Mobile-Agent-Based Network Cooperative Security Architecture against Distributed Denial-of-Service Attacks[†]

Li-Der Chou[⋆] and Shyh-Luan Lee

Department of Computer Science and Information Engineering
National Central University
Chungli, Taoyuan, Taiwan, 32054 R.O.C
TEL: +886-3-422-7151 ext 4521,
FAX: +886-3-422-2681
Email: *cld@csie.ncu.edu.tw*

Abstract

The objective of the Distributed Denial-of-Service (DDoS) attacks is to paralyze the server and the provided services. Attackers usually intrude a group of hosts, and organize these hosts into a hierarchy. The intruded hosts located in the leaves of the hierarchy may be commanded to simultaneously send a large amount of attacking packets to the victim to interrupt or stop its services. It is very difficult to pursue the real attacker. A network cooperative security architecture using the mobile agent technology is proposed in the paper, to reduce the impact of the DDoS attacks on the

[†] The manuscript is submitted to *Workshop on Artificial Intelligence*, *2002 International Computer Symposium*.

[⋆] The contact author

services provided by the victim. As long as possible DDoS attacks are detected, the server on the victim will be replicated and moved to other network nodes that support the execution environment of mobile agents. Thus the service still survives. The paper also proposes two types of mobile agents, the monitoring agent and the server agent, to cooperate in the defense of the DDoS attacks.

Keywords: distributed denial-of-service attacks, mobile agents, security architecture

I. Introduction

Many new network services and applications have been developed and provided, such as e-commerce, e-learning and e-banking. For the services are getting united with electronic commerce, the network security is getting improtant [1]. The goal of the Denial-of-Service (DoS) attacks is to paralyze the victim server and the provided services, by exhausting the server's resources, such as memory, processes and disk spaces. Because the attacker does not need to intrude the server, it is not easy to pursue the attacker. There are many kinds of DoS attacks, such as TCP SYN flooding attacks [2] [3]. Moreover, combining with the technology of distributed system, the distributed DoS (DDoS) attacks [4] [5], shown in Fig. 1 have been developed The attackers first

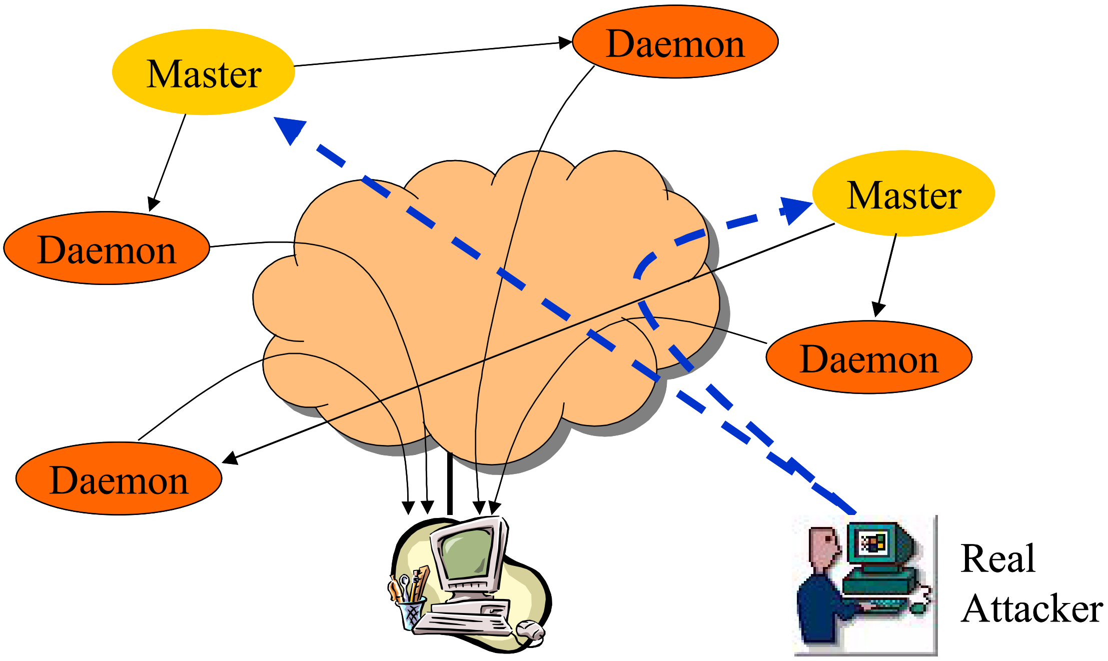


Fig. 1 Concept of the DDoS attacks.

