

# Fibered Guard – A Hybrid Intelligent Approach to Denial of Service Prevention

Marvin Oliver Schneider and Jacques Calmet

University of Karlsruhe (TH), Institute IAKS, Am Fasanengarten 5, 76131 Karlsruhe, Germany  
{marvin, calmet}@ira.uka.de

*Abstract*-This paper describes the system “Fibered Guard”, which is a hybrid intelligent web access management approach for the prevention of Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks alike. The system makes use of a logically fibered data structure to handle the necessary information, whereas a Data Mining plugin takes care of the conclusion engine. As the structure of Logical Fibered can handle millions of queries at once and treats global and local data in the same architecture, the system is a very promising solution, which has the potential to revolutionize the treatment of DoS. This paper presents the underlying principles as well as a system design overview of “Fibered Guard”.

## I. INTRODUCTION

Denial of Service attacks represent a serious security problem, which has brought well-known Internet services to complete inoperability. In a world, which relies on Internet banking, online shopping facilities and other handling of sensitive data by electronic means, DoS and DDoS attacks represent an issue of high scientific and economic interest.

Several approaches were developed to mitigate the effects of attacks (compare section III). None of them, however, aptly caught the necessities on a level abstract enough to develop a tool, which might treat all forms of different attacks at once.

This form of processing is necessary to avoid uncovered leaks between various systems (all of which only treat a specific attack) as well as eventual interferences between solutions or unnecessary system overload.

Our approach is a monolithic solution for DoS and DDoS defense.

## II. PROBLEM DOMAIN

By definition, Denial-of-Service attacks prevent legitimate users from accessing network resources, typically by producing an overload situation or taking advantage of known system bugs [8] [12].

There is a reasonable number of attack methods. This section shall give a short overview in order to convey an idea of the problem domain. Detailed approaches on taxonomy may be consulted at [22] and [15].

As a matter of fact, DoS attacks do not require a great amount of technical knowledge, which is one of the reasons, why they are so dangerous. Example of such a technically simple attack could be the so-called “Smurf”. In this attack, the attacker sends ICMP ECHO packets to the victim’s network broadcast address, employing forged IP data to make believe that the victim’s machine started the request. All machines on

the victim’s network answer to the fake request and the victim may be completely flooded with network traffic [18] [21]. Already with this simple attack, the principle of amplification, which led to the development of DDoS methods, may be observed.

Another classic, simple and effective attack is the so called “SYN-Flood”. It is based on design failures of the TCP/IP protocol, which was developed in times, when the vast development of the Internet of nowadays was not taken into account yet (see [3]). The intruder sends SYN packets from his system to the victim, whereas using a forged and unreachable IP address. The victim’s system will answer SYN/ACK to establish the connection, but will never receive an answer, thus leaving the connection in wait state. As the number of possible connections is limited and these incomplete connections are only closed upon timeout, with a few SYN packets, the attacker may already provoke that the victim’s system cannot establish any new connections [21] [33] [35].

Not based on wrong design, but moreover on real bugs in the system, DNS-attacks force the domain name server to put wrong information in cache, preventing that sites are mapped correctly, thus effectively blocking access [21]. Another example of an attack taking advantage of a bug is the so called “Teardrop”, which works upon a problem of IP-packet division [12] [21]. Whereas a simple bugfix eliminates these problems, bugs yet remain a constant feature of any new system’s versions, thus opening potential doors to new attacks of this kind.

Finally a variety of Distributed Denial of Service (DDoS) attack methods exists. A DDoS attack is characterized by the use of not only one machine in the attack, but the distribution of malicious software over a number of network nodes, which are under the remote command of the attacker and may all be sent to attack a victim a once, thus amplifying greatly the damage inflicted (figure 1). Several tools were developed by hackers in order to simplify the attack mechanism. Examples of such applications are “TFN”, “TFN2K”, “Trinoo”, “WinTrinoo” and “Stacheldraht” [21]. Whereas the methods have become more sophisticated over time with schemes of encryption, random ports and automatic updates, the general architecture has remained a “clients-handlers-agents” structure. The attacker installs handlers and agents in a clandestine way on poorly protected host machines. The handlers are guided via control streams by the client (attacker). In a second step the handlers instruct the agents to attack the victim. This structure has the advantage of hiding important information from traceback and besides increasing the attack spectrum vastly.

