Efficient Directory Harvest Attacks

Boldizsár Bencsáth István Vajda
Laboratory of Cryptography and Systems Security (CrySyS)
Department of Telecommunications
Budapest University of Technology and Economics, Hungary
{*bencsath, vajda*}*@crysys.hu*

ABSTRACT

In this paper the E-mail Directory Harvest Attacks (DHA) are investigated. We elaborated a method for optimizing the wordlist size used by the attacker in a resource limited environment. We analyzed the results and proved that our method is optimal. We also present an efficient countermeasure against DHA.

KEYWORDS: E-mail, Directory Harvest Attacks, Dictionary Attacks, Brute-force

1. INTRODUCTION

The e-mail Directory Harvest Attack (DHA) is a special form of brute force attack. The attacker's goal is to gain information about the e-mail addresses used in a domain name. The attack itself is not harmful to the attacked domain (except when the volume of the attack is high enough to be considered as a Denial of Service attack). The secondary effect of a successful DHA is that the e-mail address gets inserted into a bulk e-mail address list and spam starts flooding the identified user.

The DHA problem is not elaborated well publicly, although most commercial spam solutions state that they deal with the threat. In this paper we analyze some interesting parts of the problem area and meanwhile we also present an efficient countermeasure. The first part of the paper analyzes the attacker's possibilities and the economy of the attack. Our observations show that the attackers generally use a dictionary (wordlist) of a fixed size to attack different domains during a DHA. In this paper we will show that the optimal wordlist should be different to every domain according to the number of users expected in that domain. We also present proof that our algorithm is optimal. For countermeasure we recommend a centralized protection method, where a DHA RBL server gets information about the hosts sending e-mails to unknown addresses. Beside the method we also present information about our prototype for this protection method.

The structure of the paper is the following:

Related works are presented in Section 2. Section 3 gives the basic description of the DHA optimization problem and the definition of the algorithms, and also highlights the economical reasons to attack In Section 4 we analyze the algorithms. and prove optimality property. Our supporting simulation results are presented in Section 5.

The possible countermeasures against harvest attacks are described in Section 6. Our proposed centralized protection system against DHA is presented in Section 7. Finally, Section 8 summarizes the paper.

2. RELATED WORK

On-line brute force attacks where the goal is to identify a valid username/password pair have been known for a long time. The oldest attacks were aimed at telnet and ftp accounts. Lately, attacks are deployed against SSH, POP3, RAS, Samba, and various HTTP based services.

These attacks show similarity to DHA attacks, as the attacker's goal is to identify valid data and filter out the rest. Active countermeasure is possible for both problems if the attack is on-line: the invalid access / harvest trials can be detected and therefore the attackers can be rejected.

The typical countermeasure for brute force authorization trials is locking of the affected accounts, but this can result in a Denial of Service (DoS) attack against the users of a particular system. (see [7] for some details) The solution therefore can be extended e.g. by captcha-based unlocking mechanism: The user cannot log into the locked account, but the account can be unlocked automatically if the user proves that he is human. Captcha is an automated Turing test, a question that a human can answer, but an automatic program cannot. (e.g. recognition of characters in a picture) ([1])

Other possible protection is to insert some delay into the authentication process after a number of unsuccessful trials. Although this can deny the attack from a single host it cannot solve the problem with a distributed attack. If the whole system slows down then a DoS attack is possible. ([2] analyzes traffic analysis based protection against DDoS)

Protection against DHA is similar to the password brute-

force attacks, but an attempt cannot be bound to a particular user. One possible protection against a DHA is to falsify or deny information to the sender if an e-mail is sent to an unknown user (SMTP error 550 according to [3]). We think that this 'hack' is harmful as this information is very important to the users of the internet. Many other solutions protect against DHA by filtering out the hosts sending a larger volume of emails to unknown addresses. As for the brute force password attacks, this method does not protect the system from a highly distributive attack.

Many commercial solutions also provide countermeasures for harvest attacks. Kerio MailServer ([4]) detects emails to unknown users and after a threshold the server begins to filter out possible attackers.

Postini provides managed e-mail services with DHA protection. Their white paper ([5]) gives details about the protection method and about the DHA problem as well. Postini website also has important statistical data about current DHA activity they detected.

Secluda Inboxmaster offers customizable SMTP error message: If a spam is detected during the mail delivery (by other parts of the system), a bounce message is sent back to the sender so the sender might think that the address is not valid. The problem with this method that the e-mail message during the DHA might not be distinguished from legitime e-mails, as its goal is only to identify valid addresses, while the false positives can be misguiding to users.