intrude several hosts, and organize them into a hierarchy. The intruded hosts located in the leaves of the hierarchy may be commanded to simultaneously send a large amount of attacking packets to the victim server to intercept or stop the provided services. Up to date, there are many kinds of DDoS attacks, such as trinoo [6], tribe [7], stacheldraht [8], mstream [9], and shaft [10]. Pursuit of the real attacker for the DDoS attacks is much difficult than that of DoS attacks. Most of the current solutions for DoS/DDoS attacks are to adjust the parameters, such as timeout and buffer size, of the server to lessen the impacts. Recently, the victim can analyze the traffic from the reflectors so as to identify and filter out the attack traffic [11].

For the mobile agents have the characteristics of autonomy, social ability, reactivity, pro-activity, mobility, and veracity [12], and thus are appropriate to be applied to network management [13] and security management [14]. Mobile agents with specific functions can be dispatched to the specific network nodes, and can accomplish the assigned security management works as requested. Moreover, the distributed architecture for mobile agents provides an efficient way to update the security management policies. For some complicated security management functions, such as the detection and the defeat of the DDoS attacks may not be accomplished by a single agent, multiple agents can be adopted at a time to cooperate with each other.

In the paper, the network cooperative security architecture is proposed, using the mobile agent technology, to defeat the DDoS attacks and guarantee the sustainment of the services. Two types of mobile agents are designed: the monitoring agent and the server agent, where the server agent is the program of a specific server. As the monitoring agent detects that the server suffers the DDoS attacks, the server agent will be replicated and be dispatched to other node through the most congested link. Thus the service provided by the server agent will not be intercepted, and is survivable. Of course, each network node has to support the agent execution environment (AEE), or so-called mobile agent system (MAS).

II. Assumptions

Assumptions of the proposed network cooperative security architecture are first described as follows.

1. Assume that all of the network nodes have AEE. The assumption can be easily achieved by connecting a host with AEE support to each network node.
2. Assume that authentication and encryption mechanisms between the network nodes and their associated hosts exist, so that the host can be trusted.
3. Assume mobile agents must be authenticated and encrypted during transmissions to avoid the corruption and the fake of agents.
4. Assume each link preserves appropriate amount of bandwidth for the delivery of mobile agents. Therefore, the transmissions of mobile agents and control messages will not be congested, and will not be affected by the DDoS attacks.
5. Assume each network node has the ability of traffic measuring and traffic filtering.

III. PROPOSED NETWORK COOPERATIVE SECURITY ARCHITECTURE USING MOBILE AGENTS

For the mobile agent technology is the merit of the proposal security architecture, the section first describes the designed mobile agents, and the cooperative operation procedure among the mobile agents. Finally, a scenario is given to illustrate the operational procedure of the proposal security architecture.

(A) Two types of mobile agents

Two mobile agents are designed in the proposed network cooperative security architecture: the monitoring agent and the server agent. The monitoring agent is resident in each trusted hosts to monitor the status of the associated network node and links, including the CPU load, buffer utilization and the link utilization. Initially, the monitoring agnets are dispateched from the agent warehouse to each MAS. Then, it is not necessary to move the monitoring agents. The monitoring agent is also responsible to determine that the associated node is suffering DDoS attacksor not, and then activate the defense actions. The program code of the server can be rewritten by the AEE-supported language, so that it can be treated as a server agent and be executed in the

AEE of the trusted host.

