COMPUTERS & SECURITY 26 (2007) 361–372

available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

The impact that placing email addresses on the Internet has on the receipt of spam: An empirical analysis

Guido Schryen

RWTH Aachen University, Institute of Business Information Systems, 52062 Aachen, Germany

ARTICLE INFO

Article history:
Received 5 September 2006
Revised 13 December 2006
Accepted 22 December 2006

Keywords:
Address obfuscating techniques
Email
Empirical analysis
Honeypot
Security by design
Security by obscurity
Spam

ABSTRACT

Email communication is encumbered with a mass of email messages which their recipients have neither requested nor require. Even worse, the impacts of these messages are far from being simply an annoyance, as they also involve economic damage. This manuscript examines the resource "email addresses", which is vital for any potential bulk mailer and spammer. Both a methodology and a honeypot conceptualization for implementing an empirical analysis of the usage of email addresses placed on the Internet are proposed here. Their objective is to assess, on a quantitative basis, the extent of the current harassment and its development over time. This "framework" is intended to be extensible to measuring the effectiveness of address obscuring techniques. The implementation of a pilot honeypot is described, which led to key findings, some of them being: (1) Web placements attract more than two-thirds (70%) of all honeypot spam emails, followed by newsgroup placements (28.6%) and newsletter subscriptions (1.4%). (2) The proportions of spam relating to the email addresses' top-level domain can be statistically assumed to be uniformly distributed. (3) More than 43% of addresses on the web have been abused, whereas about 27% was the case for addresses on newsgroups and only about 4% was the case for addresses used for a newsletter subscription. (4) Regarding the development of email addresses' attractiveness for spammers over time, the service "web sites" features a negative linear relationship, whereas the service "Usenet" shows a negative exponential relationship. (5) Only 1.54% of the spam emails showed an interrelation between the topic of the spam email and that of the location where the recipient's address was placed, so that spammers are assumed to send their emails in a "context insensitive" manner. The results of the empirical analysis motivate the need for the protection of email addresses through obscuration. We analyze this need by formulating requirements for address obscuring techniques and we reveal to which extent today's most relevant approaches fulfill these requirements.

1. Introduction

With the usage of the term "spam", the Internet community seemingly describes in consensus the ubiquitous phenomenon of receiving Unsolicited Bulk Email (UBE), as specified in Spamhaus. Although this might suggest a precise and full apprehending of this kind of Internet abuse, different semantics of the terms "unsolicited" and "bulk" can, however, be found (Faigin and Bishop, 2003). Furthermore, definition refinements and specializations lead to divergent ontologies. For example, spam is (either implicitly or explicitly) associated with a commercial context and is then referred to as

E-mail address: schryen@winfor.rwth-aachen.de
0167-4048/$ – see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cose.2006.12.009

Unsolicited Commercial Email (UCE) (CAUCE), thereby excluding many other possible forms of appearance such as fraud emails, phishing emails and chain letters. Many legislative and regulative activities against UBE have resulted in laws and policies which are mainly dedicated to UCE (e.g. the German UWG, Bundestag; the U.S. CAN-Spam Act, Eighth Congress, 2003; and the Directive 2002/58/EC of the European Union, European Parliament, 2002). Another option for refining the definition of spam is the requirement that "*the transmission and reception of the message appears to the recipient to give a disproportionate benefit to the sender*" (MAPS). This diversity of definition indicates a terminological fuzziness, which we are still facing in the spam discussion.

A second divergence arises when the emergence of spam is quantified and measured. Several market research institutions and security companies periodically publish data about spam specifying the total number and the proportion of spam emails. In addition to the various definitions of spam, methodological variations and different sample sizes are factors for (divergent) results (OECD, 2003a). For example, MessageLabs reports that, in June 2005, the proportion of spam reached 67.25% (MessageLabs, 2005), however, Symantec announces the very different figure of 53% for the same month (Symantec).

Regardless of any precise definition and of the proportion of spam, private users' and companies' daily experience is that email communication is encumbered with a mass of email messages which their recipients have neither requested nor require. Even worse, the impacts of these messages are far from being simply an annoyance, as they also involve economic damage: The increased payload of networks and email servers, the consumption of employees' attention and time, fraud, and the spread of viruses, worms, and Trojan horses are just a few examples of the harm involved. The economic damage caused in total by spam emails is estimated at several billion US$ (OECD, 2003b).

Many different anti-spam measures have evolved and are deployed. Laws and regulations (Opt-in vs Opt-out), economic approaches, and technological measures (Schryen, 2004) – including filters and authentication mechanisms – provide today's most important anti-spam leverages. They address three conditions, two of which must be fulfilled by bulk mailers (motivation and capability). The third condition refers to the legal permission some bulk mailers are grasping at in order to avoid litigation. Fig. 1 illustrates the relationship between anti-spam measures and both the intrinsic as well as the extrinsic factors for sending bulk email (which is legally allowed).

This manuscript examines the resource "email addresses" which is vital for any potential bulk mailer and spammer. Section 2 motivates the empirical analysis of the impact that placing email addresses on the Internet has on the receipt of spam, and frames the analysis' goals. Section 3 proposes a methodological framework for using a honeypot for the intended exploration. Section 4 briefly reports on a pilot implementation of the honeypot's key modules. Section 5 presents empirical results of the honeypot's application and reports on today's usage of email addresses that have been placed on the Internet. Finally, in Section 6, we analyze the motivated need for the protection of email addresses through

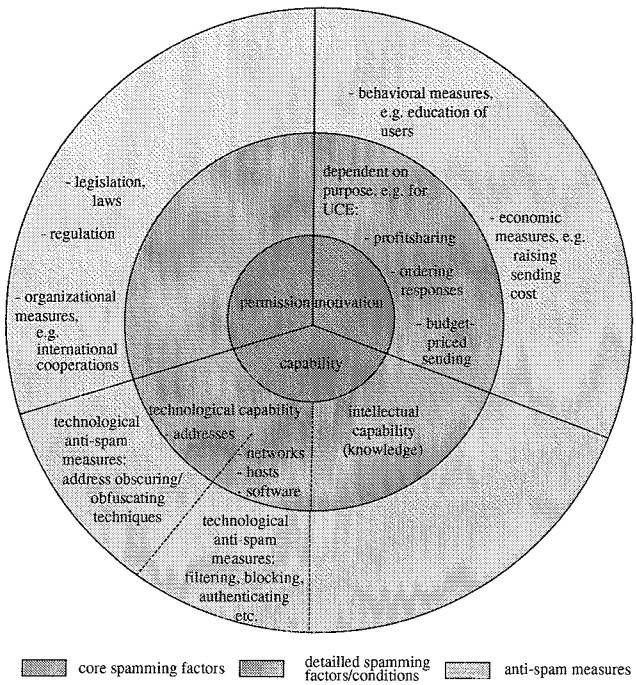

Fig. 1 – Spamming factors and their relationship to anti-spam measures.

obscuration by formulating requirements for address obscuring techniques, by presenting today's most relevant techniques, and by matching these techniques with the proposed requirements.

