



Budapest University of Technology and Economics

New Approaches to Mitigate Network Denial-of-Service Problems

Boldizsár BENCSÁTH

Summary of PhD dissertation

Supervised by:

Dr. Vajda István

Department of Telecommunications

2009

1 Introduction

The Internet and its protocols have been greatly enhanced in the recent years. During the planning of the Internet the main focus was set on to protect it against outside attacks by not relying on any central element in the Internet. The original plans did not cover the protection against internal attackers.

The Internet can be considered today as a constant battlefield. The Internet is full of continuous attacks, most of which cannot be prevented or properly handled. More and more specialists and researchers propose a complete clean slate redesign of the Internet with considerations of the dangers observed today.

Attacks on the Internet have also been enhanced in recent years. The first viruses and exploits were built to be just proof-of-concept tools or just for the sake of their own ego. Recently, viruses, worms, pieces of spyware and other malicious code have been introduced to enable spamming, phishing or collecting private data. The individual simple attacker was replaced by the commercial-scale attackers who scan thousands of computer. Their goal is to gain control over computers through their attacks and turn them into so-called zombies to help their future goals.

Denial-of-Service (DoS) attacks have been known for a long time. In recent years we have encountered many different forms of the attack. Two of the most basic and widespread versions are the attacks using software bugs to disable (freeze or slow down) the server (e.g. ping of death), and the simple flooding of the network to eat up all the available bandwidth. Meanwhile, protection against denial-of-service attacks did not improve substantially. The main reason behind this is that the best protection against DoS is the appropriate architecture that helps to deal with the problem. The Internet in its present form was not planned to focus on such problems. The DoS problem is therefore a long known problem, but definitely not solved, and to find a solution in the current architecture is difficult. My research is focused on this area, namely, on the possibility of the protection against DoS problems in the current Internet infrastructure.

There are a number of publications about the possibility of DoS attacks. This could mean that these attacks are really wide-spread, but that is not the case. DoS attacks could potentially be a much larger problem than we currently observe. The protection against DoS attacks is very difficult, but it is very easy to initiate a DoS attack. This shows that we have to take this issue very seriously. Although DoS problems are rare, but all this could change soon and we are not prepared for that.

We can also say that most of the current DoS attacks are not detected, and users only discover the results through the collapse of the system or service. Administrators could also give excuses for the problems without considering the possibility of a direct attack. In a modern IT subsystem, the complicated chain of software and hardware elements make it very hard to find the real reason, specifically the DoS attack in the background of the system collapse. Therefore, detection of such attacks can be one of the hardest problems, and sometimes the protection against a detected attack is easier than the detection itself.

The problem of spam is a good example of the problem of detection. The biggest part of today's Internet traffic is unsolicited emails, spam. The current state-of-the-art method against spam

is the usage of spam-filtering software, but they consume a lot of server resources. The larger number of e-mail messages received by the server means larger resource requirements that can lead to a denial-of-service problem. This way, the system could collapse. A DoS situation occurs, although the goal of the individual attacker was not a DoS attack and an individual attacker alone might not cause a DoS. The result of the enormous amount of e-mails is a system collapse, but the analysis of this collapse is a very hard problem because it is not a standard performance planning problem based on service capacities and client needs, but rather a problem based on the marketing processes of some attackers. The usual dependability and performance analysis might be useless in this case, because these tools focus on real service needs and not on completely different problems such as spam.

During my work I focused on problems that do not have sufficient coverage in the literature. Some may think that these problems are solved, others might not recognize the importance of these problems, or they tried to come up with solutions to these problems but did not solve them. These problems include the modelling of the DoS attacks, protection against the attacks, formal description of the attacks, analysis of the protection methods.

2 Research objectives

My preliminary research objective was to give new solutions and methods in the field of DoS that can help to make the Internet safer. My objective included giving models of the attacks, scientifically elaborated countermeasure methods, analysis, and important data for the handling of the problems.

Two different approaches can be distinguished during the analysis of Internet attacks. The academia tries to model the problems, to give formal methods, and to simplify the problems in order to handle them. This approach has limitations, sometimes the proposed method cannot be practically used because of simplifications. The other approach is the practical or engineering approach, where people try to give practical solutions to the problems, without an in-depth analysis, or scientific proof. They just think that the proposed method works, because their proposal is based on practical experience. In many cases their results were not or cannot be analyzed, there is no proof for the efficiency of the solutions, and the generalization of their work is problematic.

My goal was to incorporate the two approaches mentioned above. I wanted to handle practical problems with the approach of an engineer and meanwhile maintain the scientific method, therefore propose a scientific analysis and formalism. This can be very problematic in the field of DoS attack protection, as the attacks cannot be handled successfully using current scientific methods, and simultaneously, practical solutions can only be applied in some of these problems.

During my research I observed that Denial-of-Service takes a distinguished place among open Internet problems. The problem of DoS has close relation to many other current Internet attacks. In the course of my research I tried to find a comprehensive solution for the problem. Since there is no single general solution, I had to carry out in-depth analysis of the problem, as analyzing the protocol DoS-specific behavior and using the extra information of the protocol to enhance protection.

The main goal of my research was therefore to give new solutions to enhance the protection against DoS and to give models and methods to facilitate the scientific elaboration of these problems.

3 Research methodology

In the first phase of my research I tried to gain practical experience regarding Internet security problems. By my scientific and industrial expertise I encountered problems that could be solved using the current approaches and solutions. I started to model these problems and to plan and deploy protection methods against them. I analyzed the problem using both standard and novel methods too. I validated my analytical results with simulation to avoid errors within the formal methods of the research. After obtaining formal results I went back to the practical questions of the specific problem. I built a prototype application to show that the proposed method is workable and helps to solve the problem.

4 Definition of Denial of Service problems

I defined the denial-of-service condition (Denial-of-Service, or DoS) as follows: *If the functionality of the system or network service in the field of informatics is degraded so much that their clients cannot accept that degradation of the service (where the clients can be real persons or other entities such as software components), and the cause of the change is not a physical one, then we state that the system went into a denial-of-service condition.*

Meanwhile, Denial-of-Service attack (DoS) is defined in the following way: *If a system or network service in the field of informatics is harmed by a third party who intentionally tries to degrade the performance of the service by his or her wishes to a denial-of-service condition, then we talk about a Denial-of-Service attack*

The Denial-of-Service (DoS) attack always means that there is an attacker behind the results. Meanwhile, the DoS condition can also be the result of one of the several other circumstances: for example, if the attacker sends unsolicited e-mails to our system, and therefore he causes the mailing system to freeze (and unable to handle future requests), then we can say that a DoS problem occurs, and we cannot really say that a DoS attack happened. The problem of DoS attacks is very much related to the problem of dependability and resilience, but the goals of those areas of research differ considerably: in the area of DoS we investigate a problem of security, therefore the usable methods and solutions can differ significantly from the other solutions and results shown in the field of dependability.

