International Journal of Multimedia and Ubiquitous Engineering
Vol. 2, No. 1, January, 2007

Security Management against DDOS attack in Medical Systems Architecture

Samir Kumar Bandyopadhyay

Professor of Computer Science & Engineering, University of Calcutta, Kolkata, India
skb1@vsnl.com

Abstract

Large Scale networked information systems presents a multiple domains through which possibility of distributed denial of service (DDOS) attack is foreseen. In this paper a cooperative security management method and to elevate the service survivality to this large scale networked information system is being presented.

Keywords: *DDOS, Security Management*

1. Introduction

Internet becomes very important especially in business infrastructure. However, as the infrastructure gets bigger the vulnerabilities get even larger. DDOS (Distributed denial –of-service becomes common, [1][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.

This paper presents the existing Sensor Network for Medical Systems Torso Architecture and suggests a Security Management Enhancing Survivality against DDOS attacks. The preparation Wireless Sensor Networks (WSNs) have been a subject of extensive research with their use being advocated for a wide variety of applications. WSNs applied to medical technologies have recently emerged as an important application, with fusion of wireless secure communication and medical techniques with sensing devices [3], [4], [5]. Biomedical sensors are being developed for retinal prosthesis to aid the visually impaired. Our research forms the basis for another medical application where the sensors form a distributed network over the patient's body. The medical application is a new concept wherein we use the wireless network technology to monitor patient's vital functions and provide instantaneous medical

feedback. There has been a lot of research into medical applications but all were aimed at making the network mobile.

The security aspect of the communication was not considered as security in sensors is believed to be expensive. In this paper, we propose the Torso Architecture, which is a distributed layered approach to monitoring patients, and also explain the security protocol embedded in this architecture which makes the communication secure and efficient. The envisioned Torso distributed sensor network for patient monitoring and care has a leaf node layer which follows a ring architecture consisting of patient's sensors which are self organizing, called the SENSOR LEVEL. The intermediate layer consists of a super node, which acts as a supervisor to the leaf nodes and also resides with the patient, called the SUPER-NODE LEVEL. The final layer is the root node or the central base station. This concept provides an individual the flexibility to roam around freely without having to wait at the treatment centers and thus giving the individuals higher QOL (Quality of Life) [6]. The concept of wireless medical treatment is achieved by the patient carrying a sensor network that communicates with the root, which is the doctor's access point.

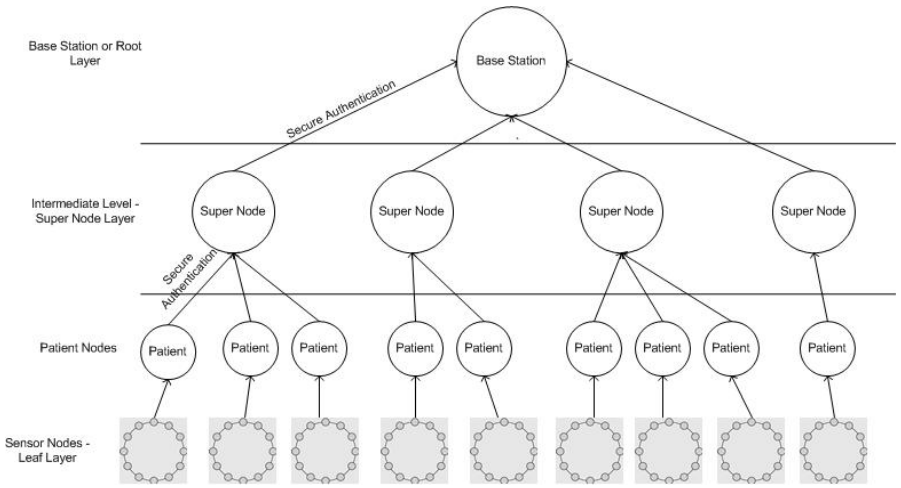

Figure 1. Torso Architecture

2. Sensor Networks

Sensor Networks are data-centric rather than address-centric. Queries are directed towards a cluster of sensor nodes rather than specific sensor addresses. The medical applications of sensor networks have long been a research area focusing on patient monitoring. However, in many cases the bottleneck of bandwidth limits the usage of these applications. In this paper, we present a way to monitor patients using distributed sensor networks to form a sensing ring architecture over the human body to monitor patient's vital information and provide instantaneous medical feedback. We present a layered approach to monitor the patient's health condition and propose a schema for better utilization of the bandwidth called the Torso Architecture. We define a criticality quotient for every patient that determines the amount of bandwidth allotted to them. The Torso Architecture provides the capability independent of