(B) Operational procedure

The decisions whether the received packets are DDoS attacking packets or the normal packets are very difficult, and the misjudgement may happen frequently. Therefore, the decisions will not be made in the paper. Instead, the degrees of link congestion and node load are adopted to indicate that the network or the server may suffer DDoS attacks. In the proposed security architecture, a monitoring agent is associated with each network nodes to measure the load of the node and the utilization of each connected link. As long as the load of the node or the utilization of a connected link exceeds a predefined threshold, it means that the resources of the node and the link are utilized too much, and the network can be treated to enter the congestion status.

As long as the monitoring agent detects that the node load or the link utilization exceed the predefined threshold, and the server agents located on the associated node will be notified by the monitoring agent. The server agents then replicate themselves and send them to other node with MAS. Among the neighboring nodes, the node whose load is the highest or whose connected link has the highest utilization is the target node that the replicated server agent will move to. That is, the server agent will be sent to the neighboring node that may be the origin of the DDoS attacks. As long as the replicated server agent is moved to the target node, the server agent on the target node then transparently intercepts all requests and provides them the required services. Therefore, these requests will not be forwarded to the original server agent. The original server agent only server the requests delivered from other neighboring nodes. It is equivalent to cut the transmission link from the target node to the original node. Thus in the network there are two server agents offer the service and share the load.

If the same server agent has executed in the target node, the replicated server agent will be dispatched to the neighboring node with the second highest load or with the second highest link utilization. The above procedure can be further applied recursively, and the server agents are replicated towards the congestion region of the network. At last the number of server agents will be steady. However, for the limitation of network resources, it is impossible to infinitely replicate the server agents. Thus the mechanism to kill the server agent must be provided. As long as the load of the node or the link utilization decrease below another threshold, and the same server agent is running on one of the neighboring nodes, the server agent will be killed. If the threshold for replicating the server agent is equal to the threshold for killing the server