2. Motivation, recent studies, and goals

Valid email addresses are among the most valuable resources for spammers, and the identification of address sources and spammers' exploiting procedures is crucial to preventing spammers from procuring addresses and subsequently misusing them. It is widely known that, besides generating addresses with brute force mechanisms and dictionary attacks, spammers procure valid email addresses by harvesting the Internet or, illegally, by purchasing or stealing them from various organizations. However, only little is known about the quantitative properties of email address abuse on the Internet and how to measure these. Gaining insight into this field allows for assessing the extent of the current harassment and its development over time and implicates the development of a test framework which also allows for measuring the effectiveness of address obscuring techniques, such as the embedding of addresses into images or the "masking" of addresses textually, e.g. by using an address such as `aliceREMOVETHISTEXT@wonderland.tv` (see Section 6). The framework's scope might also include the empirical assessment of such obfuscating techniques which restrict addresses' usability, e.g. by using single-purpose addresses (Ioannidis, 2003) or email aliases (Gburzynski and Maitan, 2004).

The author is aware of five empirical studies which focus on the extent of spam harm that is caused by placing email addresses on Internet services:

- In 1999, the Australian Coalition Against Unsolicited Bulk Email (CAUBE.AU) seeded email addresses to the Usenet, to the web and to Internet contact databases. The study CAUBE.AU (1999), which took almost 1 year, focused on spam sources and contents. Regarding the attractiveness of particular services, the study found that "*[...] the effectiveness of an email address exposure [...] is almost identical for posting a single message to USENET as it is for posting the address to a single web page.*"
- In 2002, the US Federal Trade Commission (FTC) seeded 175 different locations on the Internet (including web pages, newsgroups, chat rooms, message boards, and online directories for web pages, instant message users, domain names, resumes, and dating services) with 250 new, undercover email addresses (FTC, 2002). During the six weeks after the postings, the key findings were:
  - "*86 percent of the addresses posted to web pages received spam. It didn't matter where the addresses were posted on the page.*
  - *86 percent of the addresses posted to newsgroups received spam.*
  - *Chat rooms are virtual magnets for harvesting software. One address posted in a chat room received spam nine minutes after it first was used.*
  - *Addresses posted in other areas on the Internet received less spam, the investigators found. Half the addresses posted on free personal web page services received spam, as did 27 percent of addresses posted to message boards and nine percent of addresses listed in email service directories. Addresses posted in instant message service user profiles, 'Whois' domain name registries, online resume services, and online dating services did not receive any spam during the six weeks of the investigation.*
  - *In almost all instances, the investigators found, the spam received was not related to the address used. As a result, consumers who use email are exposed to a variety of spam – including objectionable messages – no matter the source of the address.*"
- In 2002, the Center for Democracy and Technology embarked on a project (Center for Democracy and Technology, 2003) to attempt to determine the source of spam. Hundreds of different email addresses were set up, which led to the major findings that (1) "*[...] e-mail addresses posted on Web sites or in newsgroups attract the most spam.*", (2) "*For the most part, companies that offered users a choice about receiving commercial e-mails respected that choice.*", (3) "*Some spam is generated through attacks on mail servers, methods that don't rely on the collection of e-mail addresses at all.*"
- The "Project Honey Pot" (www.projecthoneypot.org) is a distributed honeypot network to track email harvesters and the spammers who send to harvested addresses. It was opened to public volunteers in October 2004 and, as of June 20, 2005 the project is monitoring more than 250,000 active spamtrap email honeypots. The core idea is to provide a honeypot software to be installed on web servers by administrators, and to collect data about address harvesters (from these servers) and about spam emails received on harvested addresses (from assigned email servers). The collected data are stored and processed on a central honeypot server. The technological background as well as an analysis of the data collected during the first six month is provided in Prince et al. (2005). The empirical results comprise the following findings:
  - "*Approximately 6.5 percent of the traffic visiting our honey pots subsequently turns out to be spam harvesters.*"
  - "*The average time from a spamtrap address being harvested to when it receives its first message is currently 11 days, 7 hours, 43 minutes, and 10 seconds.*"
  - "*[...] we have characterized two distinct classes of harvesters. [...] The first class – the hucksters – are characterized by a slow turnaround from harvest to first message (typically at least 1 month), a large number of messages being sent to each harvested spamtrap address, and typical product-based spam [...]. The second class – the fraudsters – are characterized by an almost immediate turnaround from harvest to first message (typically less than 12 hours), only a small number of messages sent to each harvested spamtrap address, and fraud-based spam [...].*"
- The FTC conducted a study (FTC, 2005) in 2005 which explored the current state of email address harvesting, the effectiveness of anti-spam filters and the effectiveness of using masked email addresses. In the course of 3 days, 150 email addresses were posted to 50 Internet locations in total, consisting, in each case, of 12 in the category "FTC web page", "message boards", "blogs", and "chat rooms", respectively, and two in Usenet groups. One key finding of the study – which lasted five weeks – regarding the attractiveness of categories for harvesters, is that "*[...] 99.6 percent of the total amount of spam received were received by Unfiltered Addresses that had been posted on 11 of the 12 web pages, [...]*" (FTC, 2005, p. 4). This study indicates that spammers continue to harvest addresses posted on Internet locations.

The studies differ in their goals as well as in their (methodological) framework and implementation, e.g. there are differences in the analysis periods, the number of seeded addresses, the number and kind of locations used, and the categories considered. This must be taken into account when comparing results.

All studies share the result that the extent to which email addresses are harvested and misused for spamming is considerable. This significance stresses the necessity of preventing or reducing the harvesting of email addresses placed on the Internet and motivates both the development of address obfuscating techniques and (the deployment of a framework which supports) empirical studies which serve as a "controlling instrument".

The goals of this paper are (1) to propose a honeypot concept and a methodology which allows for the systematic implementation of an empirical analysis of the usage of email addresses that have been placed on the Internet, (2) to discuss important implementation issues by simultaneously presenting a pilot implementation, (3), by using statistical procedures, to present key findings of today's spam harassment which emanates from address placements on the Internet, and (4)

to propose requirements for address obscuring techniques and to show to which extent today's most relevant techniques meet these requirements. The pilot study addresses issues such as:

- the relative and absolute attractiveness of particular Internet services,
- the development of email addresses' attractiveness over time,
- the relevance of an email address' top-level domain,
- differences in the seeding of addresses at language-specific locations, and
- the relationship between the content of emails and the locations on which the recipients' addresses were placed.

The proposed "framework" is intended to be extensible to measuring the effectiveness of address obscuring techniques.

3. Methodology and honeypot conceptualization

The conceptualization and the methods used for the planning, implementation, and evaluation of a honeypot, which addresses the impact that placing email addresses on the Internet has on the receipt of spam, have to adhere, of course, to the honeypot objectives and questions to be answered. In addition to this, the collected emails provide a large data volume which is open to discovery of unknown patterns (data mining). This issue has to be especially considered in data modeling.

