource the honeypot to a provider and forward e-mail traffic there. The provider will receive messages for honeypot e-mail addresses of all customers. Each customer will then be provided with detected phishing reports. The phishing detection capability will increase with every new customer while they can focus on responding to the phishing incident.

VIII. Conclusion

In conclusion, we have proposed a general framework for phishing detection and incident response. State-of-the-art phishing detection techniques and methods of using honeypots were presented to provide a general overview and context for our work. Three areas of interest were identified and formalized in research questions. The areas were processing a phishing incident, automatic phishing detection based on honeypots, and the propagation of honeytokens to attract phishers to a honeypot.

In the first question, we asked how we can effectively handle a phishing incident to protect common users. The manual handling of a phishing incidents was laborious, so the process was standardized and served as a guideline for automatic phishing response framework. PhiGARo, the automatic phishing response tool, was developed and became the core of the proposed framework. We were able to automatically locate the victims of phishing using network monitoring and system logs from mailservers. Further harms were prevented by blocking the phishing messages and access to phishing websites. Victims were noticed via e-mail and the phishing incident documented and saved for possible investigation. PhiGARo saved hundreds of hours spent on manual incident handling.

The second question regards the automatic detection and reporting of phishing incidents. PhiGARo is responsible for automatic phishing incident response, but the incident has to be reported by users. We proposed a solution based on a honeypot to overcome the reliance on user reports. The honeypot accepts any incoming e-mails and automatically detects phishing. Simple methods, as well as state-of-the-art techniques, were discussed and evaluated. The most suitable automatic method will be then deployed in the honeypot for automatically detecting phishing and reporting it. The honeypot will immediately report any phishing incident to PhiGARo which takes charge of the incident processing. Our proposed solution reduces the reliance on user reports and the time window between phishing detection and incident response.

The third question addresses attracting phishers to a phishing detector. We have to propagate the honeytokens, i.e., e-mail addresses of the honeypot which will receive the phishing messages. The more successful we are at propagating honeypot e-mails, the more phishing campaigns will target honeypots and the more phishing incidents will be detected. We presented active and passive approaches to honeytoken propagation and proposed an automatic method of honeytoken propagation which is partially implemented in the framework. PhiGARo fills in web forms on phishing websites with the honeytokens. We also suggest an automatic response to the phishing messages from the honeypot addresses. Both methods are evaluated and will become part of the framework.

We have proposed three areas of further work which can enhance the capabilities of our framework. An investigation into a phishing incident is suggested as the next step after the phishing detection and response. The concept of a virtual organization may increase the trustworthiness of the honeypot and the propagated honeytokens. Finally, we suggest outsourcing the honeypot and providing honeypot as a service offered to automatically detect phishing and report it.

The next step, however, is finishing the implementation of the phishing detection and response framework. PhiGARo [15] was released as a software package and provided with additional modules [20]. Overall, we hope that our work can improve the security of users and prevent further harm. The results and lessons learned from the deployment of the framework in our campus network will be published for the benefits of security community.

REFERENCES

[1] "Simple Phishing Toolkit," 2013. [Online]. Available: http://sptoolkit.com/

[2] M. Khonji, Y. Iraqi, and A. Jones, "Phishing Detection: A Literature Survey," *Communications Surveys Tutorials, IEEE*, vol. 15, no. 4, pp. 2091–2121, Apr 2013.

[3] A. Almomani, B. Gupta, S. Atawneh, A. Meulenberg, and E. Almomani, "A Survey of Phishing Email Filtering Techniques," *Communications Surveys Tutorials, IEEE*, vol. 15, no. 4, pp. 2070–2090, Apr 2013.

[4] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A Comparison of Machine Learning Techniques for Phishing Detection," in *Proceedings of the Anti-phishing Working Groups 2Nd Annual eCrime Researchers Summit*, ser. eCrime '07. New York, NY, USA: ACM, 2007, pp. 60–69.

[5] M. Pandey and V. Ravi, "Detecting phishing e-mails using text and data mining," in *Computational Intelligence Computing Research (ICCIC), 2012 IEEE International Conference on*, Dec 2012, pp. 1–6.

[6] M. Chandrasekaran, R. Chinchani, and S. Upadhyaya, "PHONEY: Mimicking User Response to Detect Phishing Attacks," in *Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks*, ser. WOWMOM '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 668–672.

