Generic Denial of Service Prevention through a Logical Fibering Algorithm

Marvin Oliver Schneider and Jacques Calmet
Institute for Algorithms and Cognitive Systems (IAKS)
University of Karlsruhe
Am Fasanengarten 5, 76131 Karlsruhe
GERMANY

Abstract: - This paper presents the system FiberedGuard, which is an intelligent application for the prevention of Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. The main basis of FiberedGuard is a new data structure, which was inspired by Logical Fibering. Its advantages are flexibility, use of logical structures, interconnected global and local processing as well as simplicity to human interpretation. The alpha release of the system is currently being finalized. Results are obtained firstly in a separate laboratory structure until full approval. The system shall then be transported to the real world. The approach, which is novel, has the potential of revolutionizing the treatment of DoS and DDoS, substantially contributing to research in the area of security.

Key-Words: - Denial of Service Prevention, Logical Fibering, Internet Security

1 Introduction

Denial of Service (DoS) and Distributed Denial of Service attacks are serious security issues to any organization, which does commerce over the Internet. Infamous examples of well-known sites being brought down by simple means are Yahoo, Ebay and Amazon.com [12]. DoS attacks are generally not expected and not at all treated, but instead "worked" by a catalogue of general measures and conservative system policies, an example of which may be found in [5]. The necessity of a reform of this field is apparent and the present work shall suggest a much more appropriate way of dealing with this pressing security problem.

2 Denial of Service

An abstract definition of Denial of Service is the basis of the present approach. An attack may be described as a successful method which produces a situation in which a victim computer is not able any more to respond adequately inside a network structure. FiberedGuard concentrates on attacks, which may be treated by means of software, i.e., though strictly speaking physical attacks may be also seen as DoS-attacks, they are explicitly not treated. By principle, any other form of attack is handled. For all characteristics yet unknown, FiberedGuard's modular code offers a means of easy expansion.

2.1 Importance

Denial of Service attacks (including the special case of Distributed Denial of Service attacks) are of high economic impact. They may not only bring down e-Commerce sites easily and thus effectively prevent certain forms of commercial activity, but also inflict damage to the image of organizations, by shutting their sites down, making them inoperable and bringing about serious doubts towards their data security.

DoS attacks are powerful, because they are easy on the attackers' side and cannot be prevented in a simple way on the victim's side as sorting network traffic can be truly difficult, especially for a human administrator, who cannot follow the speed of all happenings or who has highly subjective opinions on the necessary measures. Simple defense approaches, furthermore, also have proven to lack the necessary abstraction or to be over simplistic. The need for an intelligent, flexible and generic way to treat the problem is obvious.

2.2 Common Characteristics

Denial In order to successfully treat DoS on a generic level, it was necessary to identify common characteristics for all DoS attacks. This has proven to be not at all trivial as attacks vary a lot. An example of a framework for classification is given in [9]. As a

basic overview the following groups may be distinguished.

Attacks based on a system "bug": The present approach classifies a "bug" as being something obviously mistaken, even for normal operation. In this group, e.g., DNS-attacks may be found, which pass corrupted data to the DNS-server to force it to put wrong information in cache or also the Linux "Teardrop" attack, which works on wrong package division [20]. DoS-attacks based on bugs are omnipresent and as some get fixed by simple bugfixes, new system versions bring new attack doors.

Attacks based on normal operation and "brute force": Instead of using the "back door" of system "bugs", many attacks may simple take the "front door" and use publicly available resources, which may be freely accessed over the network. Trying to trace these attacks back to system errors typically reveals heavy problems in architecture which by no means can be treated through "bugfixes" or any other straightforward method. A simple and infamous example is the so called "SYN-Flood", which uses an architectural error of the TCP/IP-protocol. As further examples "Smurf", "Fraggle" and "UDP-Flood" could be mentioned [11].

Attacks based on attack tools: Even though this might be seen as a simple special case of the previous two classifications, we consider it as a separate case. This is true, because these attacks pose a considerably higher difficulty on any defense tool as they typically work in a distributed way, hiding the attacker, creating many different forms of attacks at the same time, facilitating control by shells, automatic updates, encryption etc. Examples of such tools are: "Tribe Flood Network (TFN)", "TFN2K", "Trinoo", "WinTrinoo" and "Stacheldraht" [11][20]. The term of Distributed Denial of Service (DDoS) attacks is tightly linked to these mechanisms.

Summarizing, it may be said, that there are no common characteristics that could identify any attack by its mechanisms. What really characterizes a DoS attack is the fact that it works upon scarce system resources, such as memory, open connections and bandwidth. This is the only guiding line a generic approach can follow.