Latest antispam projects, like ProjectHoneypot ([6]) have not yet integrated protection against DHA. The ProjectHoneypot system tries to identify spammers by trap e-mail addresses. This can be extended by the identification of mass e-mail senders with many messages to unknown recipients as we propose later in this paper.

3. OPTIMIZING THE DHA

The Directory Harvest Attack can be categorized into two main categories:

- The attacker tries all possible valid character combinations with 1...N characters. This can be enhanced that only wordlike strings are used.

- The attacker uses a wordlist (or dictionary) of possible (frequent) user names (e-mail address local-parts). The wordlist is typically based on dictionary words or generated using previously known e-mail address lists.

The typical attacker cannot achieve a successful attack with ten-millions of e-mail trials to a single host, therefore we only analyze the problem of the more efficient wordlist based attacks.

3.1 Description of common attack method

The actions of a DHA attacker can be described by the following steps:

1. The attacker selects a number of destination domains. The selection can be based on public information available about the domains. (E.g. number of expected users on the system)

2. The attacker gains control over innocent victim computers and turns them into zombies. This can be done using a trojan program or e-mail worm, etc.

3. The attacker carries out the attack by controlling the zombies.

4. The attacker gathers the results from the zombies and analyzes it.

A network attack with a similar method is presented in the case study [9].

Investigated systems we have access to we experienced most harvest attacks aim hosts with immediate SMTP error reporting if a mail is coming to an unknown address. Some of our hosts simply accept all incoming e-mails and pass them to another (protected) host (e.g. a firewall). On these systems a DHA attack is still possible, but the attacker should use and maintain a valid return address to get notified about the unknown users. We found that these hosts are almost never attacked by DHA.

During the last 3 months the attackers used about 100.000 different words altogether. Most of the words show similarity to common email address local parts (like 'cpark', 'jsamson', with typical length of 5-7 characters). Sometimes the words seem to be artificially generated from syllables (eg. 'kizu', 'pugeriu'), when the typical length is 4-6 letters. Most of the words are tried multiple times even on the same domain, but we cannot distinguish if it is tried multiple times by the same attacker, or multiple attackers tried the same word. The attacks were highly distributed, but many times the single trial messages contained 20-30 different recipients.

3.2 The algorithms

Let's introduce a few notations. An attacker controls $A = \{A_1, A_2, ..., A_{N_A}\}$ zombie computers, where N_A denotes the number of the zombies.

The target of the attack is the following set of domains: $D = \{D_1, D_2, ..., D_{N_D}\}$.

Within domain j we have users $U^j = \{U_1^j, U_2^j, ..., U_{N_U^j}^j\}$.

The dictionary (wordlist) of the known email address local parts (i.e. e-mail user names) is denoted by the sequence $W = W_1, W_2, ..., W_{N_W}$ that is the size of the list is N_W.

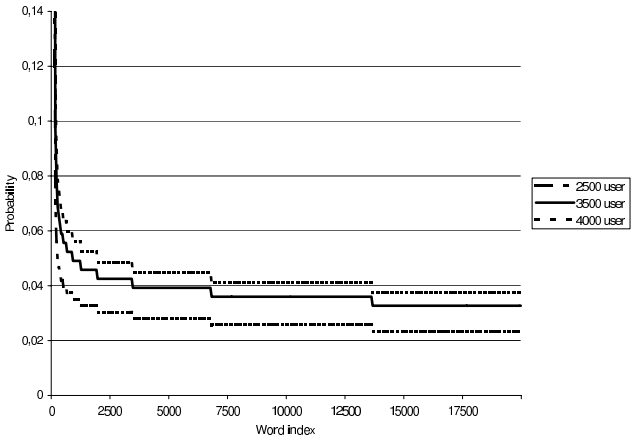

Figure 1: Probability of success for the wordlist elements in systems with different user numbers

For a geographic (cultural) region we can assume that the distribution of the local parts is very similar for every domain.

We assume that the probability distribution of the words $\{P(W_i)\}$ is known by the attacker. The wordlist is sorted according to the descending order of probabilities. ($P(W_i) \geq P(W_{i+1})$ for every i.)

Using 10 million e-mail addresses gathered from the internet we analyzed the distribution of local parts. Figure 1 show the success probability for a DHA trial on systems with various user counts.