[7] S. Garera, N. Provos, M. Chew, and A. D. Rubin, "A Framework for Detection and Measurement of Phishing Attacks," in *Proceedings of the 2007 ACM Workshop on Recurring Malcode*, ser. WORM '07. New York, NY, USA: ACM, 2007, pp. 1–8.

[8] S. Gajek and A.-R. Sadeghi, "A forensic framework for tracing phishers," in *The Future of Identity in the Information Society*. Springer, 2008, pp. 23–35.

[9] C. McRae and R. Vaughn, "Phighting the Phisher: Using Web Bugs and Honeytokens to Investigate the Source of Phishing Attacks," in *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*, Jan 2007, pp. 270c–270c.

[10] M. L. Bringer, C. A. Chelmecki, and H. Fujinoki, "A Survey: Recent Advances and Future Trends in Honeypot Research." *International Journal of Computer Network & Information Security*, vol. 4, no. 10, 2012.

[11] E. Rathgeb and D. Hoffstadt, "The E-Mail Honeypot System Concept, Implementation and Field Test Results," in *Digital Society, 2008 Second International Conference on the*, Feb 2008, pp. 1–6.

[12] L. Spitzner, "Honeytokens: The other honeypot," 2003. [Online]. Available: http://www.securityfocus.com/infocus/1713

[13] W. Yu, S. Nargundkar, and N. Tiruthani, "PhishCatch - A Phishing Detection Tool," in *Computer Software and Applications Conference, 2009. COMPSAC '09. 33rd Annual IEEE International*, vol. 2, July 2009, pp. 451–456.

[14] S. Li and R. Schmitz, "A novel anti-phishing framework based on honeypots," in *eCrime Researchers Summit, 2009. eCRIME '09.*, Sept 2009, pp. 1–13.

[15] J. Čegan, J. Soukal, M. Drašar, and J. Vykopal, "PhiGARo – tool for phishing incident processing," Masaryk University, 2012. [Online]. Available: http://www.muni.cz/ics/services/csirt/tools/phigaro

[16] "RT: Request Tracker," 2014. [Online]. Available: http://www.bestpractical.com/rt/

[17] R. Hofstede, P. Čeleda, B. Trammell, I. Drago, R. Sadre, A. Sperotto, and A. Pras, "Flow Monitoring Explained: From Packet Capture to Data Analysis with NetFlow and IPFIX," *Communications Surveys Tutorials, IEEE*, vol. PP, no. 99, pp. 1–1, 2014.

[18] P. Velan, T. Jirsík, and P. Čeleda, "Design and Evaluation of HTTP Protocol Parsers for IPFIX Measurement," in *Advances in Communication Networking*. Springer, 2013, pp. 136–147.

[19] Cisco Systems, "Remotely Triggered Black Hole Filtering," http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6586/ps6642/prod_white_paper0900aecd80313fac.pdf, 2005, whitepaper.

[20] J. Čegan, T. Šíma, J. Soukal, and M. Drašar, "PhiGARo HTTP(S) – tool for phishing incident processing," Masaryk University, 2013. [Online]. Available: http://www.muni.cz/ics/services/csirt/tools/phigaro

[21] "Clam AntiVirus," 2014. [Online]. Available: http://www.clamav.net/lang/en/

[22] T. Edwin, "Phishing signatures creation HOWTO," 2014. [Online]. Available: http://www.clamav.net/doc/latest/phishsigs_howto.pdf

[23] O. Hohlfeld, T. Graf, and F. Ciucu, "Longtime Behavior of Harvesting Spam Bots," in *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, ser. IMC '12. New York, NY, USA: ACM, 2012, pp. 453–460.

[24] A. Shulman, "The underground credentials market," *Computer Fraud & Security*, vol. 2010, no. 3, pp. 5–8, 2010.

[25] A. Zingerle and L. Kronman, "Humiliating Entertainment or Social Activism? Analyzing Scambaiting Strategies Against Online Advance Fee Fraud," in *Cyberworlds (CW), 2013 International Conference on*. IEEE, 2013, pp. 352–355.

[26] V. Nicomette, M. Kaâniche, E. Alata, and M. Herrb, "Set-up and deployment of a high-interaction honeypot: experiment and lessons learned," *Journal in Computer Virology*, vol. 7, no. 2, pp. 143–157, 2011.

[27] "HoneyDocs," 2013. [Online]. Available: http://honeydocs.com/