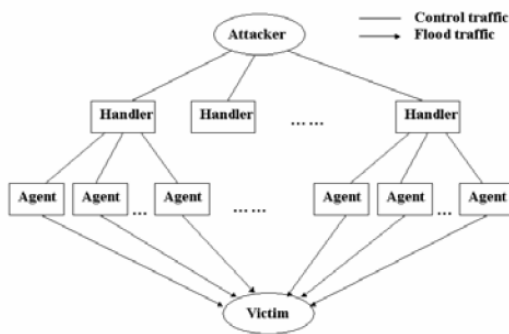


Fig. 1. DDoS attack structure [24]

### III. DEFENSE APPROACHES

The classical and also rather simple defense approach of Ingress/Egress Filtering relies on the fact that any legal packet from a domain must have an address valid in that domain. Thus addresses from other domains, which are obviously fake, may be filtered away either at the Internet Service Provider (ISP) (in the case of Ingress Filtering) or at the exit point of the customer network (Egress Filtering) [13] [32]. Undesired connections are reduced to the ones, which have legal domain addresses, but they are not fully eliminated.

IP Traceback methods seek to trace the route back to the attacker. An example would be the IP Traceback presented by [31]. In this approach packets receive marks by routers they pass through, which enhances the possibility to trace the route to the attacker. Another example is the Tracing of Spoofed Packets introduced by [6]. It takes advantage of the fact that small IP bursts on the way to the attacker will diminish the attack intensity over a short period of time. Thus, IP bursts are sent to all possible connections and the current link is followed if the attack intensity is reduced. This is done up to the attacker's machine.

In order to cope with congestions at routers caused by DoS attacks, a choice of congestion control algorithms may be used. Examples are Fair Queuing (FQ) [11] and Random Early Detection (RED) [14], among others. These algorithms do not seek the source of the problem, but only mitigate its negative effects on network traffic.

Finally, there are several interesting intelligent approaches. The Data Mining Approach to Intrusion Detection [19] uses Data Mining techniques in order to classify a situation into normal or abnormal. For this, data is learned from datasets provided by the administrator and certain characteristics are chosen. [17] proposes mobile agents for the analysis of DoS, producing less extra network traffic and enhancing speed and mobility. The approach in [5] shows a system, which uses Time Dependent Deterministic Finite Automata (T DFA) to detect the "language of DoS" and with it the necessary characteristics to overcome the problem. The analysis may be done with current or historical data. Finally [4] gives a connectionist method to intrusion detection. It employs a Self Organizing Map (SOM) for clustering and a Multilayer Perceptron for analysis and detection.

Web Access Management is the administration of access on resources throughout a network, and especially on the Internet. The field may be divided into Identity and Access Management and Internet Access Management. Web Access Management is a central security issue, which has been investigated by a whole range of commercial systems (see examples in [7] [16] [23] [30]).

According to [20] Identity and Access Management is one of the main concerns of IT-security. Its task is to authenticate users and manage access to resources accordingly. In the most common cases this may be done by the means of passwords, but also using keys or even biometrics [9]. Rules must define the profile of each user and open only the resources he is allowed and meant to access.

In Identity and Access Management security competes with the ease of use and therefore a delicate balance has to be found. This is especially true when speaking of E-Commerce where restrictive and complicated policies may drive away customers whereas neglected security may result in considerable financial damage.

Internet Access Management is the management of outgoing network connections, i.e. the administration of the visualization of content on the side of the client. This form of administration uses to be implemented in order to prevent individuals from accessing inappropriate content (mainly children accessing adult content and employees consulting sites that are not related to their working activity). Especially in an office scenario the problem is omnipresent. A survey given in [10] with 451 employees shows that more than 2/3 of them access online news sites, E-commerce sites, job searches etc. This behavior does not only put the productivity of these individuals in question, but also has negative consequences on bandwidth.