The number of attack trials is limited by integer t and the number of attacked domains is N_D.

Now we describe the algorithms: Algorithm 1 is the experienced behavior of the DHA attackers. Algorithm 2 is our proposed algorithm.

Algorithm 1: The attacker tries the first $T = t/N_D$ items of the wordlist for each domain, which are the words with the highest probability.

Algorithm 2: For every trial the attacker scans all the target domains and selects the domain where the probability of success is the greatest. The trial is carried out against the selected domain.

In pseudocode our proposed Algorithm 2 can be described as follows

for all $d \leq N_D$ **do**
  $i[d] \Leftarrow 1$
end for
$trial \Leftarrow 1$
while $trial \leq t$ **do**
  $d \Leftarrow 1, p_{max} \Leftarrow 0, target \Leftarrow 0$
  while $d \leq N_D$ **do**
    if $p(W_{i[d]} \in U^d) > p_{max}$ **then**
      $p_{max} \Leftarrow p(W_{i[d]}), target \Leftarrow d$
    end if
  end while
  Try(word $W_{i[target]}$, on domain $target$)
  $i[target] \Leftarrow i[target] + 1$
  $trial \Leftarrow trial + 1$
end while

The wordlist elements are tried one after the other, $i[d]$ denotes the index of the next word in the wordlist for a given d domain.

3.3 Reasons to attack: The economy of the DHA

The attacker might earn profit from selling e-mail addresses for spamming purposes.

The attacker tries to maximize the profit from the attack. The net profit of the attacker is calculated by the following formula $V = -(C_0 - f_c(t)) + I$ where $I = \sum_{j=1}^{t} p_s(w_j) * \mu$.

Here C_0 os the initial cost of the attack, $f_c(t)$ if the cost depending on the number of trials, furthermore I is the income decomposed into sum, where $p_s(w_j)$ is the success probability for the word tried in the step j, while μ is value for an identified address.

We do not know the exact cost function, therefore in this section our goal is to optimize (maximize) the value of the income (I). This corresponds to the maximization of the successfully identified e-mail addresses.

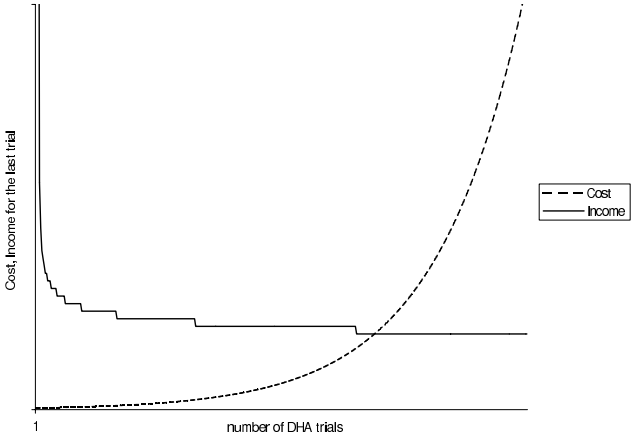

Figure 2: Cost and Income functions of the attacker

Figure 2 illustrates the cost and income curves: The intersection of them corresponds to the maximal profit of the attackers.

4. ANALYSIS OF THE ALGORITHMS

Now we give formulae for the expected number of successfully found e-mail addresses.

Let random variable S_i denote the number of email addresses found by the attacker when he attacks domain D_i, $i = 1, ..., N_D$. Random variable S denotes the total number of successes, i.e. $S = \sum_{i=1}^{N_D} S_i$. Below we give an analysis for the expected value $E(S)$ for the Algorithm 1. and Algorithm 2. First consider Algorithm 1:

$$E(S_i) = E\left[\sum_{j=1}^{T} \chi_{\{W_j \in U^i\}}\right] = \sum_{j=1}^{T} E\left(\chi_{\{W_j \in U^i\}}\right)$$

$$= \sum_{j=1}^{T} N_U^i P(W_j) = N_U^i \sum_{j=1}^{T} P(W_j)$$

where $T = t/N_D$ and χ_A is the indicator function for set A. Hence we get

$$E(S) = \sum_{i=1}^{N_D} N_U^i \sum_{j=1}^{T} P(W_j)$$

In case of Algorithm 2. let t_i denote the number of trials against domain D_i, $i = 1, ..., N_D$. Note, these numbers are deterministically determined by the probability distribution $P(W_j) = h(j)$,$j = 1, 2...$, and by the sizes N_U^i, $i = 1, ..., N_D$, as it will be detailed below. Assuming the knowledge of t_i, $i = 1, ..., N_D$, for the expected number email addresses found by the attacker applying Algorithm 2 we get:

