Cooperative Security Management Enhancing Survivability Against DDoS Attacks

Sung Ki Kim, Byoung Joon Min, Jin Chul Jung, and Seung Hwan Yoo

Dept. of Computer Science and Engineering, University of Incheon,
177 Dohwa-dong Nam-gu, Incheon, Republic of Korea 402-749
{proteras, bjmin, smjiny, blueysh}@incheon.ac.kr

Abstract. In this paper, we propose a cooperative management method to increase the service survivability in a large-scale networked information system. We assume that the system is composed of multiple domains and there exists a domain manager in each domain, which is responsible to monitor network traffics and control resource usage in the domain. Inter-domain cooperation against distributed denial of service (DDoS) attacks is achieved through the exchange of pushback and feedback messages. The management method is designed not only to prevent network resources from being exhausted by the attacks but also to increase the possibility that legitimate users can fairly access the target services. Though the experiment on a test-bed, the proposed method was verified to be able to maintain high survivability in a cost-effect manner even when DDoS attacks exist.

1 Introduction

As the Internet becomes increasingly important as a business infrastructure, the number of attacks, especially distributed denial-of-service (DDoS) attacks continuously grows [2]. Most of networked information systems adopt intrusion prevention mechanisms such as firewalls, cryptography and authentication. Nevertheless, many successful attacks exploiting various vulnerabilities are found. Intrusion detection systems (IDSs) can effectively detect pre-defined attacks but have limitations in responding to continuously created novel attacks.

The size and complexity of a large-scale networked information system such as Internet makes it impossible to centrally manage the entire management process. Moreover, it is difficult for the systems configured with different management policies to control the system without imposing any limitations. We therefore adopt a distributed management approach.

We assume that the large-scale networked information system can be divided into multiple domains. Each domain can be defined as a group of networks that contain one or more autonomous management entities called domain managers. The term 'autonomous' means that a representative manager of a domain can make a decision on management policies and uniformly apply them to the network components of the domain.

Recently, there have been a lot of research efforts to defend DDoS attacks, which include rate-limiting, blackhole routing, and IP tracing-back [3,5,9,10,11,12]. These

O. Gervasi et al. (Eds.): ICCSA 2005, LNCS 3480, pp. 252 – 260, 2005.

techniques are mainly to prevent network bandwidth from being exhausted by the DDoS attacks. Some of them have been adopted by Internet service providers (ISPs) and by network facility providers. However, not much of works have been studied to consider service survivability. Getting rid of DDoS attacks does not necessarily mean high survivability of services. Even though current measures can isolate a DDoS attack successfully, legitimate users may still suffer from being blocked to the access to target services.

In order to maintain high survivability of essential services, an inter-domain cooperation method against distributed denial of service (DDoS) attacks is proposed in this paper. The cooperation is based on the exchange of pushback and feedback messages among domain managers. This idea is not only to prevent network resources from being exhausted by the attacks but also to increase the possibility that legitimate users can fairly access the target services.

The rest of this paper is organized as follows. Section 2 summarizes related research results and explains the contribution of the research presented in the paper. In Section 3, in order to evaluate the performance of the management method, we define a survivability metric. Section 4 presents our distributed system architecture. Proposed mechanisms for inter-domain cooperative management are explained in Section 5. In order to verify the performance of the proposed mechanisms, a test-bed was implemented and several experiments were conducted. Section 6 presents the implementation and experimental results. Finally, Section 7 concludes the paper.

2 Related Works

This section is to provide background on what methods are currently available for protection against DDoS attacks and what their limitations are. Defense techniques against DDoS attacks include Access Control List (ACL), unicast Reverse Path Forwarding (uRPF), access rate limiting, traffic flow analysis, and remote triggered blackhole routing [5,9,10,11,12,13].

ACL is to cut the access off from the resources to be protected based on IP address, service ports, and contents. However, this method can be practical only when specialized hardware modules are equipped, otherwise it could be a big burden to the network facilities. It also requires access control policy to be updated in an efficient manner.

uRPF is to isolate IP spoofing attacks. As a packet arrives at a router, the router verifies whether there exists a reverse path to the source IP address of the packet. For most of DoS or DDoS attacks using IP spoofing, this technique is efficient. However, it has limitation when there are multiple routing paths. Besides, it only can prevent the IP spoofing.

When the amount of packets with a specific pattern increases up to a threshold, access rate limit technique limits the packets. This technique is also called rate filtering. The limitation of this technique is that it limits not only attacking packets but also normal packets.