#### A. Standard Architecture

A standard network architecture, which many enterprises nowadays adopt, is given in [2]. With it, the process of connection is modified, so that the user may not connect directly anymore to the Internet, but instead, has to go through a Proxy server. Inside the Proxy server a rating database defines, which access shall be granted to which users. It then forwards the access to the firewall or blocks it. The firewall itself ensures that only requests from the Proxy server are taken and no direct connections can be made.

In the same manner incoming connections may be blocked or granted, i.e. a firewall may manage accesses from the Internet to local addresses.

#### B. Firewalls

There are several types of firewalls with different mechanisms to administer connections. Some principal types are the following:

A basic kind of filter, which may be found in commercial firewalls, is the so called packet filter. This rather simple algorithm decides whether a packet shall pass or be blocked on the basis of its origin or destination address. All decisions are taken on the current packet only. ISPs may already provide

such a technology for their users, functioning on the entry or exit point of the network [13].

Application level filters are firewalls that examine the contents of the packets rather than their IP information. Thus viruses, obscene words or defective formats may be blocked at the filter, which works on a “store-and-forward” basis, i.e. incoming code is stored, examined and forwarded if everything is considered in order. Whereas user mobility is increased by this type of firewall, a reasonable performance loss may occur [36] [37].

Circuit level gateways are firewalls that work exclusively on problems of TCP/IP traffic. As this protocol has considerable flaws and can easily be tricked, a retransmission of the code byte-by-byte to its destination has positive effects. Other control methods may function in parallel [36].

Dynamic packet filters are the most common form of existing firewalls. Just as simple packet filters they examine the packets and discard some of them upon their characteristics. The difference is though, that the context is maintained. Changes may be done dynamically and data may be retransferred as with a circuit level gateway [9].

Distributed firewalls offer an interesting alternative to the common central instance. The distribution means that every node has its own firewall with all the necessary information, which makes the architecture itself more fault tolerant but also requires more space and simultaneous updates are necessary.

## V. LOGICAL FIBERING

### A. Basic Concept

Logical Fibering originates from the philosophical work of Günther which leads to Polycontextural Logic (PCL) [26] [27]. The belief that the world consists of many basic logical spaces at different places was inserted into a logical context and further developed.

Logical Fiberings actually use the mathematical constructions of fiber bundles, whereas geometrical spaces are replaced by logical spaces. Conventional fiber bundle constructions can describe geometrical images. Examples are shown in figure 2:

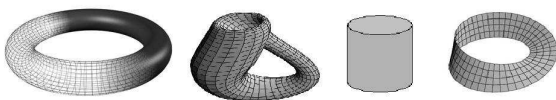


Fig. 2. Examples of trivial and non-trivial fiber bundle constructions [1]

A typical fiber space consists of the elements  $\xi = (E, \pi, B, F)$ .  $B$  is the base space, into which the enumerators are placed in the case of a Logical Fibering.  $E$  is the total space with all fibers.  $F$  is a typical fiber. Finally  $\pi$  defines the projection map  $\pi : E \rightarrow B$ . An abstract fiber space is denoted by  $\xi = (E, \pi, B)$ . A Logical Fibering is actually an abstract fiber space, in which  $F$  is a logical structure. In this case  $E$  is the disjoint union of all fibers with the base space  $B$  as the indexing set.

The simplest case of fibering is so called trivial fibering with  $\xi = (E, \pi, B, F)$  and  $E=B \times F$  whereas  $\pi$  is the first projection. This is also called a free parallel system as reasoning takes

place independently at different places, i.e. the global values are a mere coproduct of the local ones.

In more complex scenarios the possibilities of forming logical connectives within each subsystem arises. This bivariate operation may be expressed by figure 3 [25].

Bivariate operations can spread images over several subsystems ( $L$ , indexed by  $\alpha, \beta, \gamma, \delta$ ). These operations are also called “transjunctions” and they may be used to model communications in a Multiagent System [28].

