Honeypots : Fighting Against Spam

Alok Shukla
AIIT
Amity University

Kanchan Hans
Asst. Prof. AIIT
Amity University

Abstract:

Spam can be considered as most annoying cyber-pollution in the current scenario. Honeypots is an approach to fight against spam by information gathering and learning. First Section discusses the functionality and architecture of honeypots. In the next section, we have discussed the usefulness of honeypots to fight against both e-mail spam and spam in search engine. At the end, we have addressed the approaches to detect and prevent spam.

Introduction

The Internet is growing at a very rapid speed and multiplying its websites in every 30 days and the number of people using the internet is growing day by day. Hence, Global communication is playing an important role in day to day life. At the same time, computer crimes are also increasing. Counter measures are developed to detect or prevent attacks - most of these measures are based on known facts, known attack patterns. As in the army, it is important to know, who your enemy is, what kind of strategy he uses, what tools he utilizes and what he is aiming for. Congregation of this type of information is not easy but important for making new strategy. By knowing attack strategies, countermeasures can be improved and vulnerabilities can be fixed [5]. To gather as much information as possible is one main goal of a honeypot. Generally, such information gathering should be done silently, without alarming an attacker. All the gathered information leads to an advantage on the defending side and can therefore be used on productive systems to prevent attacks.

According to Lance Spitzner a honeypot as "a security resource who's value lies in being probed, attacked or compromised"[1,2]. In other words A honeypot is a computer system on the Internet that is expressly set up to attract and "trap" people who attempt to penetrate other people's computer systems [3].In other words A honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource [4]. A honeypot is primarily an apparatus for information gathering and learning. Its primary purpose is not to be an ensnare for the blackhat community to seize them in action and to press charges against them.[5] The focus lies on a silent collection of as much information as possible about their attack patterns, used programs, purpose of attack and the blackhat community itself. All this information is used to learn more about the blackhat measures and motives, as well as their technical knowledge and abilities. This is just a primary purpose of a honeypot. There are a lot of other potential for a honeypot - reroute hackers from productive systems or catch a hacker while conducting an attack are just two possible examples. They are not the perfect solution for solving or preventing computer crimes. Honeypots are hard to maintain and they need operators with good acquaintance about operating systems and network security. In the right hands, a honeypot can be an effectual tool for information gathering. In the wrong, unexperienced hands, a honeypot can become another infiltrated machine and an instrument for the blackhat community.

Role of Honeypots

Honeypots are designed to be broken into for two primary reasons. One of these is to find information about vulnerable areas of a system and those that are most likely to be attacked [6]. Essentially, by doing this one can learn how a system can be compromised by observing attack methodologies. The second main goal of Honeypots is to gather forensic information required to aid in the apprehension or prosecution of intruders. Honeypots purposely leaves a "hole" in the system that is so obvious to walk through that other areas of the system look relatively much more secure. In essence, the Honeypot then protects the other areas of the system or network by diverting attention to it.

Honeypots are always subject to scrutiny by its use because of the controversy of it being labeled as a form of entrapment. Honeypots are in fact not a form of entrapment because it lets the system afford an attack and does not encourage being attacked. Legally you can be liable if a Honeypot is compromised and used as a launching pad for other unauthorized intrusions. If the Honeypot is however virtual enough and really only simulates, then launching attacks from a Honeypot would be harmless [7].

Working Technique of Honeypots

Fighting against the spam sources cannot be done through one tool, because sending unsolicited e-mails is just the last step of a complex set of operations that are carried out by spammers[12].Hence, an adequate fight against spam sources requires a framework of different tools and, possibly ,the cooperation among interested organizations. The proposal of this paper does not yet include cooperation, but it describes a framework (*HoneySpam*) that is built up of many components among which the pivot is represented by the Honeypot technology. The traditional goal of Honeypots is to trace the activity of an attacker, locate him through a *trace back* operation, and recognize common operational patterns in attacks with the purpose of automatically detecting them in the future. HoneySpam faces the following spamming actions at the source:

- *e-mail harvesting*;
- *anonymous operations*.