During my research it was not really necessary to distinguish between a real DoS attack and an unintentional DoS situation. My proposed methods aim to handle DoS conditions whatever their real reasons are. Whenever it is really necessary to distinguish between a DoS condition and a DoS attack, I will indicate it appropriately.

5 New results

The result of my research can be divided into the following three claims or theses:

In the first part I tried to deal with the general form of DoS problems, where I analyzed and enhanced the protection of application-level protocols against general DoS problems. My results show that the outcome of this approach is not sufficient to handle every type of the problem and they have certain limitations.

In the second claim I concentrated on the problem of Denial-of-Service attacks against Internet STMP (e-mail) servers. Not counting the very service-specific solutions, there are only a few available methods against this threat. I proposed a method to protect against DoS attacks by observing and analyzing the traffic to a particular system.

No matter how excellent general solutions we manage to come up with, specific solutions against DoS problems utilizing special properties of the protected system or service can be more useful than general protection. In my third group of claims I focused on the prevention of DoS attacks. I investigated a special problem: this section contains the analysis of Directory Harvest Attacks (DHA) (a type of attack to collect e-mail addresses). DHA and infecting hosts might be used to prepare DoS attack. By giving countermeasures against these threats we also enhance protection against other problems, like DoS.

Claim 1 - Protection against DoS problems with client-side puzzle approach combined with game theoretical methods

Supporting publications: [J1], [C2], [C3]

I propose a novel game theoretical approach for the analysis of the client-side puzzle technique against DoS attacks. I determined the optimum of the game depending on the cost parameters. I constructed a novel client-side puzzle challenge, which uses multiplications of prime numbers. The advantage of the proposed algorithms is that its computational complexity can be precisely computed.

I proposed a solution against DoS attacks with the use of a client-side puzzle technique combined with game-theoretical methods. I analyzed and described in detail how current protocols can be enhanced to be protected against DoS attacks with the proposed use of a client-side puzzle technique. I described in detail how the proposed technique enhances the protection against DoS attacks. I also defined methods to describe the behavior of the attacker and the server in a game theoretical way and I also showed how the usage of mixed strategies can enhance the protection against DoS attacks. The protection against DoS attacks can be enhanced with my proposed approach that combines game theoretical analysis and client-side puzzle technique. A further advantage of my work is that with the usage of both methods the analysis of DoS protection is easier and therefore better protection methods can be proposed.

I propose a scalable client-side puzzle challenge that uses the product of prime numbers. It is not easy to propose a client-side puzzle challenge with the necessary properties against DoS attacks. The generation of the challenge should be fast and also the verification process should

be easy. The storage need of the puzzle at the server side should be low. The computational need at the client side should be given. My proposed client-side puzzle in this section carries a number of advantages and good properties, it is well understandable and the implementation can be also easy. The biggest advantage of my proposed puzzle against hash-based solutions is that it uses only basic operations such as multiplication and therefore the resource need of the puzzle is much more appropriate. To support the applicability of my proposed client-side puzzle, I give the necessary parameter computation through the analysis of the computational need of the puzzle.

Introduction and modeling

The cryptographically designed protocols (where we think about such communication protocols that use cryptographic operations) are mostly highly vulnerable to DoS attacks, as they execute complex computations such as a public key encryption. The possibilities and goals of the attackers can be very different. E.g. a DoS attackers can successfully attack a cryptographic protocol, even if it is generally considered as a safe protocol.

The DoS attackers try to enforce the attacked parties to execute a large quantity of unneeded procedures in a certain time, and they also try to fix that these procedures mean a large quantity of computational need to the server. Of course, the goal of the attackers in this case is to minimize their resource consumption. Possibly, both the attacking client and the server uses their resources optimally, they try to use up the least energy to fulfill their goals.

Researchers (e.g. [1], [2]) proposed methods to refine the protocols to mitigate the problem, but the applicability and effect of these solutions is limited. A more general approach is the client-side puzzle technique (e.g. [3],[5]), which helps to deal with multiple problems, such as DoS and spam. Although, only a few researchers analyzed how the client-side puzzle techniques is usable against distributed DoS (DDoS) attacks, and the proper analysis on the practical implementation of the client-side puzzle is also not deeply elaborated. It is a hard task to model and analyze the situation with a real-life human attacker. I tried to use game theoretical approach to analyze the practical possibility of the use of the client-side puzzle approach thus giving protection method against some of the possible types of DoS attacks.

In the case of client-side puzzle we extend the communication protocol with a new header part. This protocol header is a challenge-response solution that provides protection for the following parts of the original protocol.

I propose to increase the efficiency of the client-side puzzle approach with the use of game theoretical methods. In my proposed method the client-side puzzle protocol header is defined between the server (S) and the client (C). In the first step, the server gets a service request and answers with a puzzle (challenge) to the client. The client solves the puzzle and sends back the results. The server verifies the received result and if is correct, it executes the original (protected) protocol steps.

The attacker can choose among three main possible strategies. In the first attacking strategy the attacker only executes the first protocol step, the service request, and gives no answer to the

server in the next step. This strategy means that the attacker does not want to waste resources to even answer the request, but the reason can also be that the attacker cannot receive data from the server (e.g. in specific spoofing conditions).

If the attacker chooses the second strategy, then a false (e.g. random) solution is being sent back to enforce the server to verify the results and therefore let the server consume more resources.

If the third strategy is chosen, the attacker properly executes the steps of the client-side puzzle protocol. The attacker chooses this strategy to let the server execute the protocol steps following of the client-side puzzle protocol header, such as making a resource consuming digital signature creation.

The goal of the client-side puzzle is therefore to ensure that the attacker won't choose the third strategy for attack purposes, and therefore prevent to waste resources in the following protocol steps. The above mentioned strategies of the attacker is denoted by A_1, A_2 and A_3 .

The possible protection strategies of the server is modeled by the complexity level of the puzzle. For simplicity we assume that the server can choose between two computational complexity levels of the puzzle. The strategies of the corresponding strategies are denoted with S_1 (lower complexity) and S_2 (higher complexity). $G(A_j, S_k)$ denotes a game where the attacker chooses strategy A_j , while the server chooses strategy S_k .

Even this simplified game theoretical model can help to enhance the protection of the protocols against DoS attacks in a better way than the previously proposed methods.

With the description of the cost elements we can define the matrix of the game as shown in Table 1.