One of the main objectives of Logical Fibering is the possibility to mix heterogeneous structures, simply treating them as distinct fibers.

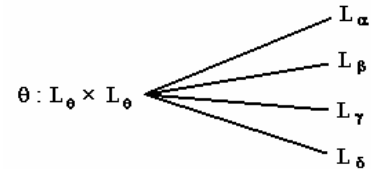


Fig. 3. Correspondent mapping to bivariate operation

### B. Applications

In literature [25] [29] mainly Multiagent applications are discussed. An example [25] [29] is a scenario of three cooperating robots, which shall perform a welding task together, one depending on the correct action of the other. Using a Logical Fibering structure, the base space  $B$  is made up of three points representing the three robots, whereas  $E$  contains the three fibers with the local knowledge of the robots. Cooperation is fulfilled via transjunctions.

[28] presents a general mathematical framework for agent systems in the following way:

A set of agents is given and shall be modeled. This set shall be denoted by  $A_g$ , whereas  $I$  defines the index, which is to be used in the base space  $B$ , being  $g$  any value from the index  $I$ . Agents of  $A_i$  are defined by  $A_{\xi}$  and may take influence on the local state space of  $A_i$  as well as on the state space of other agents. Agents must have the ability to communicate among them and show autonomy, also possibly refusing a communication or beginning one. The agents shall store knowledge inside a so called “belief set”. The notation suggested is  $Var^i$  for agent  $A_i$ . Inside the universe of agents, all agents belong to the same family by formal means.

## VI. SYSTEM DESIGN

### A. General Considerations

The system “Fibered Guard” is currently being developed. The system’s purpose is to offer an intelligent solution which manages web and network accesses preventing DoS and DDoS.

The system is being programmed with J2SE, which was chosen in order to insure portability of the system as Java-classes are platform independent.

The selected database is MySQL because it offers a sound SQL environment with a free-of-charge implementation.

## B. Logical Fibering Architecture

In the Logical Fibering structure each connection is saved [34]. In the case of our system, the incoming IP-address is chosen as an enumerator (see figure 4). Apart from connection specific data, the general health state of the system is also stored as global structure.

Each enumerator (IP) corresponds to an agent, which takes care of all connections coming from a specific address. These agents make use of sensors and actuators, which, in a Logical Fibering fashion, are connected by transjunctions in a many-valued logic system. Basically, the sensors obtain all possible information about the connection whereas the actuators define actually what should be done.

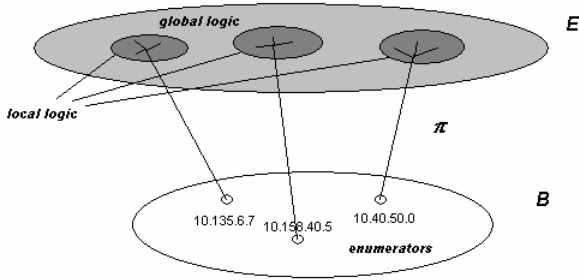


Fig. 4. Logical Fibering Structure [34]

There will be  $n$  agents in the environment, which we name  $A_0, A_1, \dots, A_n$  for the sake of mathematical representation only (they shall be enumerated by IP-addresses). The number of agents  $n$  is variable as there will be constantly new information incoming. Local logical controllers  $L_0, L_1, \dots, L_n$  will be attached to the agents. These controllers are local logical subsystems, forming the global logical model. Our global fibering follows  $\xi=(E,\pi,I,L)$  being a free parallel system  $L^n$  with the local truth value sets  $\Omega_i = \{T_i, F_i\}, i \in I = \{0,1,\dots,n\}$

and the global truth value set  $\Omega^n = \{T_0, F_0, T_1, F_1, \dots, T_n, F_n\} = \prod_{i \in I} \Omega_i$ . Beside the logical controllers, we have local fiberings

$\xi_0, \xi_1, \dots, \xi_n, i \in I = \{0,1,\dots,n\}$  and  $\xi_n=(E_n, \pi_n, I_n, L_n)$  attached to each agent, thus producing a fibered fibering. Local truth values are  $\Omega^{nm} = \{T_{n0}, F_{n0}, T_{n1}, F_{n1}, \dots, T_{nm}, F_{nm}\} = \prod_{i \in I_n} \Omega_{in}$ .