E-mail harvesting that is, the collection of legitimate e-mail addresses through automated crawlers , is fought through emulated Web services[8,9]. Two different damages are inflicted to crawlers: *crawler slowdown* and *e-mail database poisoning*. First, the crawlers are slowed down into endless loops of Web page retrievals, through the creation of Web pages with many hyperlinks. Second, databases used by spammers to collect the e-mail addresses are polluted. In particular, the pages generated by the Web service contain special, valid, dynamically created e-mail addresses that are handled by emulated SMTP services present in HoneySpam. As soon as the spammer starts to use these freshly collected addresses, it can be traced back and blacklisted. Fraudulent activities such as spamming are often achieved anonymously, through the use of open proxy chains. HoneySpam provides fake open proxy and open relay services that log every activity and block spam-related traffic [10,11].

Spammer activities

E-mail harvesting

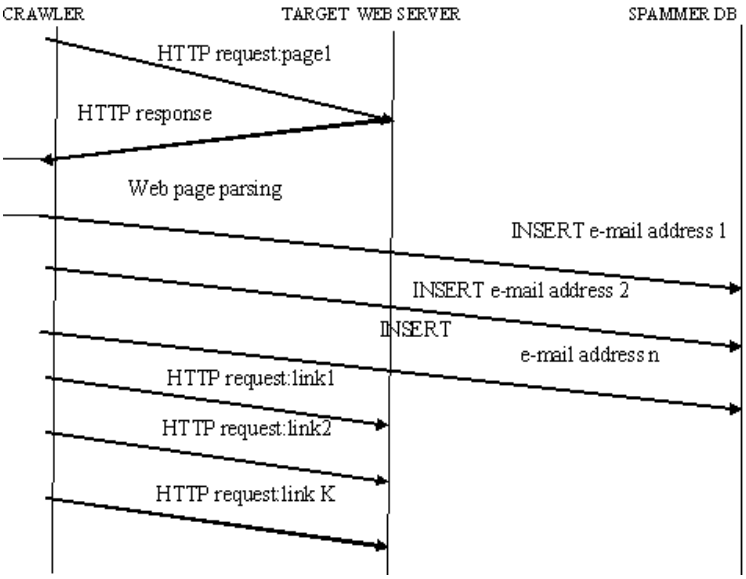
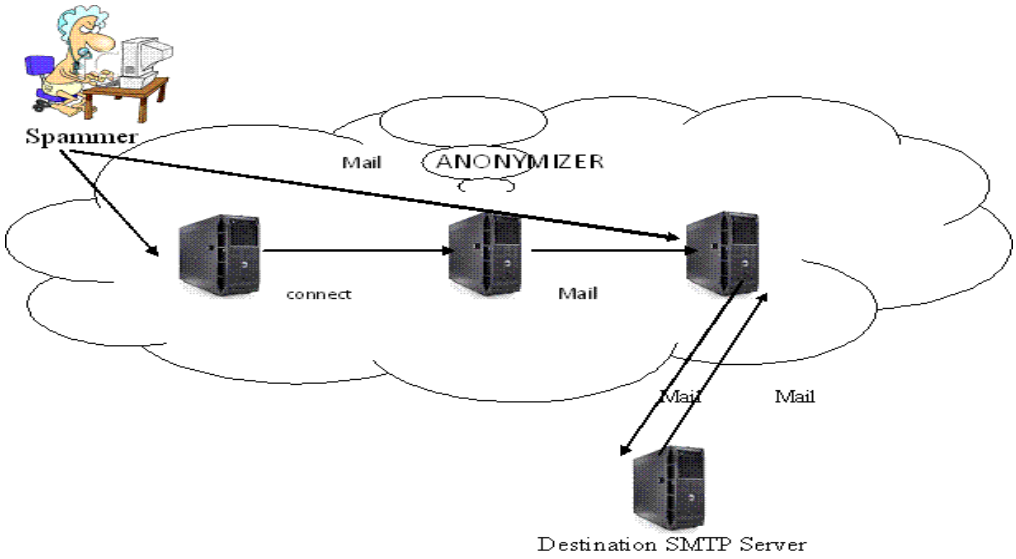


Figure shows one of the first actions performed by spammers that is, *e-mail harvesting*. An automated browsing software (called *crawler*) issues a HTTP request to fetch a Web document (*page 1*). Once the HTTP response is retrieved from the Web server, the crawler parses its content, extracting links to e-mail addresses and to other Web pages. Links to Web pages are used to continue the crawling process (*link 1* to *link k*), while links to e-mail addresses (*address 1* to *address n*) are used to compose lists or (more usually) to populate databases of potential *customers*.