wired monitoring and diagnosis by way of a ring layered architecture spread over the patient's body. The patient communicates with the supervisor that will communicate in turn with the base station which is the treatment center. The leaf layer or the sensor layer is sensors that sense the information and transmit it to the next level in the hierarchy, which is the patient layer. The patient layer is followed by the intermediate node layer and then on top of the hierarchy is the root.

2.1. Layers of the Torso Architecture

2.1.1. Leaf Layer: The leaf layer is a collection of sensor nodes forming a ring architecture across the human body which communicate over a wireless network. The individual sensors do not have any processing power but are mere sensing devices. The leaf nodes form a ring architecture for reliability, efficiency and more accurate communication with the higher levels in the hierarchy. The leaf layer is also called the Sensor Node Layer or Sensor Node Level. The formation of the leaf layer is critical to the entire network as they form a ring architecture and based on this formation the nodes sense the vital information and send the information up the Torso architecture to the root which is the treatment center. The leaf layer sensors have no additional capability then mere sensing and passing the information one layer up the hierarchy.

2.1.2. Formation of the Ring Architecture. Self-organization refers to the ability of the system to achieve the necessary organizational structures without requiring human intervention, particularly by specially trained installers and operators. Self organization is a critical attribute needed to achieve the wide use and applicability of distributed sensor networks.

2.1.2. Advantages of the Ring Architecture : The ring architecture formation over the human body gives the sensor network the all important stability with respect to the body movements. The communication protocol is simple and straightforward. The importance of any sensor or a leaf node is decided on the patient's health condition. For example, if a patient is suffering from a heart attack, the most important sensor node could be the heart beat sensor while for a patient suffering from asthma, the all important sensor node could be something totally different.

2.1.3. Patient Node Layer: This node layer represents the patient itself. They are two ways to present this node. The node can itself reside on the patient. The patient node is a supervisor of all the leaf nodes of that patient and gathers information from all the sensors and sends the information to the intermediate super node that it interacts with. The patient node collects all the information from the leaf nodes and does the processing of information and sends it to the intermediate layer. The patient node is responsible for communicating with all of the leaf nodes. The patient node receives feedback from all the sensor nodes and sends them to the intermediate node. The patient node is directly responsible to the above super node that it belongs to. Once authenticated the patient node is now responsible for that super node and sends the information to the intermediate node which in turn sends the information to the root.

2.1.4. Intermediate Layer – Super Node Layer: The super node layer is the next layer to the central root. This node is responsible for the up and down communications with the patient layer and the root. This layer receives information from all the patient nodes that are

under this layer. The distribution of the patient nodes under a particular super node is done on the basis of proximity, geographical location. Depending on the current location of the patient, the patient node is controlled by a different super node. The patient node authenticates to the node before the node can send information to the intermediate node. The super node communicates with the root and sends all the information. The link from the patient node up to the intermediate node and the link between the intermediate node and the root node are both band-limited due to the limited wireless link capacity. Both the links are prone to malicious attacks and hence require a secure and efficient communication up and down the Torso architecture. Each node in this layer will be in contact with its geographical neighbors.

3. Communication

All the communication between the layers of the Torso architecture is wireless and as the result is bandwidth limited and is also prone to various attacks. The Denial of service and eavesdropping of crucial information have always been a threat to wireless networks. Eavesdropping of information may not be externally harmful but in case of medical applications it is not desirable to send information in the open as the data being sent may contain some confidential information. Trusted third party based architectures are impractical for sensor networks because of the resource constraints that sensing devices have. A unique light weight key exchange mechanism is required for secure communication. The communication bandwidth also forms a bottleneck because of the limited bandwidth of wireless networks, a huge number of remote patients cannot transmit information at the same time which is not desirable in medical applications as the number of patients in a particular region cannot be predetermined.

Three most important bottlenecks for sensor network communication.