The “communication” (reasoning) between the agents and between their elements is performed by transjunctions. Whereas the intelligent plugin establishes the underlying principles, the representation follows:

$$\Theta : L_i \times L_i \rightarrow L_0 \prod L_1 \dots \prod L_n, i \in I = \{0,1,\dots,n\}$$

as well as

$$\Theta : L_{ni} \times L_{ni} \rightarrow L_{n0} \prod L_{n1} \dots \prod L_{nm}, i \in I = \{0,1,\dots,m\}$$

$\Theta$  defines a truth table, the values of which will be defined during program execution in order to adjust correctly to the changing situations.

## C. Data Mining Plugin

In order to help with the finding of conclusions a Data Mining plugin is used. It complements the Logical Fibering algorithm by offering a means of comparison and making the results more accessible to human analysis.

To achieve this aim, firstly a set of cases has to be found. These cases typically consist of attributes (like source IP, destination-IP, port, connection time etc.) and consequences (e.g. bandwidth, CPU load, open connections).

In a second step we make up a set of rules from the raw data by observing repetitions and cutting attributes, which are not relevant for the specific case, thus reaching simple rules like the following:

```
if (sourceIP==10.138.20.41) and (openConnections==20) then
    problem(totalOpenConnections)

if (sourceIP==10.135.10.31) and (port==8080 || port==8066) then
    problem(bandwidth)
```

More detail on classification rules may be obtained at [38].

## D. Database

The database is the way to save the data obtained. It was chosen on the one hand because of the open domain of data, which makes it very difficult to work all of this in program memory and on the other because the data can be separated from the program and analyzed apart by “add-on” utilities.

Nonetheless, a form of data cache is implemented in order to model the Logical Fibering structure and speed up processing for the most probable cases.

In the database, the following simple tables are maintained as a first step:

Table: GLOBAL (current global fibering)

1. ID Long int (Index)
2. Enumerator String[30] (IP-address)
3. Fiber Boolean (block now?)

Table: LOCAL (current local fibering)

1. ID Long int (Index)
2. ID\_Global Long int (Foreign key of GLOBAL)
3. Enumerator String[80] (Data)
4. Fiber Boolean (Operation logic)

Table: HISTORY (events inside the fibers)

1. Date Date (Date of event)
2. Time Time (Time of event)
3. ID\_Type Long int (Foreign key of TYPE)
4. ID\_Global Long int (Foreign key of GLOBAL)
5. ID\_Local Long int (Foreign key of LOCAL)
6. Value Float (Current value of the variable analyzed)
7. Remarks String[255] (Any necessary remarks)

Table: TYPE (type of events)

1. ID\_Type Long int (INDEX)
2. Description String[255] (Type description)

It is likely that this structure will still be amplified according to further development necessities.

## VII. MAIN ALGORITHM

The purpose of the algorithm is to offer web access management. Network connections are analyzed and either accepted or rejected by the machine the system is running on, depending on the result of the analysis. Obviously, this decision is provided by the system logic, which is the heart of the concrete development.

As Logical Fiberings proposes a two (or many) valued logical structure for all information, we must find a binary way of representation for all available data, i.e. we for example store the IP-address serializing it like this: 10.135.10.2 = 101010000111101010 (being 1=*true* and 0=*false*).

On the whole, the way of storage depends on how we wish to access the information (e.g. an IP-address is best serialized, as we can thus examine similarities between addresses - this surely does not apply to all data).

Every new connection is first stored in the fiberings, i.e. an agent is created, which shall take care of the specific configuration. Each agent looks on its own characteristics, which are a section of the total fiber space and shall also be able to take overall characteristics into account (like the system's health state).