The conceptualization of the honeypot comprises:

- the selection of appropriate Internet locations as well as email addresses to be seeded,
- the development of proper data and database models, and
- the selection and application of evaluation procedures.

Internet locations can be categorized by the use of the dimensions "service", "language", and "topic", as illustrated in Fig. 2.

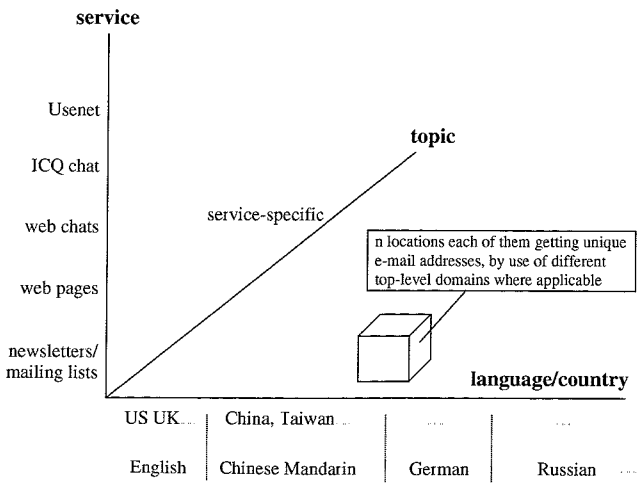

Fig. 2 – Categories of Internet locations.

There is a broad range of services, which include email addresses and which are open to harvesters in principle (Raz), such as web pages, chats, and newsgroups. Regarding the placement of email addresses, the services differ in many ways. For example, web pages permissively allow the seeding of textual email addresses as well as addresses which are embedded in a graphic (here referred to as the "representation form" of an email address), whereas newsletters and mailing lists are limited to textual addresses, and administrators of some newsgroups do not permit the placement of email addresses in the body of an article at all. A further dimension is spanned by the languages and countries involved in the empirical study. The Internet locations can also be categorized according to the topic they are dedicated to. The classification of web pages and newsletter/mailing lists, for example, can follow any e-business classification (possible topics are "education", "auctions", "logistics", etc.). Newsgroups can be classified according to the topics they are dedicated to and which are reflected in the newsgroup's name. Depending on the study's objectives, it might be desirable to define the topics service-specifically. After defining the categories for address placement, one or several locations per category can be selected. Finally, the type of addresses to be seeded has to be defined. This relates to the email addresses' top-level domain as well as to the representation form of the address. In order to trace back spam emails, it is necessary to use unique email addresses which are, ideally, invisible to users and thus transparent to harvesters only.

Emails can be stored in a flat file or in a database, the latter facilitating data analysis. If a database is selected as storage, then it should be noted that the data and database model decisively affect the analysis options and the time which evaluation procedures take. For example, if an email's envelope is discarded, its "MAIL FROM" parameter might be lost, or the storage of the body of an email as a binary large object (BLOB) does not allow an (efficient) determination of the email's multipurpose Internet mail extension (MIME) structure. The provision of a reference data model is desirable but beyond the scope of this work.

Although the selection of the evaluation procedures depends on the study's goal it might also be desirable to provide a methodological framework. This is likewise beyond the scope of this paper. However, Section 5 (empirical results) presents some analysis methods and particularities which were respectively used and taken into consideration when evaluating the data on the honeypot's pilot implementation.

4. Implementation considerations

This section describes the implementation and the exemplary application of the honeypot (conceptualization). Services included are US as well as German "web pages" and "newsletters", and German-speaking as well as English-speaking "Usenet groups". The topics of the first two services are listed in Schryen (2005), which also presents some preliminary results. The topics and names of the 21 Usenet groups are listed in the appendix.

A mail server has been set up, namely charlie.winfor.rwth-aachen.de, and three domains have been reserved – wforasp.com,

wforasp.net, and wforasp.org – for covering the email addresses of four top-level domains. All emails addressed to these domains are directed to this mail server. As thousands of email addresses had to be created, they were generated automatically by using a random generator for the user part of the addresses. In order to prevent email addresses from being guessed or generated with brute force attacks, it is necessary to define them randomly as well as to give them an appropriate number of characters. An example of an email address created in this way is wasp10208@wforasp.com.

The Internet locations serving as lures were chosen manually just as the placement of the email addresses had to be implemented manually. As soon as an email address is spread, its location and activation date are stored.

All incoming emails are classified into regular emails (ham emails), e.g. regular newsletters or such containing comments from users of discussion forums, and spam emails. This procedure was mainly executed by humans but supported by a mail parser (written in PHP) which used increasing white lists and blacklists. A second task of the mail parser was to decompose each incoming email: all entries of the header and the content were analyzed, as was the (MIME) structure of the body. Next, the emails' elements are stored in a (MySQL) database broken down into spam and ham emails. As many spam emails are not RFC-compliant, the parser's robustness against RFC violations was one of the implementation goals. Fig. 3 provides a survey of the implementation infrastructure.

Simple data analysis was undertaken by using SQL queries, whereas more complex procedures were conducted by the use of Microsoft Excel.

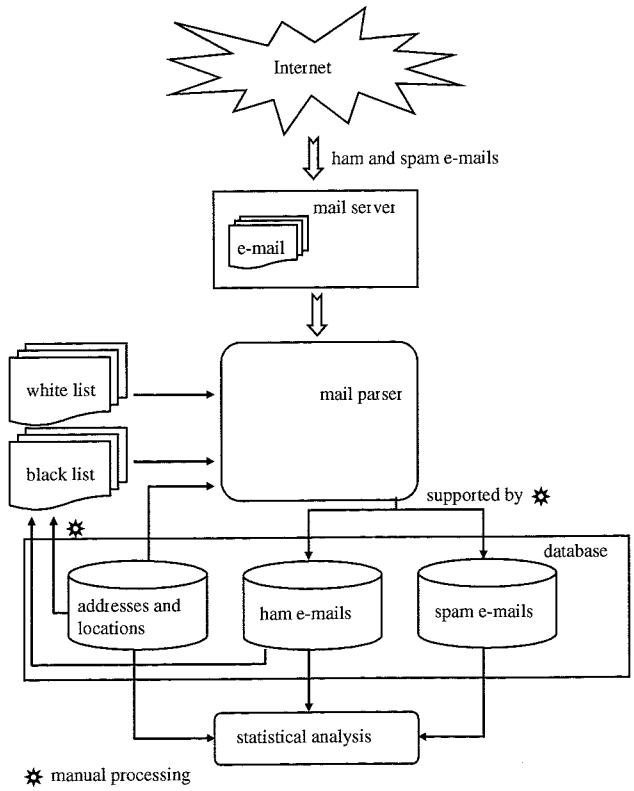

Fig. 3 – Infrastructure of the prototypic honeypot environment.