− The bandwidth bottleneck of the wireless link from the patient node to the intermediate layer and from the intermediate layer up to the root node.
− The security, confidentiality and privacy of the information being transferred.
− The limitation of battery power of the patient nodes.

Bandwidth has always been a bottleneck in wireless communication. A better utilization of the available bandwidth depends on the criticality of the patient. *Criticality* is the term used to describe the importance of the doctor's advice. We describe this value as the basis for determining the bandwidth allotted to that particular patient. The value can range anywhere from 0 to 100, where each of these values has its own meaning. The value of 100 meaning the patient requires attention and a value of 0 requires no attention and hence no bandwidth. Therefore, this scheme of better utilization of the bandwidth works really well in very constrained systems. The information exchange between the sensors is vital and should therefore be required to maintain confidentiality. The secure mechanism provides the required confidentiality at a very low cost computational power.

3.1 Bandwidth Bottleneck

The link between the root node and the intermediate layer is bandwidth constrained. The root node maintains a table for different intermediate nodes with the number of patients each of them are addressing. Based on this number, the root node determines the amount of bandwidth to be allotted to each of the intermediate nodes. Each patient is also associated

with a criticality quotient, which determines the limit of bandwidth that the patient node gets allotted The root node determines and calculates the Bandwidth Allocation Based on the input from its intermediate children and generates the BAT table as shown and calculates the allocation bandwidth percentage.

The Cardinality of the intermediate nodes is the defined as the number of patients it is supervising. We will discuss more about the cardinality as we go through the paper. The root node determines bandwidth allocation based on the bandwidth allocation table at the particular instant. The bandwidth at this instant is allocated 50% based on the BAT to an intermediate node with the Node Id 12. Thus, the root node best utilizes the available bandwidth and allows the patient with more importance of treatment process more information, thus acting as a life saver. The Intermediate node calculates the bandwidth required directly based on the criticality of the patient and distributes the allotted bandwidth among all of its patient nodes. The intermediate node 12 in the above example, distributes the bandwidth allotted to it based on the criticality of the patient. Assuming the intermediate node has just one patient node, all the bandwidth allotted to that intermediate node is utilized by the single patient that it serves and receives information from. The bandwidth utilization is therefore best utilized.

3.2 Distributed Denial of Service

An attempt to make a computer resource is unavailable to its intended users. Although the means to carry out, motives for, and targets of a DoS attack may vary, it generally consists of the concerted efforts of a person or people to prevent an Internet site or service from functioning efficiently or at all, temporarily or indefinitely. Perpetrators of DoS attacks typically target sites or services hosted on high-profile web servers such as banks, credit card payment gateways, and even root nameservers.

One common method of attack involves saturating the target (victim) machine with external communications requests, such that it cannot respond to legitimate traffic, or responds so slowly as to be rendered effectively unavailable. In general terms, DoS attacks are implemented by either forcing the targeted computer(s) to reset, or consuming its resources so that it can no longer provide its intended service or obstructing the communication media between the intended users and the victim so that they can no longer communicate adequately.

A "denial-of-service" attack is characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service. Attacks can be directed at any network device, including attacks on routing devices and web, electronic mail, or Domain Name System servers.

A DoS attack can be perpetrated in a number of ways. The five basic types of attack are:

1. Consumption of computational resources, such as bandwidth, disk space, or processor time
2. Disruption of configuration information, such as routing information.
3. Disruption of state information, such as unsolicited resetting of TCP sessions.
4. Disruption of physical network components.
5. Obstructing the communication media between the intended users and the victim so that they can no longer communicate adequately.

A DoS attack may include execution of malware intended to:

- Max out the processor's usage, preventing any work from occurring.
- Trigger errors in the microcode of the machine.

- Trigger errors in the sequencing of instructions, so as to force the computer into an unstable state or lock-up.
- Exploit errors in the operating system, causing resource starvation and/or thrashing, i.e. to use up all available facilities so no real work can be accomplished.
- Crash the operating system itself.