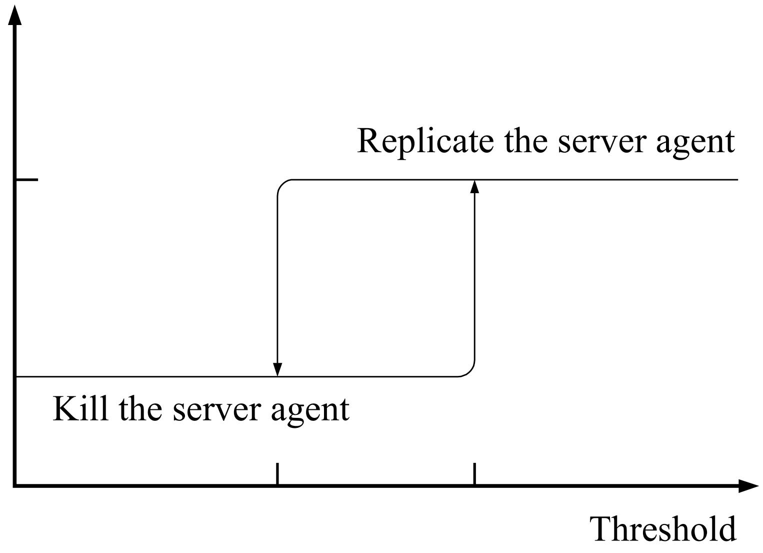
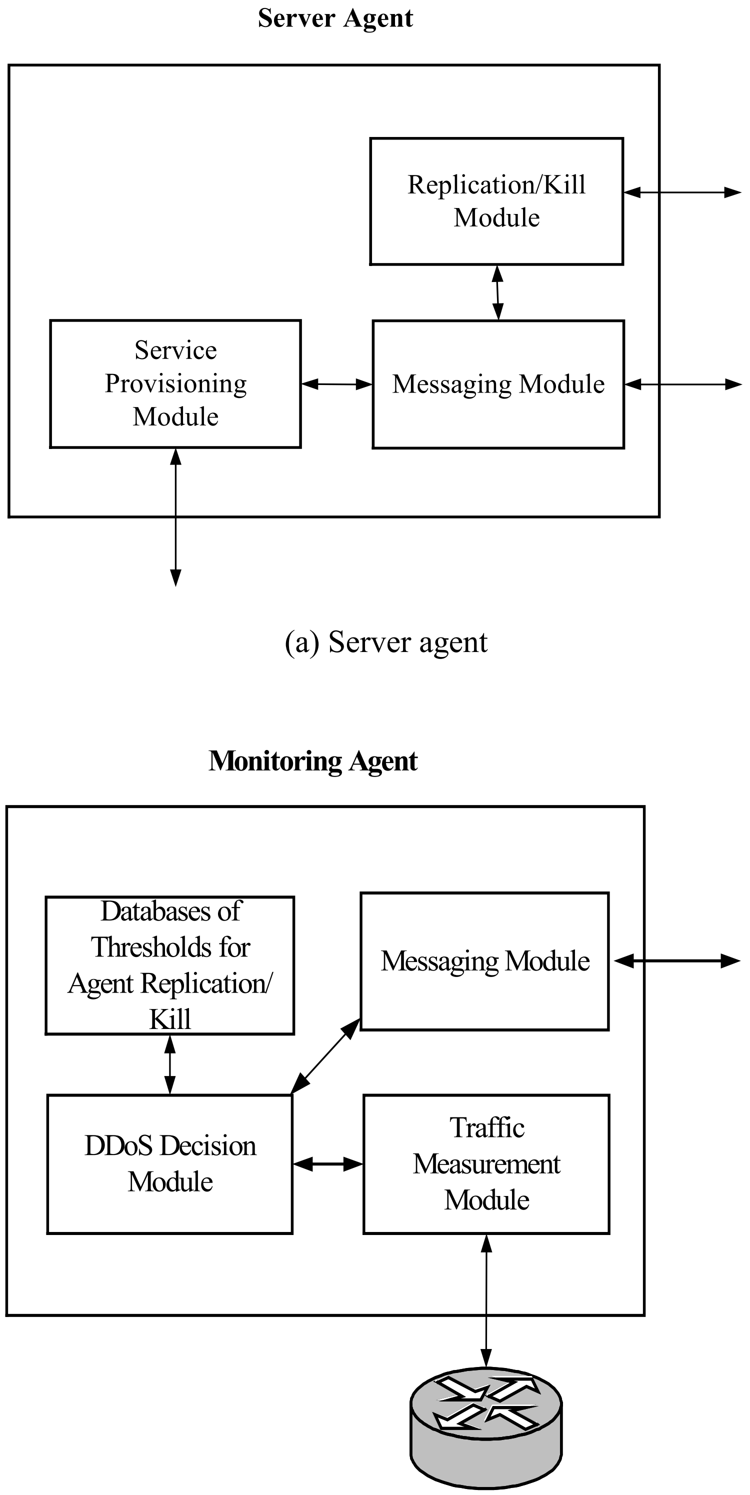


Fig. 2 Thresholds for replicating/killing the server agent

agent, the server agent may be replicated and killed again and again. To avoid the ping-pong effect, the threshold for killing the server agent must be set to be less than the threshold for replicating the server agent, as shown in Fig. 2.

Figure 3 shows the functional structures of the monitoring agent and the server agent. The service provisioning module is the rewritten program code of the server. The replication/kill module can replicate or kill the server agent. The messaging module is responsible to communicate with other agents or MAS. The traffic measurement module is responsible to collect the statistics of the traffic passing through the associated node. The database stores the predefined thresholds to replicate or kill the server agent. The DDoS decision module compares the traffic statistics and


Server Agent
Replication/Kill Module
Service Provisioning Module
Messaging Module
Monitoring Agent
Databases of Thresholds for Agent Replication/ Kill
Messaging Module
DDoS Decision Module
Traffic Measurement Module

(a) Server agent

(b) Monitoring agent

Fig. 3 Functional structure of (a) server agent and (b) monitoring agent

the thresholds to determine the status of the network and the associated node, and then to decide whether the server agent should be replicated or killed.