Nothing is pre-defined and the system does not block anything in advance, which it does not know. Over time it accumulates experience on the different forms of attack.

Attacks are identified by the noxious consequences they have on the system. Thus we concentrate on anything, which actually involves an effect, i.e., not working attacks are automatically discarded from analysis.

In general, three modes of system execution are given:

- Normal operation: No attacks are detected and thus no analysis is made. This way of processing makes the system a "light weight" tool in terms of resource usage.
- Slight problem operation: A potential attack situation is detected through scarce resources. In case the Logical Fiberings is able to decipher the attack, measures will be taken accordingly to eliminate the problem. If the fiberings does not arrive at a conclusion, the Data Mining plugin is invoked and analysis is driven until a result is reached or a severe problem situation occurs.
- Severe problem operation: "Fibered Guard" works to not let severe problems happen. However, this surely cannot be guaranteed as attacks vary a lot (just like the victim's resources also do). This way, in severe problem operation, the system frees space by brute force in order to be able to continue alive and manage to still catch the most data possible for analysis.

Problem analysis involves firstly a firm definition of what actually is a problem. This definition is not opened to any different conclusions from the system itself, actually being "hard-coded".

As mentioned above, everywhere, where we have already laid the trails to "faster" fiberings logic, we will use these trails

first and only if they do not work, we will make adjustments. In case we do not have any logical trails laid yet, we will employ Data Mining to help lay them.

Basically, a method that works, is a method which mitigates the attack notably. We will work our list of priorities from above:

- Experience (inside Logical Fiberings)
- Intelligent Conclusions (Data Mining combined with Logical Fibers)
- Try and Error (nothing works??)

It must be observed that a problem might actually arise because of a lot of random factors (HD space missing because of huge downloads, memory limited because of a lot of open applications, several downloads open and trouble with bandwidth etc.). These factors must be eliminated: or beforehand actually not identifying them as problem, or afterwards when it comes to not concluding that a problem situation exists.

"Fibered Guard" follows a principle of "smooth processing". This means that no radical decisions upon connections must be taken as long as there is no reason sound enough for that. Otherwise the program would not serve for prevention, but actually creation of DoS (by dropping potentially legal connections!).

## VIII. EXAMPLES OF PROCESSING

### A. SYN-Attack

A SYN-attack is, by itself, a non distributed denial of service attack. It may thus be handled only by means of the local fiberings, i.e., not involving global analysis.

Supposing that the incoming information was not obtained yet, a new local fiberings is created, the obtained data is stored and processing simply proceeds at the first point.

Soon the number of open connections rises, which creates a bad effect on bandwidth, memory etc. The system detects these problems and analyses the cause. It will discover an abnormality in the newly created fiber by the great number of open connections. The fiber will receive a logical structure which deduces from this great number of open connections, that its function must be blocked (fully or partially) [34].

All connections corresponding to this fiber are then dropped.

Taking the case now that a new connection attempt is made from the same address: The fiber at once is "suspicious", though, the blocking condition has not necessarily been met. Therefore, only after showing a tendency of going into the same problem direction (a great number of open connections), an immediate drop will be the cause. This mechanism takes the widely spread problem of IP faking into account, allowing to a normal extent connections from a source, which has already been marked as suspicious in order not to prevent users, who legally use this address, to access resources.

### B. DDoS Attack

DDoS cannot be treated on a local basis, but need global action. This is true, because the main characteristic of DDoS is the fact that similar access attempts from a whole range of different machines, which are misused as agents by a central

instance, are made (see section II). Single attack methods may vary greatly and thus have to be unimportant for the analysis, i.e. the system shall only consider the global attack architecture.

Just as in the first case, the starting point is the actual problem condition, i.e. a system resource becomes scarce. In this situation the system shall go through the fiber logic of the local fibers and not finding any extraordinary situation, it shall switch to global analysis.

Firstly, it will note a reasonable amount of similar (and probably faulty) connections. Secondly, analyzing history, it will find out that the connections were all established over a short period of time. These connections shall be identified and the system will start randomly dropping some of them to free system resources.