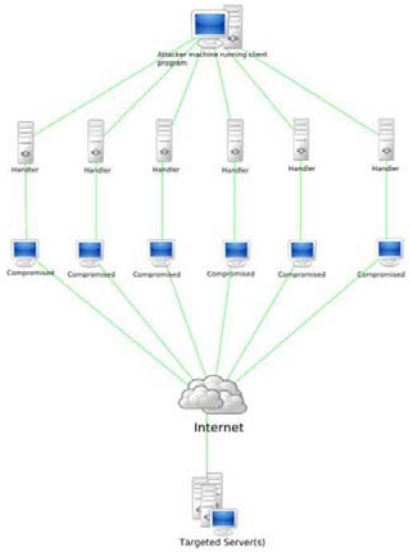

Figure 2. DDOS Attack

A distributed denial of service attack (DDoS) occurs when multiple systems flood the bandwidth or resources of a targeted system, usually one or more web servers. These systems are compromised by attackers using a variety of methods. Malware can carry DDoS attack mechanisms; one of the better-known examples of this was MyDoom. Its DoS mechanism was triggered on a specific date and time. This type of DDoS involved hardcoding the target IP address prior to release of the malware and no further interaction was necessary to launch the attack.

A system may also be compromised with a trojan, allowing the attacker to download a zombie agent (or the trojan may contain one). Attackers can also break into systems using automated tools that exploit flaws in programs that listen for connections from remote hosts. This scenario primarily concerns systems acting as servers on the web.

Stacheldraht is a classic example of a DDoS tool. It utilizes a layered structure where the attacker uses a client program to connect to handlers, which are compromised systems that issue commands to the zombie agents, which in turn facilitate the DDoS attack. Agents are compromised via the handlers by the attacker, using automated routines to exploit vulnerabilities in programs that accept remote connections running on the targeted remote hosts. Each handler can control up to a thousand agents.[8]

These collections of systems compromisers are known as botnets. DDoS tools like stacheldraht still use classic DoS attack methods centered on IP spoofing and amplification like smurf attacks and fraggle attacks (these are also known as bandwidth consumption attacks). SYN floods (also known as resource starvation attacks) may also be used. Newer tools can use DNS servers for DoS purposes. (see next section)

Unlike MyDoom's DDoS mechanism, botnets can be turned against any IP address. Script kiddies use them to deny the availability of well known websites to legitimate users.[1] More

sophisticated attackers use DDoS tools for the purposes of extortion — even against their business rivals.[9]

It is important to note the difference between a DDoS and DoS attack. If an attacker mounts an attack from a single host it would be classified as a DoS attack. In fact, any attack against availability would be classed as a Denial of Service attack. On the other hand, if an attacker uses a thousand systems to simultaneously launch smurf attacks against a remote host, this would be classified as a DDoS attack.

The major advantages to an attacker of using a distributed denial-of-service attack are that multiple machines can generate more attack traffic than one machine, multiple attack machines are harder to turn off than one attack machine, and that the behavior of each attack machine can be stealthier, making it harder to track down and shut down. These attacker advantages cause challenges for defense mechanisms. For example, merely purchasing more incoming bandwidth than the current volume of the attack might not help, because the attacker might be able to simply add more attack machines.

Although most DDoS attacks are malicious in nature, the same technique can be used to aid the Internet community. Internet fraud schemes, such as Nigerian 419 scams or phishing, commonly involve fraudulent websites that either impersonate a real website for purposes of stealing the victim's identity, or lend credibility to a scammer's fictional business venture to lure the victim into a false sense of confidence. Scam baiters, who combat these scams by posing as victims for the purpose of wasting the scammer's time and money and obtaining information that can be used by authorities, will forward sites they encounter during the course of their conversations to groups that specialize in site-killing.[citation needed] The group will first try to have a site taken down by informing the host of said site that the site is being used fraudulently. In the case where that approach fails, the group will organize a "takedown" of the site by encouraging its members to visit the site en masse and continually refresh its content (an intentional form of the Slashdot effect sometimes referred to as flash mobbing, although that term is technically reserved for real-world gatherings). Alternately, some groups have special web pages that link to images hosted by these fake sites and show the images to visitors (usually members or supporters of the site-killing group) while constantly reloading them, which is known as intentional bandwidth hogging.[citation needed] The purpose, similar to malicious DoS attacks, is to (a.) rapidly consume all of the website's allocated monthly bandwidth, after which requests for the site's content are refused, (b.) draw the attention of the site's host, who when faced with the constant onslaught on the entire hosting network's resources, will usually remove the site, and/or (c.) take up all available connections and maximum throughput of the host so that would-be victims cannot access the site.