5. Empirical results

The total number of emails received on the honeypot is 57,273, 47% (26,882) of which is spam. Of all spam emails received on the email server, 69.9% (18,792) result from placements on the Internet (denoted as honeypot spam emails), whereas the others – directed to addresses which have never been generated and which have been placed nowhere, such as `admin@wforasp.com` – are due to spammers' guessing attempts, e.g. by the use of brute force or dictionary attacks. The number of placed email addresses and their respective residence periods underlying this evaluation differ between the Internet services; only emails received before January 31, 2006, 12 p.m. MET have been considered. The average time period for email addresses placed on the web is almost a year, as is the time period for addresses used for newsletter subscriptions. The time period related to newsgroup placements is approximately half of that. Table 1 shows some statistical details of the email addresses placed on the Internet services.

Most honeypot spam emails result from web placements (62.3% of all honeypot spam emails), followed by spam caused by newsgroup placements (6.3%) and to newsletter subscriptions (1.27%). However, as the number of placed emails as well as their online time periods varies, these proportions do not precisely reflect the services' attractiveness for harvesters. To take these two issues into account, the number of emails received on each service is weighted by using the total number of online days of all emails placed on the corresponding service. Let $sp_i, i \in \{w, ng, nl\} =: S$ be the number of spam emails received on placements on the service web, newsgroups, and newsletters (see below), and let $od_i, i \in S$ be the total number of online days of all email addresses placed on service i. Then, the weighted number of received spam emails is calculated by:

$$sp'_i = sp_i \frac{\sum_{j \in S} od_j}{od_i}, \quad i \in S \tag{1}$$

The computation in Eq. (1) is time-invariant in that all online days are same weighted.

Table 2 shows the results which indicate that web placements attract more than two-thirds (70%) of all honeypot spam emails, followed by newsgroup placements (28.6%) and newsletter subscriptions (1.4%) – the latter hardly leading to the receiving of spam emails. Language-specific proportions do not considerably differ from the proportions in total.

The honeypot also allows for analyzing the relevance of email address' top-level domain (TLD) to receiving spam. Table 3 shows the empirical data. Proportions do not have to be weighted according to online days, because, at each Internet location, one address of each TLD was placed at the same time.

Trying to reject the null hypothesis that the empirical proportion of spam sent to email addresses which were placed on the Internet follows a discrete uniform distribution we use the chi-square test. We compute:

$$\chi^2 = \sum_{i=1}^{4} \frac{(q_i - 4688.5)^2}{4688.5} \approx 880.13, \tag{2}$$

and compare this value with the 0.01 critical value from the chi-square distribution with 3 degrees of freedom,

Table 1 – Service-specific residence periods of email address placements

Service	Number of placed email addresses	Residence periods	
		Empirical mean	Empirical Standard deviation
web	917	318.99	75.66
newsgroups	390	186.38	37.17
newsletter	848	361.62	67.39
Total	2155	–	–

$$\chi^2_{p=0.01,\ df=3} \approx 12.84. \quad (3)$$

As $\chi^2 > \chi^2_{p=0.05,df=3}$, the null hypothesis has to be rejected on significance level 0.01. Therefore we cannot assume the proportions to be uniformly distributed.

Interestingly, the empirical data regarding spam on email addresses which were not placed on the Internet differ from the data considered above, in that spam emails directed to "org" addresses amount to almost 85%. Brute force and dictionary attacks seem to focus on email addresses with the TLD "org".

When we look at the extent to which email addresses placed on the web have been flooded with spam, we find that more than 43% of addresses on the web have been abused, whereas about 27% was the case for addresses on newsgroups and only about 4% for addresses used for a newsletter subscription. Table 4 illustrates detailed data about this issue. The service instances, i.e. the names of web sites, newsgroups and newsletters, where those email addresses were placed, which attracted most spam emails, are listed in the appendix together with the respective number of spam emails received.

The development of email addresses' attractiveness for spammers over time (see Fig. 4) can be analyzed by regression analysis; weeks without spam emails were omitted. We find a negative linear relationship for the service "web sites" with a coefficient of determination r^2 of approximately 0.86. The Pearson coefficient r is approximately −0.93, which strongly indicates a negative linear relationship. Assuming a negative exponential relationship for the service "newsgroups", we get a coefficient of determination of approximately 0.87. Performing a logarithmic transformation of the data, we again look for a negative linear relationship. The Pearson coefficient of the transformed data is approximately −0.96, which, then, finally supports strongly the assumption of a negative exponential relationship of the original data. A regression analysis for the service "newsletter" does not appear to be reasonable due to the low number of spam emails received.

The honeypot also allows for checking the relationship between a spam email's topic and the topic of the location at which the recipient's address was placed. We manually checked 3500 spam emails in "first come first served" order and found only 53 emails (1.54%) which shared a topic. Therefore, we suppose that spammers do not to send their emails in a "context sensitive" manner.

6. The pertinence of address obscuring techniques

Address obfuscating/obscuring techniques (AOTs) belong to those technological anti-spam measures that aim at protecting email addresses from being automatically harvested, and thus addresses spammers' technological capabilities (see Fig. 1). These measures are not meant to be deployed as a substitute for other technological anti-spam measures, such as those using filters or authentication mechanisms. Rather, their implementation is intended to be complementary to that of other measures. In contrast with many other technological anti-spam approaches, AOTs are primarily targeted at the prevention of sending spam emails and not at their detection. The

Table 2 – Empirical statistics for the service- and language-specific abuse of email address placements

Service i	Language	sp_i	od_i	$od_i / \sum_{j \in S} od_j$ [a] (%)	sp'_i [a] (%)	$sp'_i / \sum_{j \in S} sp'_j$ [a] (%)
web	German	5478	156,212	42.77	12,807	63.81
	English	11,270	136,302	44.45	25,356	69.73
	Total	16,748	292,514	43.54	38,468	70.01
newsgroups	German	965	51,520	14.11	6,840	34.09
	English	737	21,170	6.90	10,676	29.36
	Total	1702	72,690	10.82	15,731	28.63
newsletter	German	182	157,468	43.12	422	2.10
	English	160	149,188	48.65	329	0.91
	Total	342	306,656	45.64	749	1.36
	Total overall	18,792	671,860	–	–	–

Language-specific numbers refer to different (language-specific) main units.
[a] Due to different main units the number in a "total" row does not represent the sum or (weighted) average of the corresponding language-specific numbers.

Table 3 – Spam emails by top-level domain of abused email address

Top-level domain	Spam resulting from email addresses placed		Spam resulting from email addresses not placed	
	Quantity q_i	Proportion (%)	Quantity	Proportion (%)
de	3833	20.45	155	1.92
com	6147	32.80	507	6.27
net	5117	27.30	609	7.53
org	3645	19.45	6815	84.28
total	18,742	–	8086	–

detection of spam emails always entails that (1) sending process has been initiated or even almost completed and (2) resources, such as network bandwidth and detection software, have already been consumed. Therefore, prevention-oriented measures deserve closer attention from researchers as well as Internet organizations, email service providers, and users.

As spammers rely on large sets of valid email addresses, these have to be harvested automatically, either by the spammers or by address suppliers (if we ignore brute force and dictionary attacks, which are inevitable anyway). Previous studies and the empirical study conducted here show the large extent to which email addresses that have been placed on the Internet are (automatically) identified and consecutively misused for spamming.