E-mail harvesting is an important step, because a spammer has to build its list of victims before sending unsolicited messages to them. It should be pointed out that (thanks to existing collections of e-mail addresses, usually available on CDs or databases), e-mail harvesting is not performed by every spammer, since long lists of (presumably valid) e-mail addresses can be bought from the Internet. Although harvesting actions have softened recently, they still remain one of the major sources of e-mail addresses for spammers[13].

Operating anonymously

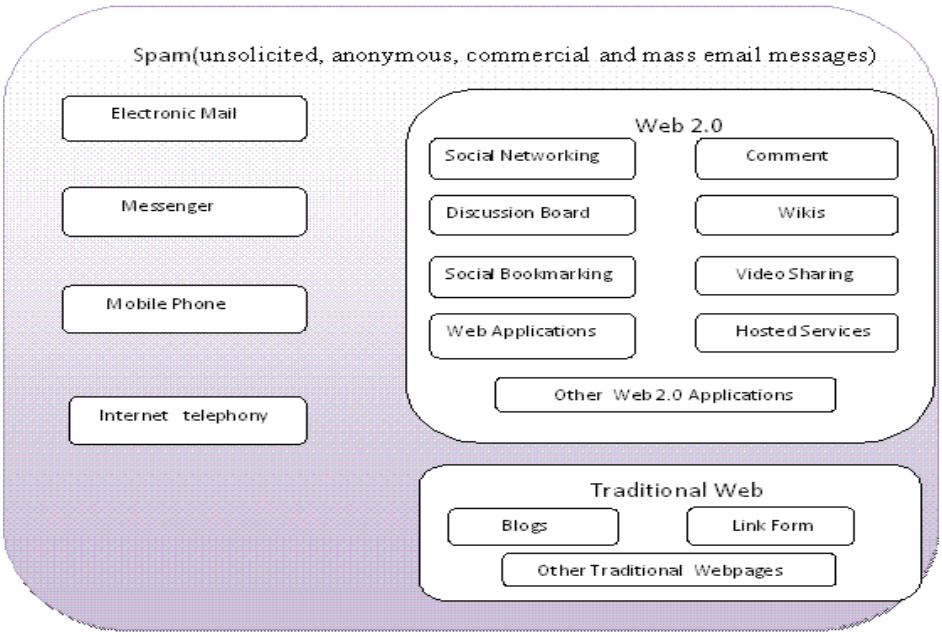


Spamming activities are illegitimate in many (but not every) countries, thus anonymity is one of the most important goals pursued by a spammer. Figure shows the actions undertaken by a spammer to gain anonymity when sending e-mails. A common technique to hide traces of malicious activities is the use of an open relay, that is an SMTP server which does not need authentication to forward e-mail messages. The open relay is contacted by the spammer, receives the e-mail message, and forwards it to the destination SMTP server. This path is detailed through crosshatch arrows in Figure . In theory, a single open relay is sufficient to provide anonymity because every SMTP request appears to be coming from the open relay IP address. However, if a spammer contacts an open relay that is configured to log any activity, the IP address of the spamming machine gets stored in the open relay log files. Using a single open relay is considered a "risky" operation for the spammer. For this reason, spammers often use one or more open proxies (which form a proxy chain) to hide their activities. An open proxy is an intermediary node that forwards traffic between a client and a server without the need of authentication. As shown by straight-arrow path in Figure , the spammer establishes a TCP connection with the first open proxy (*open proxy 1*). Next, the spammer uses this connection to establish a new proxy connection with another open proxy (e.g., *open proxy 2*) through the *CONNECT* method. In this way, a chain of proxy connections is built, until *open proxy n*. Finally, the spammer uses the proxy chain to send an e-mail message to an open relay, which is forwarded to the destination SMTP server. The use of open proxy chains makes it much more difficult to trace spammers back, because, even if the logs of every open proxy are available, the whole path of requests has to be reconstructed [14].