4. Challenges

In this section some of the unique characteristics were identified that make security in SN different from usual networks.

Scalability: The patients authenticating with different intermediate nodes should be very efficient such that it will not add overhead to the communication of the network. Contributing key establishment protocols might not be most efficient in these networks where having such a large number of network nodes might actually slow down this process. The protocol should be scalable in terms of the patient nodes. We propose a mechanism where in the scalability of the overall network could be increased by using efficient communication mechanism there by

making the network more reliable when a patient requires attention and also providing better utilization of bandwidth which adds to the scalability.

Resource Constraints: One of the most important aspects of WSNs is the limited energy constraint. Depending on their role within the network, some of these nodes have some power recharging mechanisms. In order to ensure longer life for the nodes, energy efficient mechanisms and power conserving methodologies should be adapted at every level in the network. Pottie et al. have established that the energy cost of transmitting 1Kb over a 100 m distance is the same as the energy required by a general- purpose 100 MIPS/W processor to execute 3 million instructions. The protocol implemented for security should minimize the exchange of security related setup messages. Also the cryptographic metric selected for encryption should be small enough to capture the resource constraints. Our protocol implementation takes advantage of the above aspects of sensor networks.

− Data aggregation is less expensive then transmitting data.
− Not all the data that has been transmitted need to be encrypted.

4.1 Implementation Issues

The basic implementation issues of the security protocol is described in this section We detail key exchange between the patient node and the intermediate node for transmitting data and the communication between the intermediate and root node.

4.1.1 Assumption : the assumptions underlying the model. We assume that the radio model is symmetric i.e., given a signal-to-noise ratio, the energy required to transmit an m bit message from node A to node B is the same as the energy required to transmit the same m bit message from node B to A. We also assume that the root is more resourceful than the regular sensor node. We also assume that the intermediate nodes are more resourceful than the regular sensor nodes but not in the order of the root. The root with all its resources can store all the keys and access them directly without any overhead. We assume that each sensor node is created with a unique Device Identifier (DId) which is known only by that particular node. We also assume that the root has, built into it all the DIds for all the sensor nodes that have been dispersed into the network. We also assume that all the sensors have an in-built system clock.

4.3 Design Issues

In this section two phases are defined in the protocol, the first being the intermediate nodes joining the network and then each of the intermediate nodes authenticating with its neighbors. The other phase is the patient nodes authenticating when leaving or joining an intermediate node. The first part of the protocol implementation is key setup process. The key setup process is used to authenticate the nodes as part of the network and have the node assigned a key. We use the DId that has been determined for each of the sensor node and is stored in the root. Initially, each intermediate sends a JOIN-ROOT message to the root, by encrypting the Device Identifier along with the current Time stamp, using the public key of the root. When root receives the message, it decrypts using its private key and compares the Device Identifier with that of the database that it stores. If it matches, it authenticates the node as a part of the network and sends back an authentication message. The authentication message contains a unique temporary node identifier, which is used to communicate

henceforth along with a randomly generated number, encrypted using a key that is the MAC of the DId and the time stamp. This is the encryption key for the sensor nodes. The symmetric key is also computed at the sensor node using the MAC of its DId and the time stamp and is decrypted.

$$A \rightarrow ROOT - E_{PUBLICKEY}(A_{DId}, TimeStamp)$$
$$KEY_A(M) = MAC(A_{DId}, TimeStamp)$$
$$ROOT \rightarrow A - E_{KEY\,A}(NodeId_A, R_A)$$