(C) Scenario

The scenario is explained in Fig. 4. As shown in Fig. 4(a), three nodes, MAS B, MAS C and MAS D, are connected directly to MAS A, and the thresholds of the link utilization for the three links are assigned to be 0.2, 0.7 and 0.4, respectively. The server agent is running in MAS A to provide a specific service. Assume that DDoS attacking packets are issued from somewhere to paralyze the service provided by the server agent located on MAS A, and these attacking packets pass through MAS C. As long as the monitoring agent of MAS A detects that the utilization of the link between MAS C and MAS A exceeds 0.7, the server agent in MAS A replicates itself and dispatches the replicated one to MAS C. After the replicated server agent is executed in MAS C, all requests for the service on MAS A will be intercepted transparently by the replicated server agent. That is, MAS C logically disconnects the connection from MAS C to MAS A, as shown in Fig. 4(b). The server agent in MAS A then just serves the requests sent from MAS B and MAS D. Because the server agent replicates itself before paralysis, and the load to serve all requests are shared by the two server agents,

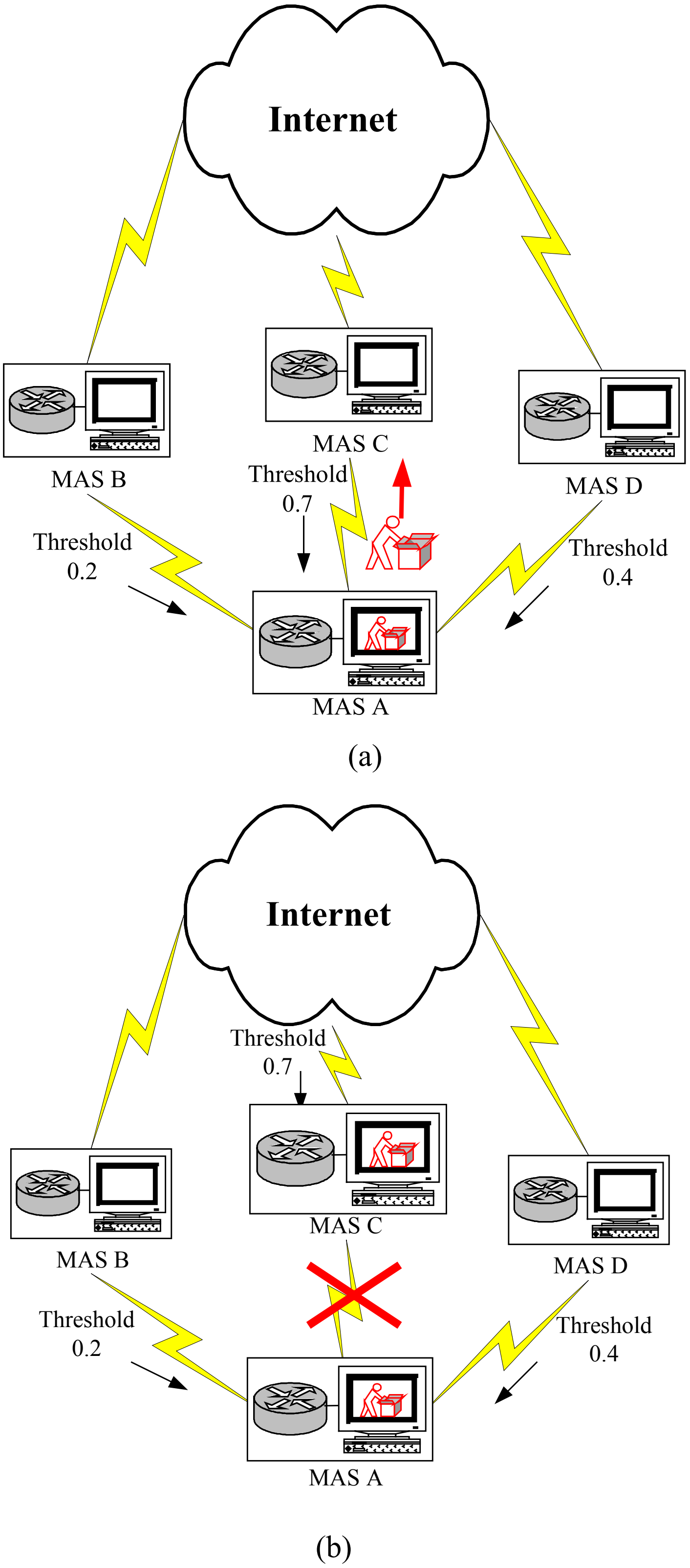
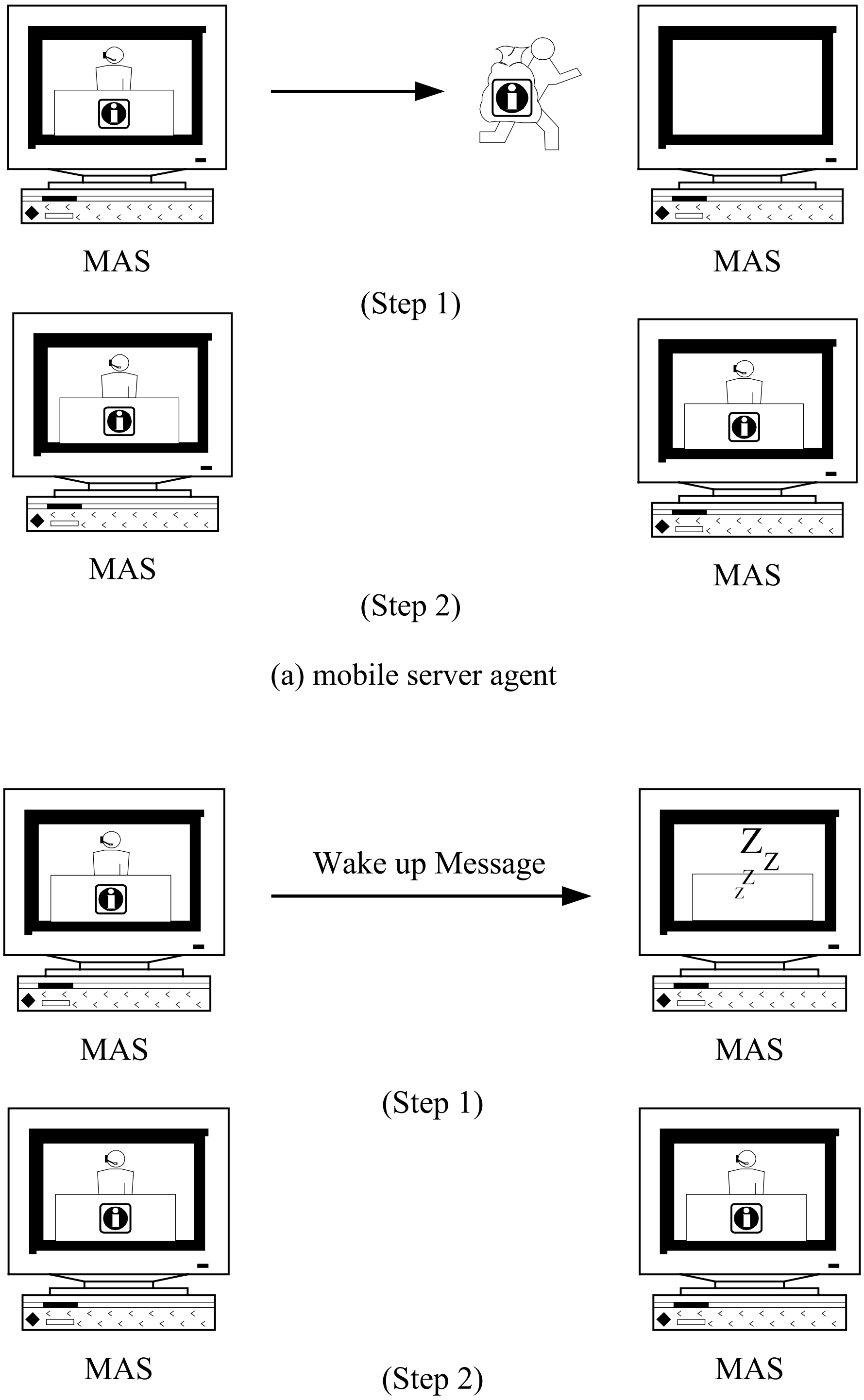


Fig. 4 Operational procedure of the proposed network cooperative security architecture using mobile agent technology

both server agents will operate normally.