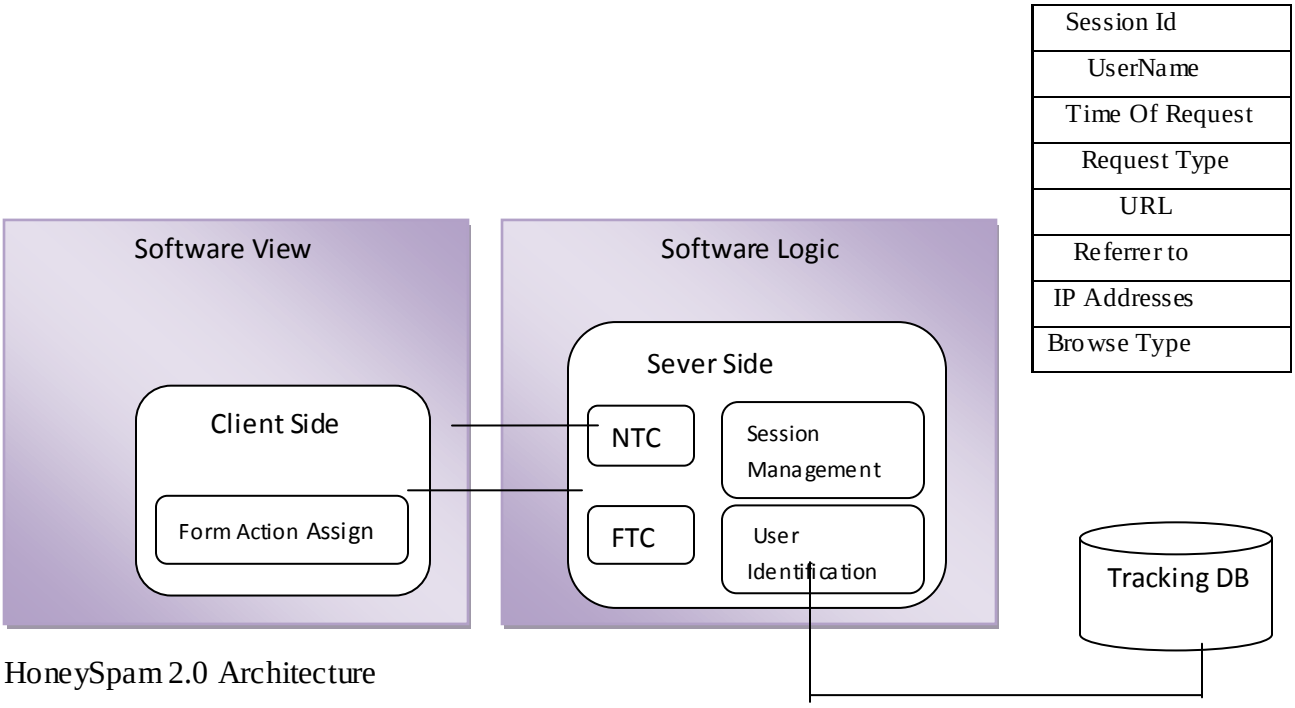
HoneySpam In Search Engine : HoneySpam 2.0

Web content which provides false or unsolicited web information is defined as Web Spam[15]. Spam content is prolific on the web due to the development and widespread adoption of Web 2.0 as spammers no longer need to purchase, host and promote their own domains. Spammers can now use legitimate websites to host spam content such as fake eye-catching profiles in social networking websites, fake promotional reviews, responses to threads in online forums

with unsolicited content and manipulated wiki pages etc. To gain insights into this kind of bot we have developed a tool (HoneySpam 2.0) to track their behavior[16].



HoneySpam 2.0 draws on the idea of honeypots (a technique for tracking cyber attackers). It implicitly tracks web usage data which includes detailed monitoring of click streams, pages navigation, forms, mouse movements, keyboard actions and page scrolling. Our results illustrate that HoneySpam 2.0 is effective in profiling web spambots. HoneySpam 2.0 is a tracking solution that can be integrated in any web application ranging from blogging tools to social networking applications.[17]



HoneySpam 2.0 Architecture

HoneySpam 2.0 is a stand-alone tool that is designed to be integrated with any web application with the aim to track and analyze web spambot behaviour. HoneySpam 2.0 is based upon the idea of a honeypot where a vulnerable server is deployed to attract cyber attackers in order to study their behaviour. HoneySpam 2.0 is designed to study the web spambots which are considered to be attackers in the spamming scenario. HoneySpam 2.0 consists of two main components: Navigation Tracking Component (NTC) & Form Tracking Component (FTC). Figure illustrates the detailed Model- View-Control (MVC) architecture of HoneySpam 2.0 inside a web application.