Once all the intermediate nodes are authenticated, we have the first two layers of the network setup. The same process is executed with the patient nodes and the intermediate nodes. The Communication between the sensor nodes and the patient nodes is assumed to be secure as they are part of the human body. The patient nodes initially broadcast the JOIN-INTERMEDIATE NODE message that is encrypted using a predetermined symmetric key which is known to all intermediate nodes when manufacturing. We determine the Time To Live (TTL) to be the time for the message to travel from one end point to the other of the range of the intermediate node. We make sure initially that each of the intermediate node is at such a distance that there are no intersections in the areas covered nor there are any places that do not come under any of the intermediate node. Initially, every patient node broadcasts a JOIN-INTERMEDIATE NODE message, which is encrypted using the symmetric key provided for communication with the intermediate nodes. The patient node broadcasts its DId and TS. All the intermediate nodes receive the message, decrypt it using their copy of the symmetric key and do any kind of action only if Time taken by the message, is less then the predetermined Time To Live (TTL). Our assumption ensures that only one intermediate node

receives the message within the predetermined TTL and so that node is the parent of the patient node. The intermediate node, then communicates with the root once, for confirmation of the DId, that the Device actually belongs to the network and once confirmed, and it receives a Node Id and a random number, forwards the message to the patient node along with its Node Id and Key for further communication.

Time = System Time - Time Stamp
Time ≤ TTL for Node A

PN □ E symkey(PNDId, TimeStamp)
A → BS – EA(PNDId, TimeStamp)
BS → A – EA(NodeIdPN, RPN)
A → PN – EPN(NodeIdPN, RPN, NodeIdPN, NDId, EA-PN)

4.3 Node Joining and Leaving

The patient nodes are mobile and can leave and join different intermediate nodes based on proximity of the patient node to the intermediate node. When the patient node needs to send information, it sends the information along with a time stamp. Each intermediate node receives the message but only that node that receives the message within the specified TTL will reply to the message and hereafter the node sends the information to only that intermediate node. Once the patient node joins an intermediate node sends all the information to the intermediate node. But patient nodes are mobile and keep moving along with the patient. Every message to be sent from the patient node has a time stamp attached to it and if the time taken by the message to reach the intermediate node is more than the TTL then the intermediate node sends an invalid message to the patient node and the patient node

broadcasts its Device identifier as done initially to know its new parent. We compare the efficiency of Torso against some other common key setup protocols in terms of their corresponding energy costs. One of the simplest key setup protocols is pre-deployment of keys before the sensor nodes are put into active operation [5]. Once deployed, the nodes already share the cryptographic keys, and therefore the protocol only requires node authentication using a challenge-response scheme. Although this protocol has a minimum overhead, it raises scalability and security concerns especially for changing mission configurations. The security protocol is different for other usual security mechanisms as it takes into consideration the energy resources and is light weight. The key size of 64 bits should be sufficient to get the required security of information. The other usual protocol is the *Kerberos*, but Kerberos requires that the server share a long-term explicit master key with every sensor node which is a potential drawback, especially for large networks. Torso architecture doesn't make any such assumptions. Torso Security mechanism also makes sure the base station assigns all the node ids. We also reduce the number of communication messages to establish authentication as just a single JOIN message can serve as both join and authenticating is done in a single message.

5. 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 [7,8,9,10,11]. 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.

6. 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 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 network components within the domain. Definition of domain at a network system which can be managed autonomously was presented. 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.

7. Proposed Mechanisms to Enhance Survivability

7.1 Management within 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.

7.2 Management within Domain

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.

8. 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 2, the domain manager of SA 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 ofV. 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 afeedback message back to the origin of the pushback message. For example, oncedomain 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.

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 3 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.

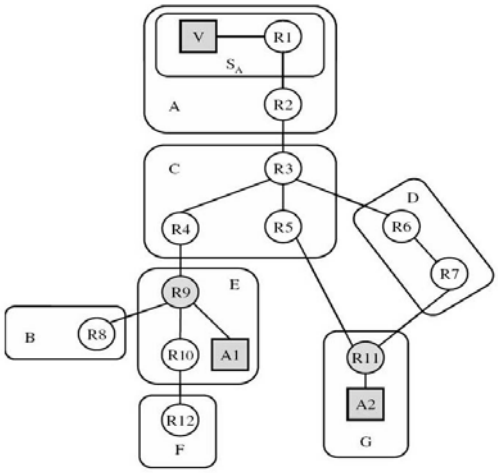

Figure 3. DDOS Attack Scenario [12]