		S_1	S_2
		x_1	x_2
A_1	y_1	$M_{11} = c_c(1) \cdot \frac{R}{c_r}$	$M_{12} = c_c(2) \cdot \frac{R}{c_r}$
A_2	y_2	$M_{21} = (c_c(1) + c_v(1)) \cdot \frac{R}{c_r + c_g}$	$M_{22} = (c_c(2) + c_v(2)) \cdot \frac{R}{c_r + c_g}$
A_3	y_3	$M_{31} = (c_c(1) + c_v(1) + c_e) \cdot \frac{R}{c_r + c_a(1) + c_p}$	$M_{32} = (c_c(2) + c_v(2) + c_e) \cdot \frac{R}{c_r + c_a(2) + c_p}$

R: available resources; costs: c_r : service request; $c_c(k)$: making a challenge with complexity k ; c_g : wrong answer; $c_a(k)$: right answer for a challenge with complexity k ; $c_v(k)$: verification of the answer; c_p : service request; c_e : service provision

Table 1: Matrix of the game

As the solution of the game the participants can choose a pure or a mixed strategy. In the case of a mixed strategy the strategy can be defined by a probability distribution among the strategies, this is denoted by $X = \{x_1, x_2\}$ for the server and with $Y\{y_1, y_2, y_3\}$ at the attacker.

The protection against DoS attacks can be enhanced with my proposed approach that combines game theoretical analysis and client-side puzzle technique. A further advantage of my work is that with the usage of both methods the analysis of the DoS protection is easier and therefore better protection methods can be proposed. In the dissertation I show how the solution of the game can be computed in typical cases and also show that if the server uses a pure strategy then

the attacker will also choose a pure strategy, in this special game the mixed strategy degenerates into pure strategy.

Proposal for a client-side puzzle

Let $T = \{p_1, p_2, \dots, p_N\}$ be a publicly known, ordered set of N prime numbers. A set S of k primes is selected randomly from T . The selection can be made with or without replacement. Below, we assume that it is done without replacement (i.e., S is a subset of T). The analysis can directly be extended to the case of selection with replacement. The elements of S are multiplied together, and the product is denoted by m :

$$m = p_{i_1} \cdot p_{i_2} \cdot \dots \cdot p_{i_k}$$

My proposal for a client-side puzzle is to use one of the following challenges:

- *Puzzle 1:* The puzzle is the product m .
- *Puzzle 2:* Let m' be a modification of m , such that ℓ consecutive bits, $m_r, m_{r+1}, \dots, m_{r+\ell-1}$ are replaced with zeros in the binary representation of m . The puzzle is the resulting number m' together with position r .

For both puzzles, the task of the client is to find the prime factors of m (or m' in case of Puzzle 2), and respond with their corresponding indices in T . (considering that T is ordered)

It is not easy to propose a client-side puzzle challenge with the necessary properties against DoS attacks. The generation of the challenge should be fast and also the verification process should be easy. The storage need of the puzzle at the server side should be low. The computational need at the client side should be given. In some cases it is also needed that a parallelization at the solution at the client side to solve the puzzle should not be possible. The client should not be able to speed-up the solution with any optimization effort. My proposed solution has very good properties according to above mentioned needs.

Analysis of the proposed client-side puzzle

One important precondition of the practical use of the proposed puzzle is to give its computational need at the client side and to give exact information about the setting of the parameters. During the analysis I show how the necessary resources can be calculated at the client side therefore I give instructions how the client-side puzzle approach can be practically implemented.

In case of Puzzle 1, the client calculates its response in the following natural way: Let μ be a variable that is updated in each step of the computation. Initially, $\mu = m$. In step i , the client checks if p_i is a factor of μ (and hence of m). If so, then μ is updated to take the value of $\frac{\mu}{p_i}$ and we store the index i ; otherwise μ does not change. This procedure is repeated until all factors of m is found (i.e., μ becomes 1).

During the above computation, at least $k - 1$ divisions are made¹, where the divisors are n bit

¹Note that the last prime factor need not be checked by division.

size primes, and the size of the dividend decreases gradually from kn to $2n$. The average number of divisions is given by the following formula:

$$D(N, k) = \sum_{i=k}^N q_i \cdot (i - 1) \quad (1)$$

where q_i is the probability that the largest index in S is i , and it is computed as:

$$q_i = \frac{\binom{i-1}{k-1}}{\binom{N}{k}} \quad (2)$$

In case of Puzzle 2, the client is forced to do more calculations. The client may choose from the following two procedures:

- The client tries possible substitutions for the missing bits. If a substitution is incorrect, then the client will likely get prime factors that do not belong to set T . In this case, the client continues with choosing another substitution. The average number of divisions required is approximately $N2^{\ell-1}$, thus the complexity of solving the puzzle is increased with a factor of $2^{\ell-1}$ on average.
- The client directly calculates different products of k primes from set T until the tested product m' is obtained. The average number of multiplications required is

$$\frac{1}{2} \binom{N}{k} (k - 1)$$

My proposed client-side puzzle is well understandable and the implementation can be also easy. I also give the necessary parameter computation through the analysis of the computational need of the puzzle. I also provide a comparison with the hash-based client puzzles proposed by other researchers. After the publication of the proposal for the new client-side puzzle, I identified a possible attack against my scheme. The attack, and the protection against it is described in the dissertation.

Claim 2 – Protection based on traffic analysis against Denial-of-Service problems in Internet e-mail environment

Publications: [C1], [C4], [C6]

I propose a new protection method against DoS attacks based on network traffic analysis. The proposed method does not need the modification of the network elements outside the victim server and minimizes the number of legitimate sources blocked by the server.

By performing measurements I show that it is possible to carry out successful Denial-of-Service attacks against SMTP servers with only a low bandwidth usage, and the possibility is even more feasible when using content filtering combined with the SMTP server.

My proposed new protection method is based on traffic analysis. In a front-end module we identify DoS attacks and filter out attackers. I show how the probability of false positive and false negative error event can be computed.

I give an upper bound to the error rate of the detection algorithm, and to the probability of false identification. These calculations help us to design the real-life parameters of the protection system. Simulations confirm analytical results and help in further investigation of the sensitivity of the parameters

I designed an architecture for my system to be successfully inserted into a real-life SMTP scenario. Based on the proposed architecture I designed and developed a prototype of my proposed protection method.

Motivation: Performance measurements of SMTP content filters

Two DoS methods against Internet sites are reported most often:

- The attacker attacks a specific (programming) error to disable the service (e.g. ping of death)
- The attacker floods the server with a high volume traffic, and the server cannot handle the amount of traffic, or, sometimes even the available bandwidth is overloaded with the attacking traffic.

This two attacking behavior is most typical in today's Internet, but in the future more sophisticated attacking methods can appear. A protection against DoS attacks can be easily, or at least possibly deployed if the behavior of the attacker can be clearly distinguished from the behavior of a legitimate user. One of the typical DoS attacks is currently the SYN flooding (without IP spoofing) where the behavior of the attacker can be easily distinguished and therefore the sources of the attack can be detected and a successful protection can be performed.