3 Related Work

Apart from the frequently recommended practical measures, several scientific defense approaches have been proposed. The following shall give a brief overview and mention the main weaknesses of each type for discussion.

Ingress/Egress Filtering [7] [16] is a classic approach to the treatment of Denial of Service attacks. It relies on the fact that any legal request from a specific domain must provide an IP-address from that domain – any other IP-address would most probably be false and thus characterize a DoS attack. Though of striking simplicity and surely providing a certain effect, it may be criticized that firstly continuous filtering needs additional resources. Furthermore this attack actually does not eliminate IP-spoofing, but merely reduces it to a specific domain.

Tracking algorithms were introduced with the merit of actually finding the source of the problem [4] [17]. This is done, e.g., by the use of an overlay network, packet marking at routers or short IP-bursts and the observation of their consequences. Surely, it is indeed interesting to find the problem's origin. However, this in itself does not characterize a valid defense against the attack, as further action (legally speaking or by the use of blocking algorithms) is required. Secondly, depending on the approach, unambiguous identification of the source is yet difficult to achieve or high implementation efforts are required.

Another way of "treating" DoS are the so-called congestion control algorithms. Examples would be Fair Queuing [6], Random Early Detection [8], Differentiated Services [2] or On-Off Feedback Control [21]. All of these algorithms establish rules to strictly limit the use of bandwidth and effectively prevent overload situations. Although interesting in the sense of keeping the victim "alive" in stress situations, these tools may be seen, strictly speaking, more as creators of DoS than defenders, since they potentially block legal traffic and DoS traffic alike.

Definitely, the problem of DoS urges for intelligent action. There is actually a number of intelligent approaches, such as the Datamining Approach to Intrusion Detection [10], Automata [3] and Artificial Neural Networks [1]. The key to their effectiveness are mainly the datasets provided, storage and retrieval procedures.

4 Logical Fibering Structure

FiberedGuard is a novel approach to Denial of Service prevention as it uses data structures in a Logical Fibering fashion. The principle of Logical Fibering was developed by J.Pfalzgraf [13] and was inspired in Polycontextural Logic (PCL) [15] and the concept of fiber bundles [14]. Basically, a Logical Fibering is a fiber topology with logical structures. In a typical, fine grained case, these structures may be classical two-valued logics (*true* and *false*). More

coarse-grained structures as used in FiberedGuard may also be employed where convenient. These coarse-grained structures may as well be seen as what is called "fibered fibering", which was not adopted as term here, in order to put more emphasis on flexible logical boundaries.

Mathematically speaking a Logical Fibering $\xi = (E,\pi,B,F)$ consists of a base space B with the indices $b \in \{1..b\}$, a total space E with all the logical subsystems $E_{b,\ b\in\{1..b\}}$ and a projection map $\pi : E \to B$ which links the indices to the total space. Typical fibers are denoted with F. Possibly, the fibers might have no global connection, only representing the disjoint union of several classical two-valued systems. The result of this construction is denominated a "free parallel system" and is a basic structure of Logical Fibering [13]. In more complicated cases, however, as in FiberedGuard, there may be global interconnections and more sophisticated local logical structures.

4.1 Justification

Logical Fibering is a flexible way of representing situations with local characteristics, which all potentially contribute to a global picture and which have logical connections. This is a good way to absorb the complex attack situation under the suspect of Denial of Service. All relevant information may be stored, logical structures may be mounted inside the fibering and retrieval is straight-forward. This also offers the necessary speed to deal with potential intruders.

4.2 Representation

Basically, FiberedGuard sees every fiber as a logical structure, which carries all relevant data for one connection from a specific IP address. Surely, IP data may be forged and cannot be relied on by any means. However, this is not really relevant to FiberedGuard, as the Logical Fibering only uses this information as most natural enumerator for connections

As shown in [18] and [19], a range of DoS attacks from one specific machine might be best treated by local logical structures, i.e., inside the logic of every fiber. The more powerful DDoS attacks, however, urge for a global analysis, which is already represented by the fibering architecture itself.

FiberedGuard uses IP-addresses as enumerators in base space B because IP-addresses may be considered specially formatted numbers and are a more natural and practical way of enumerating connections. For every fiber of the total space E_b, a specific agent A_b is created (see fig. 1). This agent is responsible for the fiber's data and local logical structure, which is an interconnection of its values by basic logical operators. The resulting formula defines the conditions under which the local fiber shall be seen as a "Denial-of-Service fiber", i.e., if the term evaluates to "*true*", a DoS-situation is identified and the corresponding connection is dropped. The end of every logical structure is defined by an end-sign rather than a fixed length. Thus structures of variable complexity may be mounted by the storage algorithm in order to ideally adjust to a present threat.