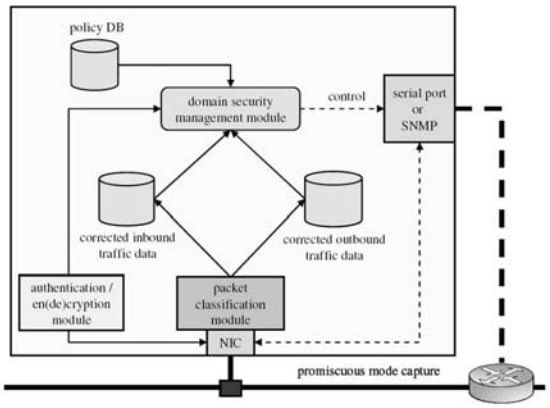

Figure 4. Structure of Domain Manager [12]

Figure 4 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. [12]

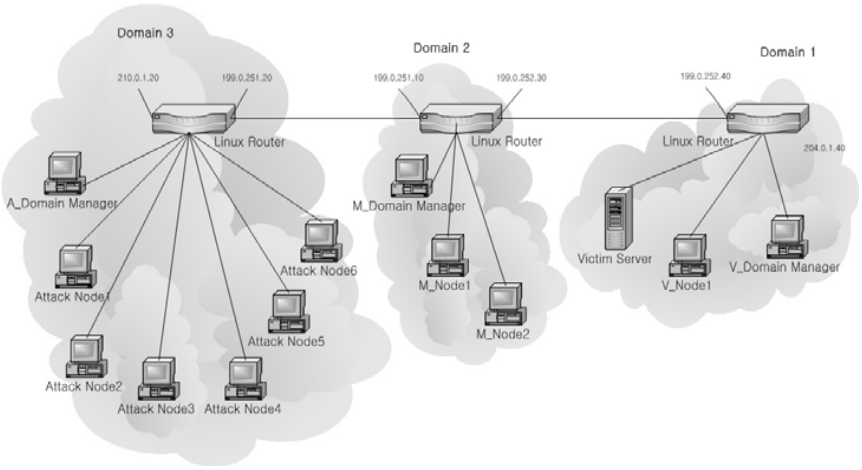

Figure 5. Test-bed System [12]

9. Conclusion

In this paper we have utilized the scenario in the existing research in [12]. Since large Scale networked information systems presents a multiple domains through which possibility of distributed denial of service (DDOS) attack is foreseen. In this paper a cooperative security management method and to elevate the service survivality to this large scale networked information system was presented.

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 [2] Jones, C.D., A.B. Smith, and E.F. Roberts, Book Title, Publisher, Location, Date.

[3] R. Colin Johnson, Companies test prototype wireless-sensor nets, EE Times (2003)

[4] L. Schweibert, S. K.S. Gupta, J. Weinmann, Research Challenges in Wireless Network of Biomedical Sensors, Proc. Of the 7th Annual International Conference on Mobile Computing and Networking (ACM SIGMOBILE) (2001) 151-165

[5] B.Woodward, M.F.A. Rasid, Wireless Telemedicine: The Next Step, Proc. Of the 4th Annual IEEE Conference on Information Technology Applications in Biomedicine (2003) 43-46

[6] P.Bauer, M.Sichitiu, R.istepanian, K.Premaratne, the Mobile Patient: Wireless Distributed Sensor Networks for Patient Monitoring and Care, Proc. Of the First Annual IEEE Conference on Information Technology Applications in Biomedicine (2000) 17-21

[7] KICS of Korea Information Security Agency : Intercept and Analysis Technologies Against DDoS Attacks, Sep(2004)

[8]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)

[9]. BGPExpert.com : How to Get Rid of Denial of Service Attacks, http://www.bgpexpert.com /antidos.php".

[10] Cisco : Unicast Reverse Path Forw -ding(uRPF) Enhancements for the ISP-ISP Edge, ftp://ft-eng.cisco.com/cons /isp/security/URPF-ISP.pdf.

[11]waterspring.org : Configuring BGP to Block Denial-of-Service Attacks, http://www.waterspring s.org/pub/id/draft-turk-bgp-dos-01.txt

[12] Sung Ki Kim, Byoung Joon Min, Jin Chul Jung, and Seung Hwan Yoo, "Cooperative Security Management Enhancing Survivability Against DDoS Attacks", ICCSA 2005, LNCS 3480, pp. 252 – 260, 2005.