A countermeasure against such attacks is still an important problem, mainly because of the organizational and coordinating efforts. However, when the attacker cannot be identified and distinguished from the legitimate users, we have a much harder problem. These attackers exploit the fact that on the application level the sending of a query to a server has only a low resource

consumption in contrast of the possible very high resource consumption at the server side. In this way the attacker can force the server to carry out complex and resource consuming efforts with only sending a few network packets.

A typical attack example is when the attacker sends e-mails to an e-mail server (SMTP server) to use up resources at the server, meanwhile only consuming a low amount of network bandwidth. This way the attacker is hardly distinguished in the network layer from the legitimate sources.

I carried out measurements to obtain information on the typical SMTP delivery performance, focusing on content filtering such as virus and spam detection. I wanted to investigate how much resource consumes the delivery of an email, what is the delivery throughput on the server. I was unable to find similar, scientific grade measurements about SMTP delivery performance. It was therefore unclear, who SMTP servers are resistant to DoS attacks or other DoS related problems (e.g. large amount of spam messages). I focused to investigate how the content filtering (spam and virus filtering) modifies the performance of the server capacity. According to my industrial experience it is not uncommon that a server is delaying the delivery of a large number of messages for hours. The literature in contrast does not explain what can we expect from a general e-mail server today.

To carry out measurements in this environment, we face to a number of difficulties. This can be the reason why we cannot find too much information regarding to this topic in the literature. To carry out measurements we have to know how the SMTP servers work. The SMTP servers have a few well distinguished phases during the delivery process. The processes on the e-mails may overlap and some processes can be delayed, and the delivery of a single e-mail can also be very different from the other e-mails. Meanwhile, the server has a lot of different parameters that can seriously affect the delivery performance, e.g. some times it also can communicate with third parties during the delivery process that consumes network bandwidth (e.g. RBL, reverse-DNS). The mail delivery process performance is therefore a complex and hard-to-investigate question. Another important problem is how we can make the necessary workload (test messages) during measurements. Sometimes the generation of test messages can also consume considerable resources (e.g. generating random e-mails for testing spam filters) and therefore this can affect the validity of our measurements. I carried out my measurements considering these problems and designed a standard measurement environment where the results can be compared to each other. The successful execution of my measurements needed a large amount of preparation work.

During my measurements I sent e-mails in the size of 4kb with a specially designed environment from multiple sources. The tested SMTP servers and content filters were selected from the most often used solutions. I collected the performance data in the case when we do not provide content filtering. In this case the delivery speed is 17-64 e-mail/sec in my standard environment.

In the second phase of my measurements I showed that a simple virus filtering (in my case Exim MTA combined with Amavisd and ClamAV) how deeply degrades performance. In contrast of the original 30 e-mail/sec performance the delivery rate declined to 6,81 e-mails/sec. This performance degradation is really significant and seriously harms the DoS resilience of the service.

Table 2 shows the measured performance in the case when we carried out both e-mail and spam filtering. The different spam-filtering configurations shown in the different rows.

Table 2: Delivery performance in the case of content filtering

used SpamAssassin modules and test e-mail	average time to delivery	std. dev.	average delivery rate (e-mail/sec)	std. dev.
Razor, bayes, fixed message	294.0	14.1	5.11	0.2
Razor, bayes random message	945.0	7.0	1.59	0.1
Local_tests_only, random message	448.0	15.5	3.38	0.2
No razor, bayes, random message	458.0	4.2	3.27	0.1
Razor, no bayes, random message	975.1	7.4	1.54	0.1
Razor, mysql-bayes, random message	789.5	9.6	1.90	0.1

As a summary of my measurements we can observe that the average e-mail throughput of a server can decline from the low 30 e-mails/sec to an even lower 2 e-mails/sec when we turn on spam and virus filtering. I show that it is possible to attack an SMTP server without consuming all of the available bandwidth. The needed bandwidth for a successful DoS attacks is highly depends on system properties, in my case it about 1 Mbit/sec. My measurements also show that the usage of content filtering highly affects the performance of the server and therefore makes it much more vulnerable to DoS attacks. In my specific case the needed bandwidth in the case of content filtering is only as large as 64 kbit/sec. These results show how important research area is the field of SMTP servers against DoS attacks, although there were not too many investigations in this field.

My results can be used as a background for investigations of the vulnerability of SMTP servers against DoS attacks. The results can also help to develop better performance tests on SMTP servers. Of course my measurements were limited in the number of products investigated and this can be broaden up to carry out larger-scale data collection or to investigate individual performance factors more deeply.

DoS front-end based on traffic measurements

Considering the DoS attack of the SMTP protocol I propose a solution when I identify the attacker from the legitimate users by traffic analysis. I tried to handle the case of DoS attacks where the modification of the protocol is impossible and therefore the introduction of many solutions, such as client-size puzzle is impossible. The attacker performs a simple query and the server uses serious resources to answer the query, therefore the resource needs of the parties are very unbalanced. The other possible case is when the attacker performs a distributed DoS attack (DDoS) ([4]) when the attacker has control over a significantly higher volume of resources than the attacked party (e.g. by using of so-called zombie computers).

In the above mentioned cases the possibility of the protection is depending on the possibility of statistical distinction of attacking and legitimate sources. If the attackers are not distinguishable then we have no serious chance to protect against them. In contrast, if the attackers are distin-

guishable from the legitimate sources, and according to my experience most cases they are, the we have the possibility to protect against them with the usage of network traffic analysis. This possibility was not deeply analyzed in the literature. I propose a method as countermeasure of DoS attacks and also analyze its mathematical background.

The most important discovery in the modelling of DoS attacks is the following: The attacker needs to more and more hosts to carry out attacks when the traffic from an attacking host is indistinguishable from the traffic of the legitimate sources. Meanwhile, if more and more computers are controlled by the attacker, the possibility of the detection of the attack (and law enforcement process against him) rises accordingly. The trade-off between this two parameters sets the optimal number of attacking sources and this result gives us a possibility for a successful protection against DoS attacks.

The algorithm of protection consists of the following sub-algorithms, which are executed consecutively:

- Attack detection algorithm, that uses statistics based on sliding windows to measure traffic jumps (algorithm A1), or detect the overflow of the buffer (algorithm A2) to detect attacks.
- Method for identification of the attacking sources, that is based on the identification the sources with the highest traffic to the server
- Suppression of the attacking traffic, based on the filtering of plausible attackers
- Algorithm to check the success of the suppression, investigating if the traffic returns to the normal state

A.) Detection Of The Attack

Algorithm A1.

The beginning of the attack is decided to be time \hat{t} , which is the time, when the following event occurs:

Event 1: The buffer length exceeds a pre-defined paramter L_1

Algorithm A2.

The beginning of the attack is decided to be time \hat{t} , which is the time, the following event occurs:

Event 2:

$$\hat{\lambda}(\hat{t}) > (1 + r)\bar{\lambda}(\hat{t}) \quad (3)$$

where r , $r > 0$ is a design parameter, $\bar{\lambda}(t)$ is a long time traffic level, while $\hat{\lambda}(t)$ is a short time measured traffic level.