In order to get an idea of how easy the harvesting of email addresses actually is, on one PC (Pentium 4, CPU 3GHz, RAM 1GB, Windows XP, Service Pack 2) we ran two harvesting tools: one for scanning the web, the other for scanning newsgroups. With regard to the web, we used the shareware "EmailSpider Gold 9.0" with the following (default) basic configuration: starting domain = www.yahoo.com, only scan super domains = com; the advanced configuration was modified by setting the number of parallel scan threads to 50 only. We terminated the search after 16 h, by when we had obtained about 154,000 email addresses (multiple occurrences included), thus reaping an average of 9625 addresses per hour. Assuming a linear relationship between the number of harvested email addresses and time, we would need about 1039 h or about 43.3 days of total computer time in order to collect 10,000,000 email addresses. As a search can be performed in parallel, the usage of n PCs reduces the total time by $1/n$. With regard to newsgroups, we used the shareware "Power Email Extractor 4.1" and its (default) news server "freetext.usenetserver.com". We scanned 1055 newsgroups that were related to computer issues, i.e. all comp.* newsgroups and received about 1 million addresses within 28 min. These addresses are promising in terms of their attractiveness for computer advertisers. The scanning of all "de"-newsgroups (593 newsgroups) took 30 min and resulted in about 1.2 million email addresses. These addresses are likely, owing to the "de" country suffix, to belong to German-speaking users, and therefore, these addresses are promising in terms of their attractiveness for German-speaking advertisers. We further scanned all 46,623 newsgroups on this particular server and obtained more than 12.5 million email addresses within about 15 h. However, the email address pools still contained duplicates. Although we did no quantitative analysis – the reason for this being that we would have had to purchase the software for performing such an analysis – we believe the duplicate factor to be smaller than 10. These experiments demonstrate how quickly and with how few resources email addresses can actually be harvested.

Both the demonstrated ease of harvesting email addresses and results from empirical studies indicate a strong need for the protection of addresses. However, such a remedy does not seem to be straightforward, because we assume that AOTs have to address the following requirements in order to be effective in the long run:

R1. Addresses must still be recognizable and usable for humans.
R2. Email communication (processes) must not become complex.
R3. All locations from which harvesters can easily collect addresses have to be covered, i.e. it is insufficient to protect email addresses on the web while address books and email folders on local PCs remain open to harvesting malware. Raz presents a list of potential email address pools.
R4. Services, such as mailing lists, that require the automatic processing of valid email addresses, need appropriate support for the processing of obscured addresses.
R5. The automatic recognition of valid email addresses must be difficult enough to ensure an effective protection. The difficulty can either rely on today's practical incapability of programs to recognize email addresses or on the computational effort that is necessary for recognition. However, the former approach involves considerable uncertainty about the duration of protection, so that the

Table 4 – Extent, to which email addresses have been abused

Service	Number of placed email addresses	Number of abused email addresses	Abuse proportion	Number of received spam emails				
				Min	Max	Mean	Median	Standard deviation
Web	917	399	43.51%	1	829	41.97	9	96.88
Newsgroups	390	106	27.15%	1	69	16.06	15.5	12.96
Newsletter	848	35	4.13%	1	66	9.77	2	15.78

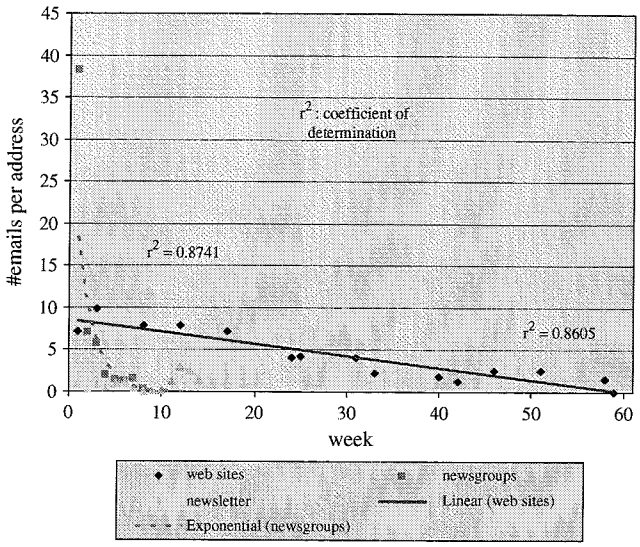

Fig. 4 – Development of email addresses' attractiveness for spammers over time.

implementation of AOTs is in danger of causing an "arms' race" with harvesters.

R6. Given that an obscured email address has been harvested and abused for spamming purposes, the concerned user should not be compelled to provide a new address to all of his/her contacts. We denote this requirement as robustness against harvesting.

Email addresses can be simply obscured by modifying or adding characters, examples are `schryen+gmx.net`, `schryen (at) gmx.net`, and `schryenREMOVETHIS@gmx.net`. Such AOTs work on a textual level only and can, hence, be used at any email address location. Another option for obscuring an email address is the modification of its representation/coding. For example, the ASCII representation of the address `schryen@gmx.net` in a web document is:

```
&#115;&#99;&#104;&#114;&#121;&#101;&#110;&#64;
&#103;&#109;&#120;&#46;&#110;&#101;&#116;
```

The modification of the coding limits the applicability of the particular AOT to those environments that allow an appropriate processing of the representation. Another (web) environment-specific AOT is the integration of comments that are not displayed, for example, the line

```
schryen<!-- This text is intended to confuse harvesters. -->@gmx.net
```

in an HTML document would result in the displaying of the actual email address. Environment-specific AOTs can work more sophisticatedly by using script languages, either on the client-side or on the server-side. This example illustrates the usage of the (client-side) script language JavaScript:

```
<script language=''JavaScript''>
<!--
document.write(''<ahref=\''mailto:''+''schryen''
+''@''+''gmx.net\''>email</a>'');
//-->
```

The following example shows the usage of both client-side JavaScript and server-side PHP script that uses Base64 decoding (http://www.lakebase.com):

```
<script type=''text/javascript''src=''http://
www.lakebase.com/jsmailer.php?v=TVRFMk1UQXhNVEV3T
kRZPU1USXdNVEE1TVRBek5qUT1NVEV3TVRBeE1USXhNVEUwTV
RBME9Uaz1NVEUx&l=TVRBNE1UQTFPVGM9TVRBNU5qaz0=''>
</script>;
```

Some users obscure their email address by replacing the textual address by an image that contains the address. This approach is motivated by the fact that the human brain is much better at visual processing than even powerful computers are (see von Ahn et al., 2003 and von Ahn et al., 2004, which present Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) algorithms).