As shown in Fig. 5, there are two methods to implement the server agent. The server that only provide simple service can be rewritten in the form of mobile agents. If the size of the program code for the server agent is not large, the server agent can be implemented to be a mobile agent, and can move among the MAS nodes. If the size of the program code is too large, it is better to implement the server agent as a stationary agent. The stationary server agent is resident and dormant in all MAS nodes initially. The dormant server agent will be invoked, as the waked up message issued by the monitoring agent of other nodes is received.



(a) mobile server agent

(b) stationary server agent

Fig. 5 Two types of server agents

IV. Conclusions

The paper adopts the concept of mobile agents to propose a cooperative security architecture to defeat the DDoS attacks, so that the service can be sustained. The mobile agents technology provides a flexible and intelligent network environment, and multiple agents can cooperate with each other to defeat the DDoS attacks. For the normal traffic and the DDoS attacking traffic are difficult to be discriminated, the proposed network cooperative security architecture does not exactly separate them. Instead, the anti-interception mechanism is activated in advance according to the degree of congstion and load. The monitoring agent measures the utilized resources in nodes and links, and issues a notification to the server agent as any abnormal situations are detected. The server agent then replicates itself and dispatches the replicated one to the neighboring heavy-loaded node or through the most congested link. All requests will be served only by one of the two server agents, and thus the service will not stop.

References

[1] R. J. Atkinson, “Toward a More Secure Internet,” *IEEE Computer Magazine*, vol. 30, no. 1, pp. 57-61, Jan. 1997.

[2] “TCP SYN Flooding and IP Spoofing Attacks,” *CERT Advisory CA-96.21*, Sept. 1996.

[3] L.-D. Chou and S.-L. Wu, "Precautionary measures against TCP SYN flooding attacks," *Proceedings of IFIP WCC 2000—World Computer Congress: The 15th International Conference on Information Security*, Beijing, China, Aug. 2000.

[4] “Distributed Denial of Service Tools,” http://www.cert. org/incident_notes/IN-99-07.html.

[5] “Distributed Denial of Service Attack,” *http://www. cert.org.tw/data/DDoS.htm*.

[6] David Dittrich, “The DoS Project's "trinoo" Distributed Denial of Service Attack Tool,” *http://staff.washington.edu/dittrich/misc/trinoo.analysis*, Oct. 1999.

[7] David Dittrich, “The "Tribe Flood Network" Distributed Denial of Service Attack tool,” *http://staff.washington.edu/dittrich/misc/tfn.analysis*, Oct. 1999.

[8] David Dittrich, “The "stacheldraht" Distributed Denial of Service Attack Tool, *http://staff.washington.edu/dittrich/misc/stacheldraht.analysis*, Dec. 1999.

[9] David Dittrich, George Weaver, Sven Dietrich and Neil Long, “The "mstream" Distributed Denial of Service Attack Tool,” *http://staff.washington.edu/dittrich/*

misc/mstream.analysis.txt, May 2000.

[10] Sven Dietrich, Neil Long and David Dittrich, “An Analysis of the ``Shaft" Distributed Denial of Service Tool,” *http://netsec.gsfc.nasa.gov/~spock/shaft_analysis.txt*, Mar. 2000.

[11] Vern Paxson, “An Analysis of Using Reflectors for Distributed Denial-of-Service Attacks,” *http://www.icir.org/vern/papers/reflectors.CCR.01/reflectors.html*, June 2001.

[12] V. A. Pham and A. Karmouch, “Mobile Software Agents: An Overview,” *IEEE Communications Magazine,* vol. 36 pp. 26-37, Jul. 1998.

[13] WJ Buchanan, M Naylor and AV Scoot, “Enhancing network management using mobile agents,” *Proceedings of the Seventh IEEE ECBS 2000 International Conference and Workshop on the Engineering of Computer Based Systems*, Edinburgh, U.K, pp.218-226, Apr. 2000.

[14] H. Reiser and G. Vogt, ”Threat analysis and security architecture of mobile agent based management systems,” *Proceedings of Network Operations and Management Symposium*, pp. 979-980, 10-14 Apr. 2000.