Algorithm A3.

The beginning of the attack is decided to be time \hat{t} , which is the time, when the earlier of the two events Event 1 and Event 2 occurs.

B.) Identification Of The Attacking Sources

Starting at time \hat{t} , we measure the aggregate and the individual traffic levels (i.e. traffic per source). If we correctly identified an attack, i.e. $t^* < \hat{t} < t^* + \delta$, we can make measurements over the resting time $(t^* + \delta - \hat{t})$.

Let the output of this measurement be denoted by $\hat{\lambda}_r(t^* + \delta)$ and $\hat{\lambda}^{(i)}(t^* + \delta)$ for the aggregate level and for source i respectively.

As we cannot determine the exact traffic from the legal sources during the attack we use $\bar{\lambda}(\hat{t} - c)$ ($c > 0$) as an estimate on the mean aggregate traffic level of legal sources in time interval $[t^*, t^* + \delta]$ and we gain an estimate

$$\hat{\lambda}_a = \hat{\lambda}_r(t^* + \delta) - \bar{\lambda}(\hat{t} - c) \quad (4)$$

on the mean aggregate traffic level of attacking sources. The set Z of active sources is decomposed into

$$Z = Z_n \cup Z_a \quad (Z_n \cap Z_a = \emptyset) \quad (5)$$

where Z_n and Z_a are the sets of legal sources and attacking sources respectively. The identification algorithm outputs a subset Z_a^* of Z . This set should correspond to Z_a as closely as possible.

The identification of attacking sources is made by the following algorithm:

Algorithm B1.

Find the maximum-sized subset $Z_a^* = \{i_1, i_2, \dots, i_v\}$ of Z , which corresponds to sources with the highest measured traffic levels, such that

$$\sum_{j=1}^v \hat{\lambda}^{(i_j)}(t^* + \delta) \leq \hat{\lambda}_a \quad (6)$$

Algorithm B2.

Omit those sources from set Z which have been active at time $(\hat{t} - c)$, $c > 0$, and use Algorithm B1.

C.) Suppression Of The Attacking Traffic

Once we have successfully identified the attacking sources, the traffic suppression algorithm is straightforward:

Algorithm C.

Discard all incoming units with sources from the set Z_a^* .

D.) Checking Of The Success Of Suppression

Algorithm D.

In case of successful intervention, by running *Algorithm C* the buffer length has to retire below length L_1 within a timeout t_{out} . If it does not occur we should discard packets from further active sources beginning with the one with the highest measured traffic level, followed by a new checking of success. These steps are repeated until the wanted decrease in the queue length is reached.

Analysis of the protection module based on network traffic analysis

In my dissertation I analyze some of the properties of my proposed algorithms, such as the probability of false detection. Two types of false detection is possible, error Type I. is when an attack happens, but the algorithm does not identify it, while error Type II. is the case when the algorithm detects an attack while there was no attack at all.

Two hypotheses at time t can be defined:

In time slot t I test the following hypothesis about the state of the system

H_0 : state of no attack

H_1 : state of attack

Consequently Type I and Type II errors are the following:

$$P_I = P(\{\text{decision on } H_0 \text{ at time } t\} | H_1) (= 0), \quad (7)$$

$$P_{II} = P(\{\text{decision on } H_1 \text{ at time } t\} | H_0). \quad (8)$$

I assume that the system has been in normal state for a long time, when a change of state occurs. Considering detection algorithm A3 I can give the following rough upper bound on P_{II}

$$P_{II,A3} = P(\{\text{queue length} \geq L_1\} \cup \quad (9)$$

$$\{\hat{\lambda}(t) > (1+r) \cdot \bar{\lambda}(t)\} | H_0)$$

$$\leq 2 \cdot \max\{P_{II,A1}, P_{II,A2}\} \quad (10)$$

where the individual probabilities for the Algorithms A1 and A2 are the following:

$$P_{II,A1} = P(\{\text{queue length} \geq L_1\} | H_0) \quad (11)$$

$$P_{II,A2} = P(\{\hat{\lambda}(t) > (1+r) \cdot \bar{\lambda}(t)\} | H_0) \quad (12)$$

Probability $P_{II,A1}$ can be calculated (designed) using approaches of standard buffer design. Probability $P_{II,A2}$ is considered below, where I give upper bound on it.

$$P\{\hat{\lambda}(t) > (1+r) \cdot \bar{\lambda}(t)\} \leq \frac{1}{w_s r^2} \left(\frac{\sigma_n}{\lambda_n} \right)^2 \quad (13)$$

(w_s size of the measurement window, λ_n : speed of normal traffic, σ_n : std. dev. of normal traffic speed)

Beside the formal analysis of the success of the attack detection I also give analysis on the error probabilities of the identification of the attackers. I show that the best strategy of the attacker is to spread uniformly the attacking traffic among as many sources as he can to avoid the identification. I also give analysis and supporting simulation results on the parameters used in my proposed algorithms.

Research also proposed some special, methods to protect against network DoS attacks, but most of them are only viable when it is widely deployed. (e.g. [6],[7]), while my proposed solution does not need such large-scale modifications.

The applicability of my proposed method depends on the protected service or protocol. My proposal and my analysis can help to design protocol-specific solutions too.

Prototype of protection based on traffic level measurements against SMTP DoS attacks

To show the applicability of my proposed method I designed and developed a prototype of the protection. The prototype is based on an SMTP server extended with virus filtering. An SMTP server is a complicated software that performs a lot of different tasks, and therefore the extension and modification of such software components is not trivial. To successfully deploy my proposed method I had to insert a solution to enable traffic level measurements, I had to develop the statistical engine. Design of test tools to verify the work of the statistical engine were also needed and I also had to integrate the module for filtering the attackers into the SMTP service.

The SMTP system consists of the following modules: Exim MTA (SMTP server), Amavisd-new content filtering middleware, and a virus filtering program that is executed by the Amavisd if necessary. This system is vulnerable to DoS attacks as the filtering of the viruses consumes lot of resources and therefore the attacker can fully overload the server by using only a considerably low amount of network bandwidth (compared with typical available bandwidth).

The topology of my prototype is shown on Figure 1. The modules shown with dark boxes were developed and inserted in the original system. The following modules were designed and developed: a TCP wrapper, a DoS front-end client and a DoS front-end engine. The TCP wrapper waits for incoming connections and if a connection is established, connects to the DoS front-end client and asks if the client is authorized to send an email (not filtered out). If no filtering is necessary then the wrapper transmits the data of the connection to the SMTP server (MTA). The DoS front-end client is a very simple program to communicate with the front-end engine to inform the engine about the new connection and to query the engine whether the client should be filtered or not. Then the answer is sent back to the wrapper. The front-end engine consists of multiple modules: It contains a statistical engine to continuously monitor network traffic by the maintenance of state variables, and a decision engine to decide about the attack status and to identify attackers. The statistical engine executes my the previously defined algorithms.