Global threats (in most cases DDoS) are defined by logical interlinks in between the established fibers. These links form logical structures, just like the local ones, combining characteristics of different fibers. The structures may also vary in length and the resulting terms define the logical conditions for a detected attack. In this case all connections corresponding to the fibers, which are involved in the structure, may be dropped as consequence. Thus, a large amount of noxious connections can be blocked at once, which is the correct way of treating DDoS attack amplification.

4.3 Storage

As soon as an attack situation is given (compare 5.1) and the existing information inside the fibering does not attend the case in a satisfactory way (i.e. by eliminating the overload situation) firstly a snapshot of the whole situation is stored in memory, i.e., for every single connection an agent is created and the corresponding characteristics are stored, interconnected by simple "and" operators. In a second step the system starts dropping connections randomly while monitoring the overall system health state. This needs to be done, as the only trustworthy way of detecting DoS is by its effect (see 2.2). In case there is a significant "jump" (threshold values are configurable) with an apparent instant problem solution at one point, the system makes the inverse test, re-allowing the connection in. In case the system health state worsens at once, the respective local fiber is marked active and the rest on stack of the snapshot (already examined, or yet to be examined) is erased from memory. In case there is no new connection or the system's state does not worsen after re-allowance, this fiber is kept and marked as "suspicious". It will be the first to be examined in the case of a next snapshot. The other fiber information of the snapshot is erased anyway, if the health state got better. This goes in line with the principles of smooth processing and not presenting unnecessary harm to legal connections.

4.4 Fiber Management

As a matter of fact the "raw" fibering data obtained by snapshots should be optimized and administrated. Firstly simplification rules on logical structures apply. Any different distinct local fibers for one connection are first linked with simple "*or*" conditions and afterwards worked on by standard optimization algorithms in order to form a single compact condition.

Secondly, the data is open to human intervention. This has mainly two reasons: On the one hand the system may be storing incorrect information. In this case a human intervention is necessary to correct the error. Thus, the system is maintained open to any manual changes (creation, erase, marking etc.) necessary. As a human operator uses to be responsible and in charge for a computer, this step is furthermore consequent already from the view point of validation. On the other hand it may also be considered that a human operator would like to gain insight into the data to run statistics or take further action against the DoS source. Thus data should be constantly open for analysis. The logical structures when optimized are of relatively simple reading to any human operator with basic logical knowledge.

5 Implementation

FiberedGuard is a system, which was implemented using J2SE for portability reasons and a mySQL database for query speed. Furthermore, both systems were chosen in order to offer a free-of-charge implementation with the possibility of distribution and scientific exchange.

5.1 System Modes

The system makes use of three system modes in order to adjust its operation to the changing environment:

Normal operation: Only the system's health state is being monitored and if everything works nicely, no analysis or further processing are done. This is crucial as the system's purpose is to prevent DoS from happening and not to provide further pressure on the system, which might make it easier for attackers to get through. Furthermore, though potential attacks may have been launched even in normal operation, only well succeeded attacks matter to FiberedGuard. This is especially true as the line between a weak attack and a normal access is extremely smooth and even more difficult to distinguish, thus easily leading to wrong operation, if treated.

Analysis mode: Operation under DoS attack with the possibility to analyze according to the mechanism mentioned in section 4. This operation makes use of tasks that still demand some computing power extra, i.e., under a serious attack, it may be useful to free resources first and analyze afterwards.

Problem mode: If there is a real serious problem and the victim is close to collapsing, the first thing which has to be taken care of, is to make the victim survive. Therefore this last mode is a simple mechanism, which drops connections and frees space by brute force and without any logical analysis. As analysis relies on available system resources, this is also done exactly for the sake of further possible analysis.

5.2 Database

It is expected that in real life scenarios on servers the received data will be huge and impossible to keep all in memory. Therefore the first version works – apart from snapshot data as described in section 4 and a small cache for very frequently used fibers – directly over the database. This is the only way the principle of infinite enumeration, which is one of the Logical Fibering benefits, can be reasonably treated. It may be recognized that database activity can become slow with a high number of entries. However, a total explosion of online data would have yet more disastrous effects.

The database stores the local and global fiber structures. Other data is treated directly in memory as processing is faster and there is no need for the reservation of large spaces. At the present moment the FiberGuard's database possesses four tables:

"Characteristics": This table is used to map a text to all the known characteristics of the system. Characteristics are, e.g. the first Boolean number of the destination IP, the second Boolean number of the time of access etc. It is important to note that all non-Boolean characteristics are coded to Boolean and that even inside one characteristic logical connections exist.

"States": Provides the description of the specific states a fiber may adopt, as "snapshot", "active", "suspicious", "excluded", "cache" etc.