The sketched approaches have the advantage of being easy to implement and of not complicating email communication processes (R2). Furthermore, they are still recognizable for human users (R1). The extent to which the obscured addresses are appropriate for automatic processing (R4) depends on the particular AOT in use. Most approaches are intended to be applied on the web, therefore, they are limited in their deployment in other email storages (R3). What all approaches have in common is that they hide the method by which the addresses are being obscured. Once the method becomes known to the public, a new method has to be chosen and implemented, which results in additional efforts. As early as 1883, this principle was already regarded as inadequate in the context of cryptology (Kerckhoffs' principle, Kerckhoffs, 1883): a cryptosystem should be secure even if everything about the system, except the key, is public knowledge. In contrast to this principle, which follows the idea of "security by design", the proposed methods are based on "security by obscurity", thus violating R5. Obscured email addresses are not designed to work as recipient-specific addresses. This missing property leads to a lack of robustness against harvesting (R6).

Another group of AOT algorithms follows the "security by design" principle. Some of the most discussed ones are Hall's virtual channels (Hall, 1996), extended email addresses as proposed by Gabber et al. (1998), single-purpose addresses introduced by Ioannidis (2003) and the similar concept "Tagged Message Delivery Agent".

Hall (1996) proposed a virtual channel concept that is applied to selectively sharing email addresses, each of them being associated with one virtual channel. Essentially, each user's email account is made accessible via a user-controlled set of channels. Each channel has a distinctly structured address which contains within it the account name and a cryptographically secure, i.e. unguessable, pseudorandom security string, known as a channel identifier. Each legitimate correspondent is allowed to know one of these channel addresses. The account owner is provided with simple controls for opening a new channel, closing a channel, and switching a channel by notifying selected correspondents of a new channel that is replacing the current one. A channelized address is an email address of the form `Username-ChannelID-@Host`, e.g. `alice-1xyz6u9uz4-@wonderland.com`. The channel ID contains a channel class indicator (1) and a security string (`xyz6u9uz4`). The security string is built by generating 45 bits pseudorandomly and using "Base32" encoding to form nine characters. If, for example, an email user wants to share $2^7 = 128$ channels, an adversary has one chance in 2^{45-7} (about 275 billion) of correctly guessing an open channel with one

message. The channel class indicator consists of one digit. This digit allows differentiation between a send-only channel, which is useful when one wants to send a message to a public address without receiving emails at this address (i.e. permanently closed to everyone), a private channel, which is open to emails from pre-determined senders (emails from other persons may be ignored on such a channel), and a public channel (permanently open to everyone). Hall (1996) proposes an even richer class system: the maintenance of channels (i.e. generating, distributing, deleting etc.) is intended to be handled by a "Personal Channel Agent". For the sake of effective email address protection, it is essential to keep channel identifiers secret. However, this requirement seems more than challenging in a world where many PCs are infected with malware that can read the entries of their local address books.

Gabber et al. (1998) suggest a similar concept which is based on extended email addresses and also aims at hiding them. An extended email address for Alice would be `Alice+xV78Yjkpl9@wonderland.com` with `xV78Yjkpl9` being the extension; the address `alice@wonderland.com` is denoted as the "core address". The extension will be calculated as e(`Alice@wonderland.com,Bob@jungle.com;`n_{Bob}) with e being a function which is not specified but described in terms of requirements and n_{Bob} being a user-specific counter (with the initial value 0). Each time Bob gets a new extended address – maybe because the current address has been incautiously forwarded by Bob to someone else or it has been read by an address harvester – the counter is incremented by 1. In contrast with Hall's concept, an email address is bound to a specific user. When Alice gets an email from a user claiming to be Bob and to an address with extension e', then Alice checks whether $e' = e$(`Alice@wonderland.com,Bob@jungle.com;` n_{Bob}). If $e' \neq e$, the address is non-genuine and Alice has different options on how to proceed. One option is to accept this email if the sender belongs to a set of users who may be allowed to use this address, maybe because they are friends of Bob. Another option would be to reject the email and ask the sender to apply for an extended email address. To get such an address, the inquirer is involved in a payment-based procedure which might be CPU-based (Dwork and Naor, 2002), for example. While a single user can perform this challenge–response procedure easily, a spammer would be forced to do millions of handshakes. This approach faces the problem of having to hide email addresses, too. Furthermore, extended email addresses built this way are far removed from being guessable. To create an email address (circumventing any resource-consuming challenge-response procedure) which can be used by Bob to send emails to Alice, an adversary or spammer respectively needs to know the function e, Alice's and Bob's core addresses, and Alice's counter n_{Bob}). As a matter of cryptographic principle, the keeping of secrets should not rely on the algorithm used, so that e would be known or easily guessable. Alice's and Bob's core addresses are public data. In most cases, the counter, although not being public and only stored on Alice's side, would be easily guessable, as Alice is not believed to have very often created a new extended email address to be used by Bob. Thus, the counter should be a value of between 0 and, let's say, 1000.

The concept of Ioannidis's Single-Purpose Address (SPA) goes even slightly further. It, too, addresses cases in which it is irrelevant whether an address is simple and readable (e.g `schryen@winfor.rwth-aachen.de`, or completely obscure (e.g. `VP72W24KM7IH7FT4O@winfor.rwth-aachen.de`) and where it is important to be able to limit the use of an address to only those purposes for which it was issued. The concept is both to prevent a party from sending advertising material in the future (which most online vendors do, despite their assurances to the contrary), and to prevent abuse of the supplied address by third parties who, with or without the cooperation of the merchant, acquire our email address. This is achieved by encoding rules as part of the email addresses in such a way that the potential senders cannot alter these rules without, at the same time, invalidating the alias. These address-specific rules are applied when an email has been sent to a particular address. This way, the user does not have to store any per-address rules locally or keep track of multiple email addresses, which rules out the problem of the size of the alias list and the size of filtering rules growing without bound. The SPA consists of two parts: an indication of the addressee, and an appropriately encoded description of the policy that will be applied when the message is received. The addressee can simply be identified by his or her username, with the policy part given as the extension, as in a "user + extension" convention. Since, presumably, the "naked" (with no extension) main address of the user would still be valid, it is recommended that users who want to use SPAs acquire a second address, and set up their systems so that mail to the naked second address is rejected. The creation of the second part of the SPA proceeds as follows (the parenthesized details refer to the prototypic implementation of Ioannidis):

1. A rule, as part of a user overall email policy, is encoded. For example, a rule could be "*accept this mail between January 30, 2003 and March 20, 2003, and only if the user is sending it from some machine in cs.miskatonic.edu; if accepted, forward the mail to* `seldon@trantor.gov`" (Ioannidis, 2003, p. 3). The encoding results in a bit-oriented representation of the rule (112 bit representation), its hash (MD5, 16 bit) or even MAC value is generated and added resulting in a structure called "SPA block" (128 bit representation). Only in the case of a MAC being generated, using a user-specific (symmetric) key, will the SPA block be user-specific.
2. The SPA block is encrypted under a symmetric key (256 bit AES-key in CBC mode) known only to the user creating the SPA.
3. The output of the encryption is a string of randomlooking bits and, as such, it is not suitable for use as an email address. It must, therefore, be encoded (Base32 encoding) by using a set of characters that are legal for email addresses. The resulting string forms the second part of the SPA and is called SPABEE (SPA block encoded and encrypted).