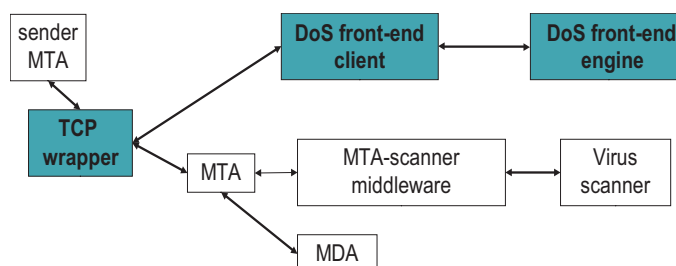


Figure 1: Topology of the prototype

My prototype shows that my proposed method is workable in real-life situation. Meanwhile the prototype is advanced enough to use in production environment, and it can provide real protection against DoS attacks. During the usage of my prototype extra caution should be taken, as the prototype uses filtering against the attackers. Due to the filtering there is a chance of false positives, therefore the fine-tuning and maintenance of the system is necessary.

Claim 3 – Protection against DoS preparation attacks

Publications: [J2], [J3], [C7], [C8], [C5], [O13], [C9]

In this claim I propose new methods to counter some DoS preparation attacks, such as preventing the Directory Harvest Attack and identifying malware infected computers. Concerning Directory Harvest Attacks I propose a new architecture and method to protect against the attacks and I define an optimal algorithm of the DHA attack. My proposed protection method is analyzed by simulations and I also created a software prototype.

I analyzed e-mail Directory Harvest Attacks. DHA is typically used to collect e-mail addresses for spamming, and at the same time, it can lead to a DoS condition by itself. I designed an optimized DHA attack, where the optimization is based on the probability distribution of e-mail local parts and the expected number of valid addresses in the targeted domain. I give formula for the expected number of successfully collected valid e-mail addresses. I prove that in my model the attack is optimal, the expected number of the successfully collected e-mail addresses is maximal. I illustrated with simulation results that the optimal attack method is more efficient than the non-optimized attack. The simulation results based on real data supported the feasibility and efficiency of my proposed algorithm. I designed a new centralized, real-time blacklist (RBL) based architecture against DHA attacks. I also developed a prototype system based on the proposed architecture to show the applicability of my proposal.

Optimized Directory Harvest Attack(DHA)

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 attacker sends a trial e-mail message for a target e-mail address. If the attacker receives an error message, then the target e-mail address probably does not exist. In contrast, if the attacker does not get an error message, then the e-mail address is possibly valid and can be collected into a list. 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 ([8],[9]). No detailed information is available how these commercial programs provide protection against 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.

I found that most of the attackers use a fixed size dictionary during the DHA attack and they attack only a few selected domains. In the dissertation I show, that this attack method is not optimal. I show that the attack of a resource-constrained attacker can be optimized by using certain additional information (frequency distribution of the known e-mail local-parts in a region, estimated number of e-mail addresses in the target domains). Although currently the attacks are not optimized in the presented way, we can expect that the attackers will also optimize their attack in the future. The analysis of the optimal attack is therefore necessary to be able to give effective countermeasure against the DHA. In the optimal case the attacker distributes its effort according to the estimated number of valid e-mail addresses of the target domains. If the estimated number of valid e-mail addresses is larger, then the attacker uses up more trials from the dictionary against the targeted domain.