Traffic flow analysis method is to monitor the source and destination addresses, the number of packets in each flow, and the upstream peer information. It can identify the interface from which spoofed traffics come. But, it requires access to other network facilities between the attacker and the victim.

Blackhole routing is to drop attacking packets toward a specific destination, by forwarding the packets to a virtual interface called Null0. Since this technique uses the forwarding function of the network facilities, it does not incur overload as ACL. However, it is confined only to layer 3 filtering.

In remote triggered blackhole routing, we need to install this function into edge routers. These routers are driven by blackhole routing servers in the same networks. The servers advertise it using Border Gateway Protocol (BGP) to multiple edge routers in order to forward packets with specific patterns to the blackhole IP block. This server can be designed to announce new routing information to other edge routers. It can be managed in Network operations centers (NOCs) or Security Operations Center (SOC) in order to manage novel attacks. This technique seems efficient in blocking DDoS attacks. But once an IP address is isolated, the service through the IP address is not accessible even by the legitimate users.

When we detect DDoS attacks, the most important step is how to react to the attacks. The common reaction to DDoS attacks is to put a filter in the router or the firewall where DDoS attacks are found. By filtering the malicious traffic, the particular website or local network could survive the attack. However, there are two aims for DDoS attacks. The first one is to flood a particular server and another one is to congest the network links. Although we can protect the server by blocking the malicious traffic locally, the attacker can still achieve his goal by flooding the network links. Thus, the best way is to push the filter back to the attack source. The closer the filter is to the source, the more effective is to protect the network link from being flooded. In this scheme, the downstream router needs to contact all its upstream neighbors and all the upstream neighbors need to estimate the aggregate arriving rate. This additional processing makes the router implementation much more complicated [4].

The contribution of this paper is demonstrating a cost-effective approach to support high survivability of essential services against DDoS attacks. We propose a cooperative management method based on the exchange of pushback and feedback messages among domain managers. The management method is designed not only to prevent network resources from being exhausted by the attacks but also to increase the possibility that legitimate users can fairly access the target services. Though the experiment on a test-bed, we have verified the performance of the method.

3 Survivability Metric

In this paper, we define the survivability metric. This is to measure the estimated survivability of a service.

If a legitimate user obtains results on service requests in a timely manner whenever he or she wants to access the service, we should say that the service survives in good shape. On the other hand, if the user cannot attain any result in spite of repeated requests in a long time span, the service should be considered as dead. Therefore, we define the metric as the average of aggregated ratios of the number of replies returned in a time limit to the number of requests sent to the server.

$$\text{Survivability} = \sum_{i=1}^{n} \left(\frac{\text{Reply}_i}{\text{Request}_i} \times \text{P}_i \right) \; / \; \sum_{i=1}^{n} \text{P}_i$$

where, n is the number of users considered in a time slot, $Reply_i$ and $Request_i$ are the number of replies returned to the user i within the time limit and the number of requests sent to the server by the user i, respectively, and P_i is weight of user i. If all users are treated with the importance, P_i may be integer 1. Otherwise, it can be set differently for each user.

The metric should be between one and zero. When all the requests sent by every user are replied, it is one. If no reply is return from the server, it is zero, which can be interpreted as server crash.

4 Architecture for Cooperative Management

This section presents distributed system architecture. We need to redefine networked information system in order to fully support cooperative security management. The following requirement should be satisfied in such system architectures.

(1) Practically, the architecture should be applicable to the current information infrastructure. Heterogeneous resources including routers, switches, and network servers cannot be replaced at once. Apparently, drastic changes in the network would incur tremendous costs.

(2) High speed network performance should not be harmed too much. Degradation of network server performance should be acceptable at the cost of security management.

(3) The architecture needs to be suitable for automatic management process. We need to reduce the involvement of manual operations as much as possible.

We assume that the large-scale networked information system can be divided into multiple domains. Each domain can be defined as a group of networks that contain one or more autonomous management entities called domain managers as depicted in Figure 1 and Figure 2. Intra-domain architecture is illustrated in Figure 1. As shown in the figure, each domain can be further divided into sub-domains. The boundary of a domain defines autonomous management, which means that a representative manager of a domain can make a decision on management policies and uniformly apply them to the netw ork components within the domain. Figure 2 shows inter-domain relationship.

We define domain at a network system which can be managed autonomously. In a domain, there should be a representative manager which can assign management policies. A domain can be subdivided into multiple sub-domains.

Domains are connected each other through edge routers. An edge router is connected to a computing node which is able to monitor inbound and outbound traffics. This node is called a domain manager.