$$E(S) = \sum_{i=1}^{N_D} N_U^i \sum_{j=1}^{t_i} P(W_j)$$

For the calculation of the number of trials, we use the Lagrange's multiplicator technique. We have to maximize function

$$H(\underline{t}) = n_1 h(t_1) + ... + n_k h(t_k)$$

under conditions $g(t) = t_1 + ... + t_k - t = 0, t_1 \geq 0, ..., t_k \geq 0$ where $\underline{t} = (t_1, ..., t_k)$, $k = N_D$, $n_i = N_U^i$.
We start from function

$$G(\underline{t}) = H(\underline{t}) + \lambda g(\underline{t})$$

and we solve the following system of $k + 1$ equations:

$$\frac{\partial G}{\partial t_i} = n_i h'(t_i) + \lambda = 0, \ i = 1, ..., k$$

$$\frac{\partial G}{\partial \lambda} = t_1 + ... + t_k - t = 0$$

where h' denotes the derivative of function h. Eliminating λ we arrive to the following system of k equation in unknowns t_i, ($i = 1, ..., N_D$):

$$n_1 h'(t_1) - n_{i+1} h'(t_{i+1}) = 0, \ i = 1, ..., k - 1$$

$$t_1 + ... + t_k - t = 0$$

For instance, we show how to solve this system, when function h has the following form $h(x) = a/x^b$, $h'(x) = c/x^d$, $c = -ab$, $d = b + 1$. After some straightforward algebra we arrive to the following system of linear equations:

$$m_1 t_{i+1} - m_{i+1} t_1 = 0, \ i = 1, ..., k - 1$$

$$t_1 + ... + t_k - t = 0$$

where $m_i = n_i^{1/d}$. Whence we get the following result

$$t_i = t \cdot n_i^{1/d} / \sum_{i=1}^{N_D} n_i^{1/d}, \ i = 1, ..., N_D$$

It is easy to check that $t_1 \geq t_2 \geq ... \geq t_{N_D}$ and $t_1 + t_2 + ... + t_{N_D} = t$.
Algorithm 2 is optimal in the sense that the expected number of successfully identified e-mail addresses by the attacker is maximal.
Sketch of proof: Assume that there exists a certain set of trials (X) that is better than the set (Y) produced by Algorithm 2. (both sets have the same size t) Therefore there should exist an element (word, domain pair) $x \in X$ for which $x \notin Y$. Because Algorithm 2 selects elements with highest possible success probabilities, therefore element x must have success probability less than or equal to the probability for any element from Y. Let's substitute the element x in the set X by an element y, $y \notin X$ and $y \in Y$. Let the modified set be denoted with X^*. The sum of success probabilities for X^* compared to the same of X will be higher or it remains the same. As we have shown above this sum of probabilities is actually the expected number of successfully identified e-mail addresses. In case when the summed probabilities over X^* is greater than that of over X we arrive to a contradiction. (X is not better than Y) In the other case when the two sums of probabilities are equal, we repeat the step of substitution. Exhausting all possible substitutions we arrive to a contradiction or we'll that the result of Algorithm 2 is not inferior to the optimal set X.

5. SIMULATION

To support our results we carried out simulations. Two possible scenarios with different number of target domains and number of users were investigated. The main results are summarized in Table 1.
The heading "Total user" denotes the total number of users in D, furthermore "succ. Alg i" denotes the average of successfully identified addresses by Alg. i. The simulation clearly shows the superiority of Algorithm 2.

Table 1: Simulation results

N_D	Total user	succ. Alg. 1.	succ. Alg 2.
11	1730	87	180
6	23000	5395	6754

6. COUNTERMEASURE AGAINST DHA

We can separate the possible countermeasures in two categories:

- Host based protection: An autonomous system has its own protection method, without relying on other parties.
- Network based protection: The system is cooperating with other parties to protect itself from the DHA. This method can be centralized: a server coordinates the protection.