## IX. CONCLUSIONS

The system “Fiber Guard” is a hybrid web access management solution for the treatment of DoS and DDoS, which makes use of a Logical Fiber algorithm with a Data Mining plugin to treat attacks on a global and local scope level. Global treatment is required to handle distributed attacks, which have the main characteristic of connect attempts from several sources whereas for simple DoS, which is not amplified through distribution, the thorough analysis of a single type of connection is executed.

Through this flexible behavior and the fact that fiberings may be infinitely enumerated, “Fibered Guard” offers a framework, which has the possibility to substitute all current defense approaches.

## X. REFERENCES

- [1] Answers.com – Fast Facts: fiber bundle, <http://www.answers.com/topic/fiber-bundle>, 2005.
- [2] Baker, B.S., Grosse, E.: Local Control over Filtered WWW Access. In: Fourth International World Wide Web Conference, Boston/USA, 1995.
- [3] Baran, P.: On Distributed Communications. Memorandum RM-3420-PR <http://www.rand.org/publications/RM/RM3420/1964>, 1964.
- [4] Bivens, A., Palagiri, C., Smith, R., Szymanski, B., Embrechts, M.: Network Based Intrusion Detection using Neural Networks. Rensselaer Polytechnic Institute, New York, 2002.
- [5] Branch, J., Bivens, A., Chan, C-Y., Lee, T-K., Szymanski, B.K.: Denial of Service Intrusion Detection Using Time Dependent Deterministic Finite Automata. In: Proc. Graduate Research Conference, Troy, NY, 2002, pp. 45-51
- [6] Burch, H., Cheswick, B.: Tracing Anonymous Packets to Their Approximate Sources, In: 14<sup>th</sup> Systems Administration Conference, LISA2000), New Orleans/USA, 2000.
- [7] Computer Associates International Inc.: Security Management – eTrust Web Access Control, <http://www3.ca.com/Solutions/Product.asp?ID=3224>, 2005.
- [8] Cert Coordination Center: Denial of Service Attacks, [http://www.cert.org/tech\\_tips/denial\\_of\\_service.html](http://www.cert.org/tech_tips/denial_of_service.html), 2001.
- [9] Cheswick, W.R., Bellovin, S.M., Rubin, A.D.: Firewalls e Segurança na Internet – Repelindo o hacker ardidoso. 2<sup>nd</sup> ed., Porto Alegre, 2005.
- [10] Cyberoam: Why Internet Access Management? <http://www.cyberoam.com/>, 2005.
- [11] Demers, A., Keshav, S., Shenker, S.: Analysis and simulation of a fair queuing algorithm. In: Internetworking: Research and Experience, vol.1, no.1, 1990, pp. 3-26
- [12] DFN-CERT: Distributed Denial of Service Angriffe, <http://www.cert.dfn.de/infoserv/dib/dib-2000-01.html>, 2001.
- [13] Ferguson, P., Senie, D.: Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. <http://rfc2267.x42.com/>, Natick/USA, 1998.
- [14] Floyd, S., Jacobson, V.: Random Early Detection gateways for Congestion Avoidance. In: IEEE/ACM Transactions on Networking, V.1 N.4, 1993, pp. 397-413
- [15] Hussain, A., Heidemann, J., Papadopoulos, C.: A Framework for Classifying Denial of Service Attacks. ISI Technical Report 2003-569, 2003.
- [16] IBM Inc.: IBM Tivoli Access Manager for e-business. <http://www-306.ibm.com/software/tivoli/products/access-mgr-e-bus/>, 2005.
- [17] Jansen, W., Mell, P., Karygiannis, T., Marks, D.: Mobile Agents in Intrusion Detection and Response. In: Proceedings of the 12<sup>th</sup> Annual Canadian Information Technology Security Symposium, Ottawa/Canada, 2000.
- [18] Lau, F., Rubin, S.H., Smith, M.H., Trajovic, L.: Distributed denial of service attacks. In: IEEE International Conference on Systems, Man, and Cybernetics, pp. 2275-2280, Nashville/USA, 2000.