"Local_Fibering": Stores the local fibering data, such as the enumerator (IP-address in letters), position in the specific logical chain, current characteristic and state (states may even vary inside a single fiber), value and the next operation, which is a string of logical operators and possibly brackets.

"Global_Fibering": Stores the global fibering data. The characteristic is chosen from a local fibering position whereas new logical operators may apply in the global structure.

4 Conclusion

This paper presents an overview of the system FiberedGuard, an intelligent solution for defense against DoS and DDoS attacks on the basis of a flexible Logical Fibering data structure. Attack situations are saved in the fibering, the obtained data is analyzed and logical connections are mounted. As a special characteristic, the necessity of local or global action (depending on the attack scheme) is considered. Thus, one of the main problems which many approaches present in defending only against specific attacks is successfully solved.

FiberedGuard is a promising method of adapting adequately to the confusing and noisy situation of DoS-attacks and has the potential of revolutionizing the treatment of DoS and DDoS.

References:

[1] Bivens, A., Palagiri, C., Smith, R., Szymanski, B., Embrechts, M.: Network Based Intrusion Detection using Neural Networks. Rensselear Politechnic Institute, New York, 2002

[2] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W.: An Archictecture of Differentiated Services, Network Working Group, Request for Comments 2475, http://asg.web.cmu.edu/rfc/rfc2475.html, 1998

[3] Branch, J., Bivens, A., Chan, C-Y., Lee, T-K., Szymanski, B.K.: Denial of Service Intrusion Detection Using Time Dependent Deterministic Finite Automata. *Proc. Graduate Research Conference*, Troy, NY, 2002, pp. 45-51

[4] Burch, H., Cheswick, B.: Tracing Anonymous Packets to Their Approximate Sources, *14th Systems Administration Conference*, LISA2000), New Orleans/USA, 2000

[5] Cert Coordination Center: Denial of Service Attacks, http://www.cert.org/tech_tips/denial_of_service.html, 2001

[6] 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

[7] 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

[8] Floyd, S., Jacobson, V.: Random Early Detection gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, V.1 N.4, 1993, pp. 397-413

[9] Hussain, A., Heidemann, J., Papadopoulos, C.: A Framework for Classifying Denial of Service Attacks. *ISI Technical Report*, 2003

[10] Lee, W., Stolfo, S.J.: Data mining approaches for intrusion detection. *Proceedings of the 7th USENIX Security Symposium*, 1998

[11] McClure, S., Scambray, J., Kurtz, G.: *Hackers Exposed – Network Security Secrets & Solutions*, McGraw-Hill, 2005

[12] Mirkovic, J., Dietrich, S., Dittrich, D., Reiher, P.: *Internet Denial of Service: Attack and Defense Mechanisms*, Prentice Hall, Upper Saddle River, NJ, 2005

[13] Pfalzgraf, J.: The Concept of Logical Fiberings and Fibered Logical Controllers. *Proceedings Computing Anticipatory Systems: CASYS 2000*. Liège, Belgium. AIP Conference Proceedings, Vol. 573, 2001, pp. 683-693

[14] Pfalzgraf, J., Edtmayr, J.: The concept of Logical Fiberings: distributed logics for multiagent systems. *Proceedings 17th European Meeting on Cybernetics and Systems Research (EMCSR'2004)*, Vienna (2004)

[15] Pfalzgraf, J.: Logical Fiberings and Polycontextural Systems, *Fundamentals of Artificial Intelligence Research*, International Workshop FAIR '91, Smolenice, Czechoslovakia, 1991

[16] SANS Institute: Global Incident Analysis Center – Special Notice – Egress Filering v 0.2. http://www.sans.org/y2k/egress.htm, Maryland/USA, 2000

[17] Savage, S., Wetherall, D., Karlin, A., Anderson, T.: Practical network support for IP traceback. *Proceedings of the 2000 ACM SIGCOMM Conference*, Uppsala/Sweden, 2000

[18] Schneider, M.O., Calmet J.: Denial of Service Prevention through Logical Fibering. *Proceedings of the IIAS 05*, Baden Baden/Germany, 2005

[19] Schneider, M. and Calmet, J.: FiberedGuard – A Hybrid Intelligent Approach to Denial of Service Prevention, *Proceedings of the IAWTIC'05*, Vienna, Austria, 2005

[20] Ulbrich, H. C., Della Valle, J.: *Universidade Hacker – Desvende todos os segredos dos submundos dos hackers*. 2ª. Ed., Digerati, São Paulo, Brazil, 2003

[21] Xiong, Y., Liu, S., Sun, P.: On the defense of the Distributed Denial of Service Attacks: An On-Off Feedback Control Approach, *IEEE Transactions on System, Man and Cybernetics-Part A: Systems and Humans*. Vol. 31 No.4, 2001