I define the pseudocode of the optimal behavior 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

```

N_D : number of targeted domains; $i[d]$: number of tried words in the domain with index d ; t : resource constraint of the attacker (possible trials); U^d : the set of users in domain d ; W : list of e-mail address local_parts in descending order of the frequency; p denotes possibility, p_{max} , $target$, d : internal variables
 The attacker chooses a target domain and the next element of the dictionary against that domain in each step. This way the attacker tries that word that is most probably valid in that step.

Analysis of optimal attack

The expected number of the successfully collected addresses for the normal, non-optimized attack method is the following:

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

where S is the number of successfully collected addresses, and T is the number of executed trials against the specific domain.

For the optimized attack the expected number of collected addresses is given with the following formula:

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

For further analysis I use the Lagrange multiplier technique. We try to find the maximum of

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

where $n_i = N_U^i$. For the case when $h(x)$ is a function in the form of $h(x) = a/x^b$, then $h'(x) = c/x^d$ $d = b + 1$, and we get the following result:

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

In the dissertation I analyze the optimal attack and prove that it is optimal. The results are furthermore supported with simulation results. For my simulations I use real-life data and the results show that the optimal attack can identify significantly larger amount of valid e-mail addresses.

The simulation covers two scenarios, in one case I defined 11 target domains, in the other 6 domains were given. In the first case 1730 valid addresses were considered, while 23000 valid addresses were given to the other scenario. The table 3 shows that the Algorithm 2 (optimal attack) resulted a significantly higher number of successfully identified e-mail addresses to the attacker with the same resource consumption.

Table 3: DHA simulation results

N_D	Total user	Identified addresses, Alg. 1.	Identified addresses, Alg 2.
11	1730	87	180
6	23000	5395	6754

Protection method against DHA with centralized approach

I propose a network based, centralized protection against the DHA attacks, based on real-time blacklist (RBL). I am not aware on any other detailed protection in the literature proposed against DHA attacks. The proposed protection works the following way: We continuously monitor the e-mail traffic at the

servers to identify e-mail messages without valid recipients. If we detect a number of messages coming from a single IP address to the server with illegal recipients, then it is reported to a central system. The central system functions as a trusted third party, who collects information from the clients, analyzes and aggregates the information and defines a list of the IP addresses of the suspected attackers. The clients can perform a query to the central server before the arrival of each message determining if the sending server is a suspected attacker or not. If the sender computer is identified as an attacker, the client can interrupt the mail delivery and close the connection. The proposed method is one kind of real-time blacklist (RBL) and therefore it can be implemented with some level of compatibility to existing solutions.

I designed and developed a prototype to show the applicability of my proposed protection. The protection consists of a log analyzer, a reporting tool for suspected attackers, a central server engine for handling reports and to maintain the database of suspected attackers. The solution also contains a filtering tool that can query the server about the status of a e-mail sending party. The filtering tool is an extension of the DoS front-end prototype (that uses traffic level measurements to protect against DoS attacks).

Simulations based on the different algorithms and parameters were carried out to show the behavior of the proposed protection methods. In the simulations I show that using higher number of detectors in the network helps the identification of the attackers, but also causes higher ratio of false positives. I also show similar results when the attack detection threshold is low. Simulations were also carried out to show how different aging parameters affect the number of false positives. Finally, an adaptive technique is proposed for lowering the number of false positives.

6 Application of the results

During my research I always tried to support my results with practical prototypes. I designed and developed the following prototypes during my work:

- DoS front-end based on traffic level measurements
- Protection prototype against DHA attacks

The protection against DHA and DOS (based on traffic level measurements) were deployed into real-life product environment to protect hundreds of users of some SMTP servers. To successfully deploy my proposed methods I had to refine parameters to avoid the false positives. I also had to use white-lists of some servers, that the protection detects as attacker. These hosts in fact can be considered as real attackers, but many times these are servers of reputable Internet providers forwarding attacks from infected hosts in their internal network.

The SMTP related protection methods are also used in our research during the DESEREC (Dependable security by enhanced reconfigurability) project of the EU FP6 framework programme (contract no. 026600).

References

- [1] K. Matsuura and H. Imai "Protection of authenticated key-agreement protocol against a denial-of-service attack" In Proceedings of the International Symposium on Information Theory and Its Applications (ISITA'98), pp. 466–470, October 1998.
- [2] J. Leiwo, T. Aura, and P. Nikander "Towards network denial of service resistant protocols" In Proceedings of the IFIP SEC 2000 Conference, August 2000.
- [3] C. Dwork and M. Naor "Pricing via processing or combatting junk mail" In Advances in Cryptology – Crypto '92, Springer-Verlag, LNCS volume 740, pp. 139–147, August 1992.
- [4] F. Lau, S. H. Rubin, M. H. Smith, and L. Trajovic "Distributed denial of service attacks" In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, pp. 2275–2280, October 2000.
- [5] A. Juels and J. Brainard "Client puzzles: A cryptographic countermeasure against connection depletion attacks" In Proceedings of the IEEE Network and Distributed System Security Symposium (NDSS '99), pp. 151–165, February 1999.
- [6] Ferguson, P. and D. Senie "Network Ingress Filtering: Defeating Denial Of Service Attacks Which Employ IP Source Address Spoofing" RFC 2827, May 2000.
- [7] Ioannidis, J. and S. M. Bellovin "Implementing Pushback: Router-based Defense Against DDoS Attacks." In Proceedings of Network and Distributed System Security Symposium, Reston, VA, USA, Feb. 2002, The Internet Society.
- [8] Kerio MailServer - state-of-the-art secure email server, http://www.kerio.com/kms_home.html. 2004.
- [9] Postini Enterprise Spam Filtering. The Silent Killer: How Spammers are Stealing Your Email Directory, <http://www.postini.com/whitepapers/>, June 2004.
- [10] S.Hird "Technical Solutions for Controlling Spam" In the proceedings of AUUG2002, Melbourne, 4-6 September, 2002.
- [11] Ronald F. Guilmette, wpoison – small CGI script to combat junk email, <http://www.monkeys.com/wpoison/>.
- [12] Devin Carraway - Sugarplum automated spam-poisoner, <http://www.devin.com/sugarplum/>.
- [13] The HoneyNet project, Know Your Enemy: Learning About Security Threats. Addison-Wesley, 2002.
- [14] L. Spitzner "Honeypots: Tracking Hackers" Addison-Wesley, 2002.

Publications

Journal papers

[J1] B. Bencsáth, I. Vajda, Internet Denial of Service attacks in game theoretical model (in hungarian), *Alkalmazott Matematikai Lapok* 23, 2006, pp. 335-348.

[J2] B. Bencsáth, I. Vajda, Efficient Directory Harvest Attacks and Countermeasures, *International Journal of Network Security*, vol 5. no 3. pp. 264-273. 2007.

Citations (1):

C. Chris Erway. MicroID considered harmful (to privacy). Technical Report, Brown University, 2008.

[J3] Géza Szabó, B. Bencsáth, Protection against DHA attack with central filtering (in hungarian), *Híradástechnika*, 2006, vol. LXI, pp. pp. 2-9, 05.