- [19] Lee, W., Stolfo, S.J.: Data mining approaches for intrusion detection. In: Proceedings of the 7th USENIX Security Symposium, 1998.
- [20] Martin, R.: Survey Indicates Identity and Access Management a Chief Security Concern, in: Entprise Networks & Servers, Austin/USA, 2004.
- [21] McClure, S., Scambray, J., Kurtz, G.: Hackers Expostos – Segredos e Soluções para a Segurança de Redes. 4th ed., Editora Campus: Rio de Janeiro/Brasil, 2003.
- [22] Mirkovic, J., Reier, P.: A Taxonomy of DDoS Attack and DDoS Defense Mechanisms. ACM, 2004.
- [23] Oblix Inc.: Solutions for Web Access Management and Single Sign On – User Access Management for Secure Business Interactions. [http://www.oblix.com/solutions/access\\_management/](http://www.oblix.com/solutions/access_management/), 2005.
- [24] Perlegos, P.: DoS Defense in Structured Peer-to-Peer Network. Technical Report UCB/CSD-04-1309, University of Berkeley, Berkeley/USA, 2004.
- [25] Pfalzgraf, J.: The concept of Logical Fiberings and fibered logical controllers. In: Proceedings Computing Anticipatory Systems: CASYS 2000. Liège, Belgium. AIP Conference Proceedings, Vol. 573, 2001, pp. 683-693
- [26] Pfalzgraf, J.: On Logical Fiberings and Automated Deduction in Many-valued Logics Using Gröbner Bases. Revista Real Academia de Ciencias, Serie A de Matemáticas, RACSAM, Vol. 98 (1), 2004.
- [27] Pfalzgraf, J., Edtmayr, J.: The concept of Logical Fiberings: distributed logics for multiagent systems. In: Proceedings 17th European Meeting on Cybernetics and Systems Research (EMCSR'2004) Vienna, 2004.
- [28] Pfalzgraf, J., Meixl, W.: A logical approach to model concurrency in multiagent systems. In: Proceedings 15th European Meeting on Cybernetics and Systems Research, EMCSR'2000), Vienna, 2000.
- [29] Pfalzgraf, J., Sigmund, U., Stokkermans, K.: Towards a general approach for modeling actions and change in cooperating agents scenarios. In: Special Issue of IGPL (Journal of the Interest Group in Pure and Applied Logics), IGPL 4 (3), 1996, pp. 445-472
- [30] RSA Security Inc.: Web Access Management. <http://www.rsasecurity.com/node.asp?id=1185>, 2005.
- [31] Savage, S., Wetherall, D., Karlin, A., Anderson, T.: Practical network support for IP traceback. In: Proceedings of the 2000 ACM SIGCOMM Conference, Uppsala/Sweden, 2000.
- [32] SANS Institute: Global Incident Analysis Center – Special Notice – Egress Filtering v 0.2. <http://www.sans.org/y2k/egress.htm>, Maryland/USA, 2000.
- [33] Schmidt, J.: Dämme gegen die SYN-Flut, <http://www.heise.de/security/artikel/43066>, 2005.
- [34] Schneider, M.O., Calmet J.: Denial of Service Prevention through Logical Fiberings. In: Proceedings of the IIAS 05, Baden Baden/Germany, 2005.
- [35] Ulbrich, H.C., Della Valle, J.: Universidade Hacker – Desvende todos os segredos do submundo dos hackers. 2<sup>nd</sup> ed., Digerati: Sao Paulo/Brasil, 2003.
- [36] Wack, J., Cutler, K., Pole, J.: Guidelines on Firewalls and Firewall Policy. Recommendations of the National Institute of Standards and Technology (NIST), Gaithersburg/USA, 2002.
- [37] Wiltenburg, E.v.: Routers & Firewalls. Lecture at the University of Victoria, British Columbia/Canada, 2004.
- [38] Witten, I.H., Frank, E.: Data Mining – Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann Publishers, San Francisco San Diego New York Boston London Sydney Tokyo, 2000.