Navigation Tracking Component

This component is designed to track page navigation patterns and header information of incoming traffic. The following information is captured by this component: the time of request, the IP address, the browser identity and the referrer URL. NTC is also designed to automatically distinguish individual visit sessions.

Session information is captured to identify user's web usage behaviour during each visit. NTC stores the captured data along with session identity (session id) into the tracking database, which is used during an analysis and profiling stage.

Form Tracking Component

Other than the NTC, we felt the need to develop additional functionality that could provide us insights into how web spambots use web forms. It is a common belief that web spambots do not interact with web forms, however, our study aims to test this here. We developed a form tracking component to specifically study form input behaviour. This component captures and stores the following form related actions such as:

- Mouse clicks and movements
- Keyboard actions
- Form field focus and un-focus
- Form load and submission
- Page navigation

For each of the above mentioned actions we also captured header information e.g. IP address, browser identity, referrer link session id etc. We decided to store header information since we wanted to ensure that form action events are originating from the same session with the same header information. Additionally, both components check whether requests are initiated from registered users or visitors. If it was initiated from a registered user then it stores the username along with other web usage information for that request. This was done to differentiate registered web users usage data and visitor usage data, which can be used for further analysis. Visitors are termed as guests in our database. Also, most web applications require the registration of a valid username before further interaction with the system.

Approaches for Detecting and Preventing Spam

1. Spam Content Identification and Filtration

This approach is one of the first approaches developed and implemented to manage spam. The basic idea here is to identify and filter out spam content from genuine content. To achieve this, several techniques have been proposed in the literature and many of them are currently implemented in commercial anti-spam toolkits. For example, detection methods can be

categorized into two: content based & metadata based. The former uses content to analyze spam and hence is computationally intensive and more reliable whereas the latter only uses metadata i.e. links or url or email headers and is relatively fast but comparatively less reliable. Both approaches rely on data mining techniques which can either be supervised, semi-supervised or unsupervised. Supervised methods require a labeled data set for spam classification whereas unsupervised do not [18]. Some anti-spam methods are language dependent and hence may not be able to apply to non English language spam, which can be a problem.

2. Users Flag Content as Spam

This approach is a detection strategy, where the end users are involved in helping to fight spam. This feature is commonly seen in free email services like Yahoo Mail, Hotmail or Gmail etc., where the users have the option to select an email and tag it as Spam. Lately this feature has also become popular in blogs and forums. This feature is very good, as it can help the anti-spam detection algorithms to build up a spam data set; however, the downside of this approach is that spammers can equally use this feature to tag genuine content as spam. So studying the effectiveness of this method is very interesting. Currently there are no publicly available results to show whether this strategy is working [19,20].

3. CAPTCHA

It is a prevention approach that was developed to defeat spambots by requesting spambots to go through an online test named as CAPTCHA. The aim of this test was to distinguish human users from spambots. The test requires the user to type in unclear, curvy or ghostly characters from in an image to a registration form. Most users should be able to surpass this test easily but bots would fail even if they use optical character recognition techniques. CAPTCHA is used in almost all commercial emails sites (Yahoo, Google), online forum, blogs, and social networking sites to prevent automated registrations. CAPTCHA also helps bloggers in dealing with comment spam. However there are some drawbacks e.g.:

- it relies on human visibility, hence it is inconvenient for users with bad vision.
- at times it is even very difficult for normal users to decipher the CAPTCHA.
- free CAPTCHA servers incur longer delays in processing
- many spammers have developed OCR techniques to automatically read CAPTCHA.
- as Optical Character Recognition (OCR) techniques improve, CAPTCHA's images become harder and hard to decipher even by humans. This damages the typical users web experience.
- as computers get more powerful, they will be able to decipher image and voice CAPTCHA requests similar to humans.