[J4] I. Askoxylakis, B. Bencsáth, L. Buttyán, L. Dóra, V. Siris, D. Szili, and I. Vajda Securing Multi-operator Based QoS-aware Mesh Networks: Requirements and Design Options, *Wireless Communications and Mobile Computing (Special Issue on QoS and Security in Wireless Networks)*, accepted for publication in 2009.

Conference papers

[C1] B. Bencsáth, M. A. Rónai Empirical Analysis of Denial of Service Attack Against SMTP Servers, *Proceedings of The 2007 International Symposium on Collaborative Technologies and Systems, IEEE*, 2007, pp. 72-79.

[C2] B. Bencsáth, L. Buttyán, I. Vajda, A game based analysis of the client puzzle approach to defend against DoS attacks, *Proceedings of IEEE SoftCOM 2003* 11., Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split, 2003, pp. 763-767.

Citations (13):

E. Altman and K. Avrachenkov and G. Miller and B. Prabhu. Discrete Power Control: Cooperative and Non-Cooperative Optimization, *Proc. of IEEE Infocom 2007*, Anchorage, Alaska, USA, May 6-12, 2007.

Mehran Fallah, A Puzzle-Based Defense Strategy Against Flooding attacks Using Game Theory, *IEEE Transactions on Dependable and Secure Computing*, 12 Feb. 2008.

A. Patcha and J-M. Park. A Game Theoretic Formulation for Intrusion Detection in Mobile Ad Hoc Networks, *International Journal of Network Security*, Vol. 2, No. 2, 2006, pp. 131-137.

A. Patcha. A game theoretic approach to modeling intrusion detection in mobile ad hoc networks, *IEEE Workshop on Information Assurance and Security*, June 2004.

T.J. McNevin and J.M. Park, and R. Marchany Chained puzzles: a novel framework for IP-layer client puzzles *Wireless Networks*, 2005 *International Conference on Communications and Mobile Computing*, pp. 298-303.

Yi Gao, Willy Susilo, Yi Mu, and Jennifer Seberry, Efficient Trapdoor Based Client Puzzle Against DoS Attacks, *Book Chapter in Network Security*, 2006

T. J. McNevin, J.M. Park, and R. Marchany. pTCP: A Client Puzzle Protocol For Defending Against Resource Exhaustion Denial of Service Attacks, Technical Report TR-ECE-04-10, Dept. of Electrical and Computer Engineering, Virginia Tech, Oct. 2004.

V.Laurens and A. El Saddik and A. Nayak. Requirements for Client Puzzles to Defeat the Denial of Service and the Distributed Denial of Service Attacks The International Arab Journal of Information Technology. Vol. 3. No. 4., pp. 326-333, 2006.

Lin, C. and Wang, Y. and Wang, Y. and Beijing, PR. A Stochastic Game Nets Based Approach for Network Security Analysis CHINA 2008 Workshop (Concurrency methOds: Issues aNd Applications), pp. 24-35.

Yuanzhuo Wang, Chuang Lin, Yang Yang, Junjie Lv, Yang Qu, A Game-Based Intrusion Tolerant Mechanism for Grid Service,pp.380-386, Fifth International Conference on Grid and Cooperative Computing (GCC'06), 2006

Network Research Foundations and Trends EU FP6 project, Deliverable D4.1 State-of-the-art report on tools and techniques for achieving Autonomous Network Operation, 2006.

Sen, J. and Chowdhury, P.R. and Sengupta, I. A Mechanism for Detection and Prevention of Distributed Denial of Service Attacks, Lecture Notes in Computer Science, Vol. 4308, pp. 139-144, Springer, 2006.

A. E. Goodloe. A foundation for tunnel-complex protocols. PhD Thesis, University of Pennsylvania, 2008.

[C3] B. Bencsáth, I. Vajda, A game theoretical approach to optimizing of protection against DoS attacks, presented on the Second Central European Conference on Cryptography (Hajducrypt), July, 2002

[C4] B. Bencsáth, I. Vajda, Protection Against DDoS Attacks Based On Traffic Level Measurements, 2004 International Symposium on Collaborative Technologies and Systems, The Society for Modeling and Simulation International, 2004, Waleed W. Smari, William McQuay, pp. 22-28., The Society for Modeling and Simulation International, San Diego, CA, USA, January, Simulation series vol 36. no. 1., ISBN 1-56555-272-5.

Citations(6):

Tan, H.X. and Seah, WKG, Framework for statistical filtering against DDoS attacks in MANETs,2005. Second International Conference on Embedded Software and Systems, pp. 456-465, 2005.

Kumar, K. and Joshi, RC and Singh, K. An Integrated Approach for Defending Against Distributed Denial-of-Service (DDoS) Attacks, IRISS 2006, Madras, 2006.

Sardana, A. and Joshi, R. and Kim, T. Deciding Optimal Entropic Thresholds to Calibrate the Detection Mechanism for Variable Rate DDoS Attacks in ISP Domain, International Conference on Information Security and Assurance, pp. 270-275, ISA 2008, 2008.

Kumar, K. and Joshi, RC and Singh, K. An ISP level Distributed Approach to Detect DDoS Attacks, Innovative Algorithms and Techniques in Automation, Industrial Electronics and Telecommunications, pp. 235, ISBN: 978-1-4020-6265-0, Springer, 2007.

El Hassan, A.C.M. and Maalouf, S. and Zouheiry, A. A Survey of DDoS Defense Mechanisms, American University of Beirut.

Yuan-Shun Dai, Xukai Zou, Yi Pan. Trust and Security in Collaborative Computing (book), ISBN 9812703683, World Scientific, 2007.

- [C5] B. Bencsáth, The problems and connections of network virus protection and the protection against denial of service attacks, Proceedings of the Networkshop 2004 Conference, NIIF, Hungary, 2004, NIIF, Hungary.
- [C6] Géza Szabó, B. Bencsáth, Statistical analysis of the results of the DHA protection system (in Hungarian), Proceedings of Networkshop 2006 conference, NIIF, 2006, NIIF.
- [C7] B. Bencsáth, I. Vajda, Efficient Directory Harvest Attacks, Proceedings of the 2005 International Symposium on Collaborative Technologies and Systems, pp. 62-68., IEEE Computer Society, July 2005.
- [C8] B. Bencsáth, Az internetes vírus- és spamvédelem rendszerszemléletben, HISEC 2004 konferencia, 2004, 10., Budapest, in Hungarian.
- [C9] B. Bencsáth, Géza Szabó, Components to improve the protection against spam and viruses, HSN LAB Workshop, 2005, Jun.

Other publications

- [O1] B. Bencsáth, I. Zs. Berta, Empiric examination of random number generators of smart cards, HTE-BME 2002 Korszerű távközlő és informatikai rendszerek és hálózatok konferencia, BME, 2002, BME.
- [O2] B. Bencsáth, I. Vajda, Collecting randomness from the net, Proceedings of the IFIP TC6 and TC11 Joint Working Conference on Communications and Multimedia Security 2001, Kluwer, 2001, pp. 105-111, Kluwer, May.
- [O3] I. Zs. Berta, B. Bencsáth, Hiteles üzenet küldése rosszindulatú terminálról, NetWorkShop2004, NIIF, CD Proceedings, Győr, 2004.
- [O4] B. Bencsáth, S. Tihanyi, Home-made methods for enhancing network security (in Hungarian), Magyar Távközlés, 2000, vol. X, no. 4, pp. 22-27..
- [O5] B. Bencsáth, T. Tuzson, B. Tóth, T. Tiszai, G. Szappanos, E. Rigó, Sz. Pásztor, M. Pásztor, P. Papp, P. Orvos, P. Mátó, B. Martos, L. Kún, Z. Kincses, T. Horváth, M. Juhász, B. K. Erdélyi, A. Bogár, G. Vid, Az informatikai hálózati infrastruktúra biztonsági kockázatai és kontrolljai, IHM - MTA-SZTAKI, 2004.
- [O6] I. Zs. Berta, I. Vajda, L. Buttyán, B. Bencsáth, T. Veiland, E-Group Magyarország Specification of the Hungarian electronic ID card (HUNEID) Információs Társadalom Koordinációs Tárcaközi Bizottság, Intelligens Kártya Munkacsoport, <http://www.itktb.hu>, 2004
- [O7] B. Bencsáth, Simple, free encrypted tunnels using linux, Presented on Networkshop 2000, Gödöllő, Hungary, 2000
- [O8] I. Vajda, B. Bencsáth, A. Bognár, Tanulmány a napvilágra került Elender jelszavakról, 2000, Apr. (átvéve: Chip, Alaplap, On-line oldalak)
- [O9] B. Bencsáth, S. Tihanyi, Problem areas of the security aspects of network operating systems, Scientific student groups (TDK) 1999

- [O10] B. Bencsáth, Multiple Security Flaws Lead to Netenforcer Privilege Escalation (TAR Issue Details), report, BugTraq, <http://www.securiteam.com/securitynews/6V00R0K5QY.html>, CVE-2002-0399, 2002.
- [O11] B. Bencsáth, Sudo Private File Existence Information Leakage Vulnerability, report, Bugtraq <http://www.securityfocus.com/bid/321>, CAN-1999-1496, 1999.
- [O12] A. Szentgyörgyi , G. Szabó , B. Bencsáth. Bevezetés a botnetek világába (in Hungarian), Híradástechnika, 2008. vol. LXIII, Nov. 2008. (Received the HTE Pollák-Virág Award)
- [O13] B. Bencsáth, I. Vajda, Trap E-mail Address for Combating E-mail Viruses, Proceedings of IEEE SoftCOM 2004 12. International conference on software, telecommunications and computer networks, University of Split, 2004, pp. 220-224, University of Split, October.