Within a domain, each node contains an agent, which is to monitor usages of resources such as CPU, memory, and network bandwidth. The agent is also responsible to trigger resource reallocation in the node and to report its situation to the Domain Manager.

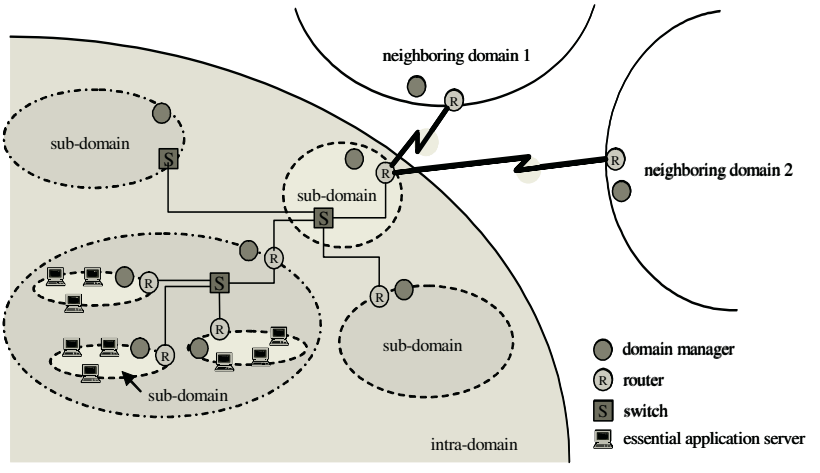

Fig. 1. Intra-Domain Architecture

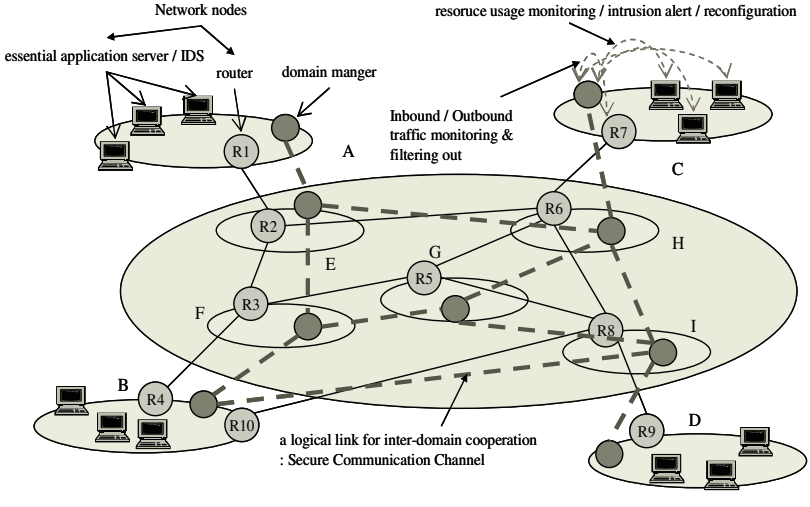

Fig. 2. Inter-Domain Architecture

5 Proposed Mechanisms to Enhance Survivability

This section is to explain management mechanisms proposed in this paper. They are in two categories. One for intra-domain, and the other is for inter-domain.

5.1 Management within a Domain

Since the number of network nodes is confined in a domain, it is relatively easy to treat DoS attacks. Therefore, it is necessary to monitor outbound traffics generated in

a domain. The objectives of the monitoring are to detect abnormal outbound traffic flows and to provide essential services in the domain with enough bandwidth.

A domain manager collects packet headers periodically. From this information, it can detect IP spoofing and service port access violation. Statistics based on traffic flows also can be obtain in the process.

5.2 Inter-domain Cooperative Management

Inter-domain cooperation should be based on trust. Messages exchanged among domain managers are authenticated. In order not to be revealed to any attacker, the messages are encrypted and handled by the domain managers.

For this purpose, domain managers conduct inbound traffic monitoring. It is to detect abnormal traffics and to control bandwidth for essential services.

There are two types of messages exchanged among domain managers. One is the pushback message to cut off the traffic toward a certain victim node. The other is the feedback message. The feedback message is to increase the survivability as much as possible. Once an attack is controlled successfully by the virtue of the pushback message, the domain manager issues the feedback message back to the origin of the pushback message. Other domain managers receiving the feedback message cease the rate limit and return to the status before the corresponding pushback message was generated.

6 Implementation and Experimental Results

In this section, we explain implementation of a test-bed, experimental environment and the results.