6.1 Host based protection

A DHA attacker uses information gathered about unknown users to identify valid users. Some protection methods try to achieve protection against DHA by falsifying or denying information about unknown users. As we mentioned before, these protection methods rise new problems for the legitimate human users, therefore any protection method based on this behavior should be avoided.
Filtering based on error reports: When a computer is sending e-mail for an unknown user, we insert the address of the computer to a list and filter out all requests coming from computers on the list. To deal with the problem of false positives we only insert computers into the list if the number of e-mails to unknown recipients exceeds a threshold. The problem with such host-based filtering algorithm is that it is not resistant against distributive attacks. Experience shows that DHA attacks are highly distributive, the maximum of trials coming from an IP address can be as low as 1-2. Although the filtering algorithm will filter out thousands of computers, the attacker will continue the attack from thousands of other, not filtered zombies.
The simple host based filtering can be extended:

- We can examine the address field of attacking e-mails. Some attackers sort the wordlist in alphabetical order. Similar suspicious properties can serve a starting point of a protection method.
- The attackers generally use the most frequent e-mail addresses during the attack. This list can be reconstructed by observing the trials of the attackers. An enhancement to the basic protection method can filter out all the IP addresses trying to send emails to unknown users where the particular username tried exists in this wordlist.

Although these enhancements could be successful for a short time period, the attackers can also adapt to the protection and the trials won't be statistically distinguishable from legitimate e-mails. Even if this protection would be successful to protect a single host, this approach won't help others and the internet to efficiently fight against DHA.
The filtering approach can be real-time or log-based. If the protection is based on log analysis and is not real-time then then the attacker has wide time windows to carry out attack enabling him to test lots of addresses from a single attacking host.
The behavior of the server to the offending computers (computers on the list) can also be different: Some protection methods deny any IP traffic coming from those computers, while other servers only reject receiving e-mails from the attackers. It is also possible that the server only reports a temporary error (like in greylisting [10]) to the attacker thus the attacker won't know if he is filtered or the delivery should be retried later.

6.2 Network based protection method

Our proposed solution is also based on filtering but with a centralized approach. Parties of the protection are the following: the attacking computer, the attacked server host and the centralized DHA RBL (Real-time blacklist) server. (check [8] for general some information about anti-spam techniques like RBL)
If an attacker sends an email to an unknown address on the attacked server, the attacked server will send an error report to the central DHA RBL server. The error report contains the offending IP address, the recipient address tried, and the time of the attack. The centralized server collects the reports from the protected servers. If the number of trials exceeds a limit, the server inserts the address of the attacker in the blacklist. The server also stores the time when the last e-mail observed from the given attacker with unknown recipient.
As usual, the RBL list can be queried by sending an address as a request to the server questioning if an address exists in the list. The server does not publish addresses on the list, instead it simply answers with yes or no.

6.3 Aging

For every blacklist-based method it is very important to decide how the addresses are cleared from the list. After removing the address from the blacklist the attacker can continue the attacks, but if an address is remained too long on

the list, then legitimate traffic might be blocked for a long time.
Several methods for clearing the blacklist:

Administrator-driven aging: The administrator of the RBL server manually decides about the removal of the address. The owner of the attacking zombie computer can also ask the RBL administrator to remove the computer from the list. (e.g. the backdoor used to carry out the attack is removed from the computer)

Simple aging: After a given amount of time elapsed from the insertion of the address into the RBL the address is removed automatically. The problem with simple aging is that an attacker can easily estimate when the address is cleared from the RBL, therefore he can immediately restart attacking hosts.

Multi-phase aging: After the address of the attacker inserted into the list we can expect that no more error report will arrive to the server according to the given attacker. (Every server instantly filters out the traffic coming from the attacker) In multi-phase we define some protected hosts (automatically, manually, or randomly), which computers won't filter out traffic from the attacker, but report any attacking traffic. If the attacker renews or continues the attacks, the server may have fresh information about the attacker and so it can remain on the list for a longer time.

7. DHA PROTECTION PROTOTYPE

Our prototype system uses the DNS protocol for transferring queries and reports to the server and also to transfer the answer from the RBL server. The DNS protocol is widely used in RBL environment as it is very robust and the inherited caching mechanism of the DNS servers can help lowering the workload of the server.
The procedure of incident reporting is presented in Figure 3.

step 1 The attacker sends an e-mail to an internet mail server (MTA).

step 2 The server answers with valid information: the user is unknown in the system.

step 3 The MTA sends an incident report to the server. This is done by a DNS query with a special format. The queried DNS name contains the information about the offending host.

step 3b The DNS server of the MTA forwards the query to another DNS server or directly to the DHA RBL server. The RBL server decodes the query and processes it.