Fig. 5 summarizes the process of generating an SPABEE. The address of an SPA email can either be checked by the receiving MTA or by the MDA. The processing has to be done in reverse order, as described. Thus, the processing node needs to have both the symmetric user-specific key for decoding and the email address that the email was sent to. This address is given in the RCPT command that an MTA has access to, but if an MUA is intended to process the SPA, any of the MTAs involved in email delivery must put this information in the

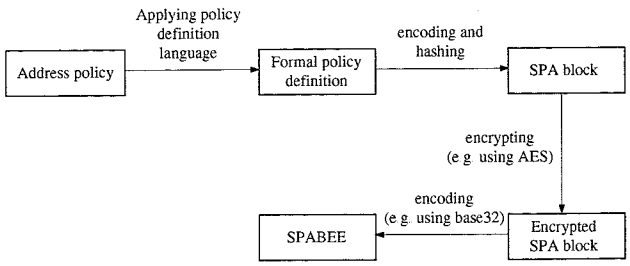

Fig. 5 – SPABEE generation process.

header, e.g. in a "Received:" header line. The "user" part of the SPA-address is used for identifying the recipient and the corresponding key for decrypting. The decoded and decrypted SPABEE gives a binary representation which is checked up on being a valid SPAB. This is the case if, and only if, the hash value or MAC corresponds to the binary representation of the rule encoded. If not, the address is not a valid SPA-address and the email will be discarded. Otherwise, the SPAB is decoded, then the email is checked against this policy and, subsequently, the MTA or MUA either delivers it, bounces it, or discards it accordingly. Compromising the system is possible if an attacker gets the symmetric key either from an unprotected key store or thanks to a successful cryptanalysis. A further attack is to create an SPABEE which represents, after decryption, a valid SPAB. However, this means the generation of a bit sequence that represents, following its decryption, a valid SPAB, i.e. a valid encoded rule and a compliant hash value or MAC. Thus, the deployed algorithms and key lengths have to be chosen appropriately, making these attacks negligible. Like the other obscuring approaches, the protection of users' local address books may remain an unsolved problem. Furthermore, a legitimate first-contact communication via email is complicated because the sender has no means of easily procuring an SPA. Consequently, this approach suffers from the same limitations and drawbacks as does Hall's approach.

Tagged Message Delivery Agent (TMDA) (http://tmda.net) also uses the concept of using the email address to create SPAs. Aside from formatting and implementation details, the main architectural difference is that policy is not explicitly described in the email address, but rather that the address is used to look up the policy in local tables. This means that, for each special address created, policy must be kept so that it can be processed in the future, causing such policy tables to grow without bound when addresses without expiration dates are used.

The presented "security by design" approaches make email addresses difficult to use for humans and, dependent on their particular approach, make email communication more complex (violation of requirements R1 and R2, respectively). The extent to which addresses are protected against harvesting in other email address pools, such as local email directories (R3), depends on a particular approach and its implementation. For example, a "Hall" address, which is stored unencryptedly, can be read and used by any sender. Likewise, a user-specific counter of a "Gabber" address must not be stored unencryptedly. As long as all addresses are textual addresses that need not to be processed by an application, such as a script interpreter, the automatic processing of such obscured addresses is possible (R4). The "security by design" principle, which is inherent in the presented approaches, relies on (mathematical) complexity. Thus, we can use (complexity) theory in order to assess the level of security thereby circumventing an "arms's race" with harvesters (R5). These approaches also provide means for sender-specific email addresses, so R6 is met.

Table 5 provides a glance of the matching between requirements and proposed approaches.

Looking at AOTs, we face the known trade-off between security and ease of use (pragmatics). However, we believe "security by design" approaches to be more promising and long-term effective than "security by obscurity" ones are. The effectiveness of any implemented AOTs should be tested continuously and empirically, for example by the usage of our proposed framework.

7. Conclusion

The empirical results of the study presented in this paper confirm the findings of previous studies, i.e. that particularly web pages and Usenet groups belong to the most vulnerable Internet spots regarding email address harvesting. Although the detected extents to which addresses were misused for spamming differ from each other, Internet locations are still a very attractive and heavily exploited source for address harvesters. In order to hamper spammers in easily procuring email addresses from the Internet at low expense, it seems worthwhile to further elaborate on address obscuring techniques, which fulfill the proposed requirements, in order for these to be effective in the long run. Their deployment is intended to be used complementarily to other anti-spam techniques and should be continuously accompanied by honeypot-based

Table 5 – SPABEE generation process

Requirement/Approach	AOTs following "security by obscurity"	AOTs following "security by design"
Usability of addresses (R1)	+	–
Simplicity of communication (R2)	+	depends on particular approach
Holistic coverage (R3)	–	depends on particular approach and its implementation
Automatic processing (R4)	depends on particular approach	+
Assessment of security level (R5)	–	+
Robustness against harvesting (R6)	–	+

studies which allow the measuring of their practical effectiveness. The honeypot conceptualization and the methodology presented in this paper can serve as a basis for these studies.

Appendix.

Table 6 – Usenet groups

de.comp	de.test
de.admin	alt.[a]
de.alt	comp.[a]
de.comm	free.[a]
de.etc	novell.[a]
de.markt	microsoft.[a]
de.rec	rec.[a]
de.sci	sci.[a]
de.soc	soc.[a]
de.talk	uk.[a]
de.org	

[a] Some Usenet groups of the particular category were selected (arbitrarily).