6.1 Implementation of Test-Bed

TFN2K is a typical tool that is used to create a DDoS attack. It contains most of all kinds of DDoS attack methods. Master programs sending attack command messages communicate with agent programs by exchanging encrypted messages. The attacker can distribute attacking agents to computer systems with weak security measures while the attacker itself is hidden.

In Figure 3, the domain manager of S_A in which the victim V is contained forwards a pushback message to upstream domain manager of A. The pushback message requests rate-limit of packets which is directed to a certain service port of V. The domain manager A checks whether spoofed attacking packets exist. If the domain manager A cannot find them, it forwards the pushback message to the next hop domain manager C. This continues until the source of the attacking traffics. And then the corresponding domain manager isolates the attacker and generates a feedback message back to the origin of the pushback message. For example, once domain manager E detects and isolates A1, it forwards a feedback message through the pass of R9-R4-R3-R2-R1. This is to increase the survivability of the service to legitimate users.

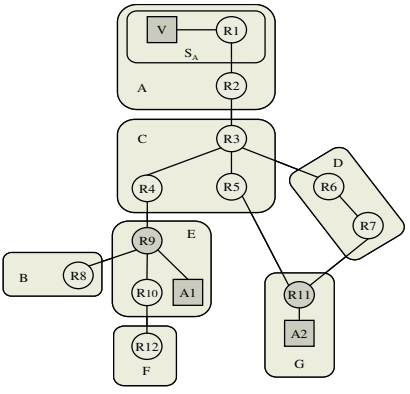

Fig. 3. DDoS Attack Situation

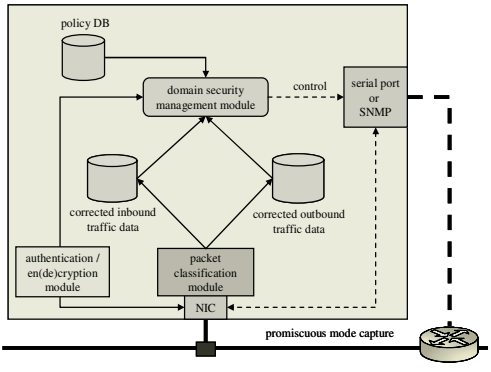

Fig. 4. Structure of Domain Manager

A domain manager is closely coupled with a router to monitor inbound and outbound traffics. It logs IP source addresses, monitors available network bandwidth, and detect abnormal flows. Besides, it exchanges control and policy information with neighboring domain managers through secure communication channels. The messages exchanged among domain managers include pushback messages to filter attacking traffics toward a victim and feedback messages to recover traffic flow after the filtered situation made by the pushback messages. Figure 4 shows the structure of a domain manager.

The message structure includes an array storing 16 IP addresses, a source address table containing up to 5,000 collected addresses, authorization information, flag notating either pushback or feedback, and message identification. As the message passes by domain managers, each of them records its address into the array of the message. When the trace is over, the message is coming back to the origin of the message as a feedback. The message identification number is attached when it is created in a domain manager.

6.2 Experimental Results

Figure 5 depicts the experimental environment. It consists of three domains. In each domain, there is a domain manager. The domains are connected each other through Linux Routers.

We select the service provided by victim server as a file transfer. The average size is 130 M Byte. Domain managers take samples of packets in every 1 m sec. We use 6 attackers to simulate DDoS attacks. Spoofed ICMP packet flooding is generated with periods of 1 m sec, 10 m sec, 50 m sec, and 100 m sec. Figure 6 shows the raw data obtained from the experiments.

We measured the survivability metric defined in Section 3. By using the cooperation mechanism, the survivability can be increased from 0.2 to 1.0 in the best case when the service deadline is set to 140 seconds in the experiment.