4. Poisoning Spammers Database (attack approach)

This is a relatively new approach to address spam. The basic idea is to infiltrate spammers' database and poison it with fake email address, with an aim to reduce the effectiveness of spamming campaigns. So instead of waiting for the spammers to attack, this method takes an active approach towards attacking spammers. This method generates random email addresses and waits for bots to index them. Once the spammer realizes that their database is full of invalid

information, it will reduce the effectiveness of their spam campaigns. However the main concern is that whether the list of random emails generated are really invalid or they do belong to someone. An additional concern is that spammers may not bother if their database is poisoned, they may just send spam to all the emails addresses they have, since they are not using their resources any way. WPOISON and SUGARPLUM are examples of such services available on the Internet.

Conclusion

In this paper, we have addresses how honeypots can be instrumental in detecting spam. We have also discussed the approaches that could be used to fight against spam. However, spam detection is still in its infancy and holds agood future for research.

References:

[1] Honeypots : Definitions and Value of Honeypots by Lance Spitzner
http://www.enteract.com/~lspitz
[2] Lance Spitzner, "Honeypots Tracking Hackers", 2003, Pearson Education, Inc
[3] Crytography and Information security by v.k.Pachghare
[4] Cyber Security Essentials by Rick Howard
[5] Reto Baumann and Christian Plattner, "White Paper: Honeypots", 26 February 2002, URL: http://www.inf.ethz.ch/~plattner/pdf/whitepaper.pdf
[6] Klug D., 13th September 2000, HoneyPots and Intrusion Detection[online] SANS Institute, Available: http://www.sans.org/infosecFAQ/intrusion/honeypots.htm

[7] Graham R., March 20th 2001, FAQ: Network Intrusion Detection Systems [online]Available: http://www.robertgraham.com/pubs/network-intrusion-detection.html

[8] CENTER FOR DEMOCRACY & TECHNOLOGY. Why Am I Getting All This Spam?, Mar 2003. http://www.cdt.org/speech/spam/030319spamreport.pdf.

[9] UNSPAM, LLC. Project Honey Pot, 2005. http://www.projecthoneypot.org.

[10] OUDOT, L. Fighting Spammers With Honeypots: Part 1 and 2, Nov 2003.
http://www.securityfocus.com/infocus/1747.
[11] PROVOS, N. A Virtual Honeypot Framework. In *Proc. of 13th USENIX Security Symposium* (Aug 2004).
[12] Mauro Andreolini, HoneySpam: Honeypots fighting spam at the source
http://static.usenix.org/event/sruti05/tech/full_papers/andreolini/andreolini.pdf
[13] PRINCE, M., KELLER, A. M., AND DAHL, B. No-Email- Collection Flag. In *Proceedings of the First Conference on Email and Anti-Spam (CEAS)* (Jul 2004).
[14] THE ATOMINTERSOFT TEAM. AliveProxy, 2005. http://www.aliveproxy.com/.
[15] Gyongyi, Z., Garcia-Molina, H.: Web spam taxonomy. In: Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web, Chiba, Japan (2005).
[16] Hayati, P., Potdar, V.: Toward Spam 2.0: An Evaluation of Web 2.0 Anti-Spam Methods.
In: 7th IEEE International Conference on Industrial Informatics Cardiff, Wales (2009).
[17] Pedram Hayati,HoneySpam 2.0: Profiling Web Spambot Behaviour
http://kevinchai.net/wp-content/uploads/2011/06/honeyspam-2.0-profiling-web-spambot-behaviour.pdf
[18] Hayati, P., Potdar, V.: Toward Spam 2.0: An Evaluation of Web 2.0 Anti-Spam Methods.
In: 7th IEEE International Conference on Industrial Informatics (INDIN 2009), Cardiff, Wales, (2009)
[19] Sheffield, M.: 'Flag Spam,' the Preferred Tool of the Left's Web Censors. 2008 [cited 14,];
2008/10/07/flag-spam latest-tool-censors-left (July 2009),
http://newsbusters.org/blogs/matthewsheffield/
[20] userscripts.org. Flagging Content Feature. [cited 14 July 2009]
http://userscripts.org/topics/1362