Table 7 – Locations seeded with addresses which attracted the most spam

#spams	Web location
829	http://jeepbrokers.com/jeepbrokers_guestbook.htm
536	http://www.theaterhaus.com/easync/easync_page.php?id=1,4,1&page=forum_geastebuch.htm
534	http://www theaterhaus.com/easync/easync_page.php?id=1,4,1&page=forum_geastebuch.htm
518	http://www.la-palma24.net/de_visitas/guestbook.php3
510	http://www.la-palma24.net/de_visitas/guestbook.php3
499	http://www.la-palma24.net/de_visitas/guestbook.php3
453	http://www.beaufortrlty.com/guestbook.html
412	http://www.cyber-kitchen.com/cgibin/gbook/guestbook.cgi
396	http://www.germantownnews.com/guestbook
393	http://www.cyber-kitchen.com/cgibin/gbook/guestbook.cgi
383	http://www.kelso.gov/
378	http://www.cyber-kitchen.com/cgibin/gbook/guestbook.cgi
337	http://www.bowlsengland.com/efgbk00.htm
337	http://jeepbrokers.com/jeepbrokers_guestbook.htm
332	http://jeepbrokers.com/jeepbrokers_guestbook.htm
304	http://jeepbrokers.com/jeepbrokers_guestbook.htm
275	http://www.metager.de
253	http://www.ourchurch.com/view/?pageID=111918
237	http://www.radiojamaica.com/guest-book/
222	http://books.dreambook.com/dawsadopt/main.html
	newsgroup
69	de.rec.sport.paintball
67	de.rec.tv.buffy
60	alt.drugs
52	de.sci.misc
50	alt.america
41	alt.airports
35	alt.fan.brad-pitt
30	de.rec.sport.misc
29	de.talk.jokes
29	de.org.ccc
28	de.soc.weltanschauung.misc
27	de.rec.tv.technik
26	alt.fan.shania-twain
26	alt.games.microsoft.age-of-empires
26	microsoft.public.microsoft.transaction.server.integration
25	de.etc.selbsthilfe.angst
23	de.talk.jokes.d
23	alt.windows-me
22	alt.off-topic
	newsletter
66	Churchill College, University of Cambridge; http://www.opendays.com/newsletter/
45	Jayde B2B Search Engine; http://www.jayde.de
44	Jayde B2B Search Engine; http://www.jayde.de
40	Jayde B2B Search Engine; http://www.jayde.de
22	Jayde B2B Search Engine; http://www.jayde.de
21	Churchill College, University of Cambridge; http://www.opendays.com/newsletter/
21	Weisser Ring, http://www.weisser-ring.de/bundesgeschaeftsstelle/newsletter/index.php
16	Central Florida Photography Club; http://www.cflphotoclub.com/home/newsletter_form.htm
6	French Erotic Site; http://www.sexy.legratuit.com
6	Lowell Jaks Welcome Page; http://www.lowelljaks.com/
6	Churchill College, University of Cambridge; http://www.opendays.com/newsletter/
6	French Erotic Site; http://www.sexy.legratuit.com
6	Churchill College, University of Cambridge; http://www.opendays.com/newsletter/
5	French Erotic Site; http://www.sexy.legratuit.com
3	Edvisors.com: International Student Newsletter; http://www.edvisors.com/cgi/page.cgi?p=newsletter
3	Edvisors.com: International Student Newsletter; http://www.edvisors.com/cgi/page.cgi?p=newsletter
3	Lowell Jaks Welcome Page; http://www.lowelljaks.com/
2	Casinomeister; http://www.casinomeister.com/newsletter.html
2	Casino Bielefeld; http://www.casino-bielefeld.de/newsletter.php
2	Edvisors.com: International Student Newsletter; http://www.edvisors.com/cgi/page.cgi?p=newsletter

Acknowledgements

The set up of the honeypot was strongly supported by Reimar Hoven, manual work regarding the classification of incoming e-mails had to be done, Stephan Hoppe sacrified much time in performing this task. Many thanks also go to Jan Herstell and to Christine Stibbe for proofreading.

REFERENCES

von Ahn L, Blum M., Hopper N, Langford L. CAPTCHA: using hard AI problems for security In: Proceedings of Eurocrypt; 2003. p. 294–311.

von Ahn L, Blum M, Langford L. Telling humans and computers apart automatically. Comm. ACM 2004;47(2):57–60.
Bundestag der Bundesrepublik Deutschland. Gesetz gegen den unlauteren Wettbewerb (UWG), Bundesgesetzblatt Jahrgang 2004 Teil I Nr. 32; 07/03/2004.
CAUCE: Coalition against unsolicited commercial email, <http://www.cauce.org/>.
CAUBE.AU. The CAUBE.AU spam survey, <http://www.caube.org.au/survey.htm>; 1999.
Center for democracy & technology, Why Am I Getting All This Spam? Unsolicited commercial e-mail research six month report, <http://www.cdt.org/speech/spam/030319spamreport.shtml>; 2003.
Dwork C, Naor M. Pricing via processing or combatting junk mail. In: Boneh D, editor. Proceedings of the 22nd annual international cryptology conference (CRYPTO 2002), number 740 in LNCS. Springer; 2002. p. 137–47.
Daniel Faigin, Matthew Bishop, Tasneem Brutch. PANEL – Miracle cures and toner cartridges: finding solutions to the spam problem. In: 19th annual computer security applications conference, Las Vegas; 2003.
FTC, Email address harvesting: how spammers reap what you sow. FTC Consumer Alert, <http://www.ftc.gov/bcp/conline/pubs/alerts/spamalrt.htm>; 2002.
FTC, Email address harvesting and the effectiveness of anti-spam filters, report, <www.ftc.gov/opa/2005/11/spamharvest.pdf>; 2005.
Gburzynski Pawel, Maitan Jacek. Fighting the spam wars, a remailer approach with restrictive aliasing. ACM Trans. Inter. Tech. 2004;4(1):1–30.
Gabber E, Jakobsson M, Matias Y and Mayer AJ. Curbing junk e-mail via secure classification, In: Proceedings of the second international conference on financial cryptography; 1998. p. 198–21.
Hall R. Channels: avoiding unwanted electronic mail. In: Proceedings of the DIMACS symposium on network threats; 1996.
Ioannidis, J. Fighting spam by encapsulating policy in email Addresses, In: Network and distributed system security symposium (NDSS '03); 2003.
Kerckhoffs Auguste. La cryptographie militaire. Journal des sciences militaires:5–83. pp. 161–191, Feb. 1883, http://www.petitcolas.net/fabien/kerckhoffs/, Jan. 1883;IX.
MAPS. Definition of spam, <http://www.mail-abuse.com/spam_def.html>.
MessageLabs. MessageLabs intelligence report 2005, <http://www.messagelabs.com/publishedcontent/publish/threat_watch_dotcom_de/threat_statistics/spam_intercepts/DA_136867.chp.html>; 2005.
OECD. Issues on the measurement of unsolicited electronic messages; 2003.
OECD. In: Background paper for the OECD workshop on spam; 2003.
One hundred eighth congress of the United States of America. Controlling the assault of non-solicited pornography and marketing act of 2003; 2003.
Opt-in vs. Opt-out legislation, <http://notebook.ifas.ufl.edu/spam/Legislation.htm>.
Prince Matthew B, Holloway Lee, Langheinrich Eric, Dahl Benjamin M, Keller Arthur M. Understanding how spammers steal your e-mail address: an analysis of the first six months of data from project honey pot. In: second conference on email and anti-spam CEAS 2005, <www.ceas.cc/papers-2005/163.pdf>.
Raz U. How do spammers harvest email addresses, <http://www.private.org.il/harvest.html>.
Spamhaus. The definition of spam, <http://www.spamhaus.org/definition.html>.
Symantec. Spam statistics, <http://www.symantec.com/region/de/PressCenter/spam.html>.
Schryen G. Effektivität von Loesungsansaetzen zur Bekaempfung von Spam. Wirtschaftsinformatik 2004;46(4):281–8 [English version is not published but available from the author].
Schryen G. An email honeypot addressing spammers' behavior in collecting and applying addresses. In: Proceedings of the sixth IEEE information assurance workshop, Westpoint; 2005. p. 37–41.
The European parliament and the council. Directive 2002/58/EC of the European parliament and of the council concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications); 07/12/02.

Guido Schryen graduated from the RWTH Aachen University (Germany), where he earned a Master in Computer Science and a Master in Operations Research. He got his PhD from the Faculty of Business Administration and Economics of RWTH Aachen University where he now holds a postdoc position. His current research activities focus on Internet security and anti-spam measures.