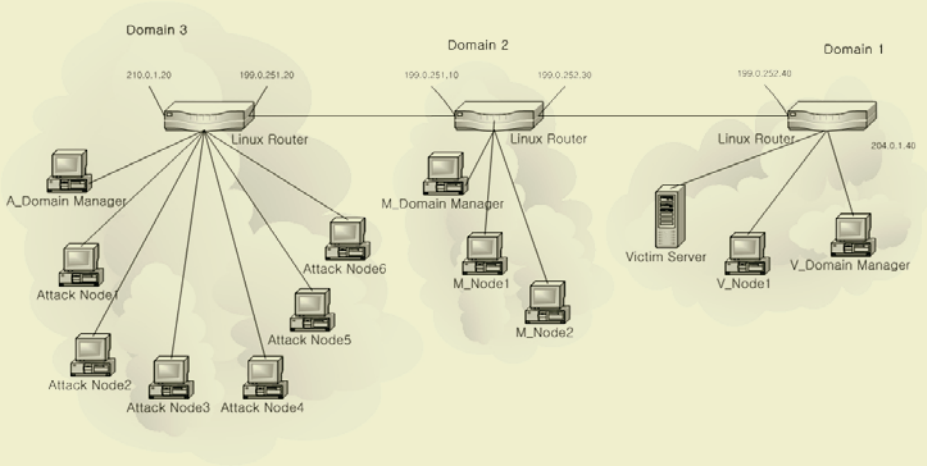

Fig. 5. Test-bed System

Packet Flooding interval	resource usage in victim's domain			spoofed packet detection rate (based on 1 hop)		file transfer completion times (130 M Byte file size, CBR stream)			
	observed the average # of flows in every a 5 second period	pps	bps	# spoofed packets	# detected Spoofed packets	condition without attacks	condition with attacks		
							condition without CM	condition with CM	elapsed time until PM triggered
1 ms	837	1330	6 Mbps	1000	916	135 sec	187 sec	139 sec	10 sec
10 ms	219	459	4.6 Mbps	1000	925		151 sec	137 sec	15 sec
50 ms	77	482	2.8 Mbps	1000	963		141 sec	137 sec	30 sec
100 ms	32	129	1 Mbps	1000	992		138 sec	138 sec	not measured

Fig. 6. Data Obtained from Experiments

7 Conclusion

In this paper, we have proposed a cooperative management method to increase the service survivability in a large-scale networked information system. The system is composed of multiple domains and there exists a domain manager in each domain, which is responsible to monitor network traffics and control resource usage in the domain. Inter-domain cooperation against distributed denial of service (DDoS) attacks is achieved through the exchange of pushback and feedback messages. The management method is designed not only to prevent network resources from being

exhausted by the attacks but also to increase the possibility that legitimate users can fairly access the target services.

In order to evaluate the performance of the method, we have implemented a test-bed, and conducted a set of experiment. As a result, we found that with the help of the method, the survivability of services can be increased fundamentally with reasonable amount of inherent management cost.

This method can be integrated with network management systems in the future. Through this work, we were able to demonstrate a cost-effective approach to support high survivability of essential services against DDoS attacks.

Acknowledgement

This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment).

References

1. William Aiello, John Ioannidis, and Patrick McDaniel : Origin Authentication in interdomain routing, Proceedings of the 10th ACM conference on Computer and communications security, Oct.(2003)
2. Tatsuya Baba and Shigeyuki Matsuda : Tracing Network Attacks to Their Sources, IEEE Internet Computing, March-April(2002), 20-26
3. Andrey Belenky and Nirwan Ansari : On IP Traceback, IEEE Communications Magazine, July(2003)
4. John Ioannidis and Steven Bellovin : Implementing Pushback: Router-Based Defense Against DDoS Attacks, Proceedings of the Network and Distributed System Security Symposium, Feb(2002)
5. KICS of Korea Information Security Agency : Intercept and Analysis Technologies Against DDoS Attacks, Sep(2004)
6. Anukool Lakhina, Mark Crovella, and Christophe Diot : Characterization of Network-Wide Anomalies in Traffic Flows, IMC'04, Oct(2004)
7. Ratul Mahajan, Steven M.Bellovin, Sally Floyd, John loannidis, Vern Paxson, Scott Shenker : Controlling High Bandwidth Aggregates in the Network, ACM SIGCOMM Computer Communications Review, Vol. 32, No. 3, Jul(2002)
8. Byoung Joon Min, Sung Ki Kim, and Joong Sup Choi : Secure System Architecture Based on Dynamic Resource Reallocation, WISA 2003, LNCS2908, Aug(2003)
9. Tao Peng, Christopher Leckie, and Kotagiri Ramamohanarao : Defending Against Distributed Denial of Services Attacks Using Selective Pushback, Proceedings of the 9th IEEE Int'l Conference on Telecommunications, Jun(2002)
10. BGPExpert.com : How to Get Rid of Denial of Service Attacks, http://www.bgpexpert.com /antidos.php".
11. Cisco : Unicast Reverse Path Forw -ding(uRPF) Enhancements for the ISP-ISP Edge, ftp://ft-eng.cisco.com/cons /isp/security/URPF-ISP.pdf.
12. waterspring.org : Configuring BGP to Block Denial-of-Service Attacks, http://www.water springs.org/pub/id/draft-turk-bgp-dos-01.txt