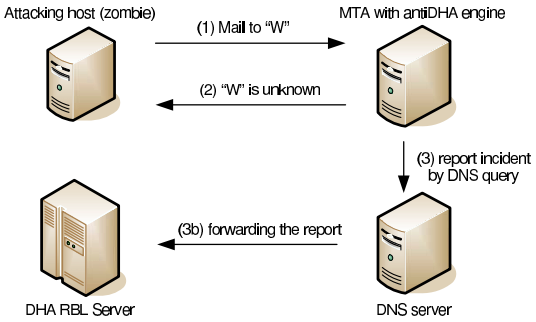

Figure 3: Reporting an incident to the antiDHA system

The filtering mechanism in our prototype system is very simple: After a low number of e-mails sent to unknown addresses (currently set to 10) we insert the offending address into the RBL list.
Figure 4 shows the procedure when a enlisted attacker tries to send a new mail to an arbitrary protected host.

step 1 The attacker tries to send an e-mail to an internet mail server (MTA).

step 2 The protected hosts sends a DNS query with the address of the client (the attacker) embedded in the query.

step 3 The DNS server of the MTA forwards the query to the server.

step 4 The RBL server answers the query with a special form of IP address, meaning "Yes, the computer is in the RBL list". The DNS server can cache the answer for a given address, the caching time (TTL- time to live) can be controlled by the RBL server.

step 5 The DNS server sends back the answer to the protected host.

step 6 The protected host denies the connection with the attacker. This can be done at TCP level, or the attacker can be denied with a proper SMTP error code and explanation.

The results of centralized filtering that the attacker can carry out only a very limited number trials against the protected domains. Of course, our method does not limit the attacks carried out against unprotected domains. The effect of the protection therefore modifies the expected income of the attacker.

8. SUMMARY

In this paper we analyzed the e-mail Directory Harvest Attacks. A protection can be the most efficient we are aware the best possible methods for attacks. An optimal attacking

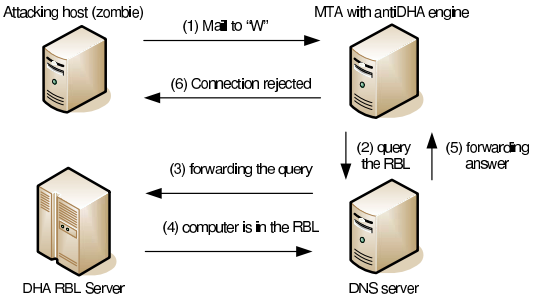

Figure 4: Filtering attacker using the data from the DHA RBL

algorithm based on wordlist statistics has been presented. We derived formulae for the expected number of successfully attacked e-mail addresses for the presented algorithms. Our simulation results based on real data supported the feasibility and efficiency of our proposed algorithm. We investigated in detail the possible countermeasures. Our proposed centralized, blacklist based DHA protection systems is implemented on the level of a prototype.

References

[1] L. von Ahn, M. Blum, N. Hopper, and J. Langford. CAPTCHA: Using hard AI problems for security. In Proceedings of Eurocrypt, 2003.

[2] B. Bencsáth, I. Vajda, Protection Against DDoS Attacks Based On Traffic Level Measurements, 2004 International Symposium on Collaborative Technologies and Systems, pp. 22-28., The Society for Modeling and Simulation International, San Diego, CA, USA, 2004.

[3] J. Klensin, Editor. Simple Mail Transfer Protocol, RFC 2821, April 2001.

[4] Kerio MailServer - state-of-the-art secure email server, http://www.kerio.com/kms_home.html. 2004.

[5] Postini Enterprise Spam Filtering. The Silent Killer: How Spammers are Stealing Your Email Directory, http://www.postini.com/whitepapers/, June 2004.

[6] Project Honeypot - Distributed system for identifying spammers, http://www.projecthoneypot.org.

[7] Corsaire Limited, Stephen de Vries. Corsaire White Paper: Application Denial of Service Attacks, April 2004, http://www.corsaire.com/white-papers/040405-application-level-dos-attacks.pdf

[8] S.Hird. Technical Solutions for Controlling Spam. In proceedings of AUUG2002.

[9] Steve Gibson. The strange tale of the denial service attack against grc.com. http://www.grc.com/files/grcdos.pdf.

[10] E. Harris. The Next Step in the Spam Control War: Greylisting, http://greylisting.org/articles/whitepaper.shtml, 2003.