Trust Based Approach to Detect and Prevent Distributed Denial of Service Attacks and Flash Crowds in VoIP Services

N. Jeyanthi [1] and N.Ch.S. N..Iyengar [2]
[1] *School of Information Technology and Engineering,*
[2] *School of Computing Science and Engineering,*
VIT University, Vellore - 632014, Tamilnadu, India
njeyanthi@vit.ac.in, nchsniyr@vit.ac.in

Abstract

Voice-over-IP (VoIP) is being widely used to packetize and relay voice information over existing IP networks. Due to the cheaper costs involved in using VoIP as opposed to traditional PSTN networks it is quickly gaining momentum and has seen widespread use in both personal and business domains. If this technology is to grow and gain over normal phone calls it must fulfill certain conditions that Public Switched Telephone Networks (PSTN) currently provide. Since VoIP uses the same routes used by normal Internet traffic, it is prone to a wide range of security threats, similar to those faced by normal packet data. These include, but are not limited to, Denial of Service (DoS), intruders in the network who either eavesdrop or are able to make free calls, man-in-the-middle attacks, etc. It thus becomes imperative that a protocol be developed that is able to both guard against and take corrective action in the event of the occurrence of any attack. In this paper we propose a protocol that detects and prevents the occurrence of a distributed Denial of Service (DDoS) attack. It also enforces security mechanisms to ensure that an attacker has not gained access to a user's password and thus making illegitimate calls. We develop a trust mechanism that can be used to detect an attacker in the network. The protocol has been rigorously examined through a wide range of experiments and the results have been found to be quite promising.

Keywords: *Voice over Internet Protocol, Denial of Service, Routing, Attack Detection, Attack Prevention, Trust.*

1. Introduction

Voice over IP grows exponentially than the expected growth, due to its cheaper cost and making use of the existing data networks. Soon it will replace the conventional Public Switched Telephone Networks (PSTN) and Wireless Phone services completely. It can not be achieved if it does not scale up its Quality to the level of PSTN.

Voice over IP works on multiple protocols for communicating between the end points. SIP-based VoIP uses, Session Initiation Protocol for establishing and terminating connections, Real-Time Protocol (RTP) is used for voice discharge. Similarly, VoIP works across different layers such as Transport layer and Application Layer. Hence it's susceptible to a range of attacks which is not limited to DoS. At the same time, discrimination between the DoS and Flash Events is also crucial.

1.1 Denial of Service

A Denial of Service attack aims to cripple the network or the server by flooding it with requests to the server. Due to the limit on the number of requests a server can handle, an excessively large number of simultaneous requests will result in the server being unable to handle them and therefore denying its services to legitimate requests. This constitutes a DoS attack in which legitimate users are denied access to the server's resources because the server is busy servicing malicious requests originating from the server.

An attack originating from distributed sources with each source sending a portion of the total number of requests leading to the denial of service attack is called a Distributed Denial of Service (DDoS) attack. A form of Dos attack is Physical attack, where the electrical power supply has been removed intentionally. Other types include the modification of configuration and authentication mechanisms.

1.2 Flash Crowds

Whenever a new website has been launched, there will be number of visitors with the eager to surf it. The server will be flooded with more number of calls at the same time. But they are all legitimate users, known as Flash Events. Their request should be considered and responded with in the finite amount of time. This attack also suspends the server activity.

1.3 Related Work

Many works deal with the detection of DoS attacks. VoIP being a real-time application, it sometimes becomes difficult to implement the more computationally intensive protocols due to it degrading the quality of service associated with VoIP.

Savage et al [1] describes a method in which the routers probabilistically mark packets with partial path information during packet forwarding. The victim then reconstructs the complete paths after receiving a modest number of packets that contain the marking. This is called the IP marking approach. The advantages of this method are that there is a low overhead for the routers and it allows for the incremental deployment of the network. But the server suffers from an extremely high overhead of having to reconstruct the original path information.

In the fragment marking scheme [20], there is a very high computation overhead for the destined node, 'victim, v', when the number of DoS attackers increases. At the same time the victim produces more false positives, during it's backtrack. This method is also susceptible to the generation of false paths due to the impersonation of compromised router as an uncompromised one while marking the fragments. The router which is in the original path, but not in the reconstructed graph, is known as false negative. A false positive is a router that is not in the real attack graph, but in the reconstructed attack graph.

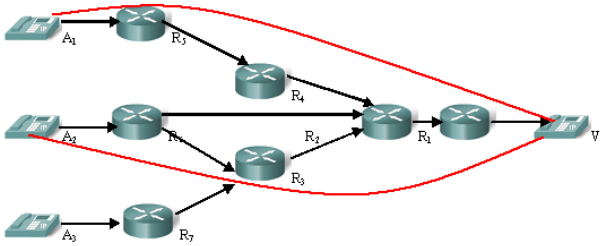


Figure 1. Upstream Router Map from Victim v

The edge sampling mechanism in turn writes details about the link into the packets. This approach receives information about two static fields such as the size of the IP address, start and end and a static distance in the field. The packets are marked by each router with a probability of q. For a particular packet if the marking probability is q, then the router stores its own IP address into the start field. Before updating the distance field, it checks for its value. If the distance field value is found to be zero, which means the fragment has been already marked by the previous router, it just copies its IP address in the distance field. Else it writes a Zero into the distance field. Thus link between itself and the previous routers are notified. Finally if the router doesn't mark the packet, then it always increments the distance field. This field indicates the number of routers the packet has traversed from the router which marked the packet to the victim.

In the fragment marking scheme, the router's IP address and redundancy information is split into 8 fragments and each router probabilistically marks the IP fragment with one of the eight fragments. This approach works well with a single attacker, but when multiple attackers are present as it is the case for a distributed denial of service attack, it suffers from the following problems:

- High computation overhead – The victim needs to check a large combination of fragments before it is able to reconstruct the path to the attacking node.
- This method leads to a large number of false positives because redundancy information check is insufficient.
- This method is not robust against a compromised router

The Advance Marking Scheme [20] proposes to encode the IP address of the router as a hash value. Let us assume that the victim has a map (Figure 1) of its upstream routers. The 16 bit IP field is divided into 11 bit edge field and 5 bit distance field as shown in fig.2. 5 bits can represent 32 hops which is sufficient for almost all Internet paths.

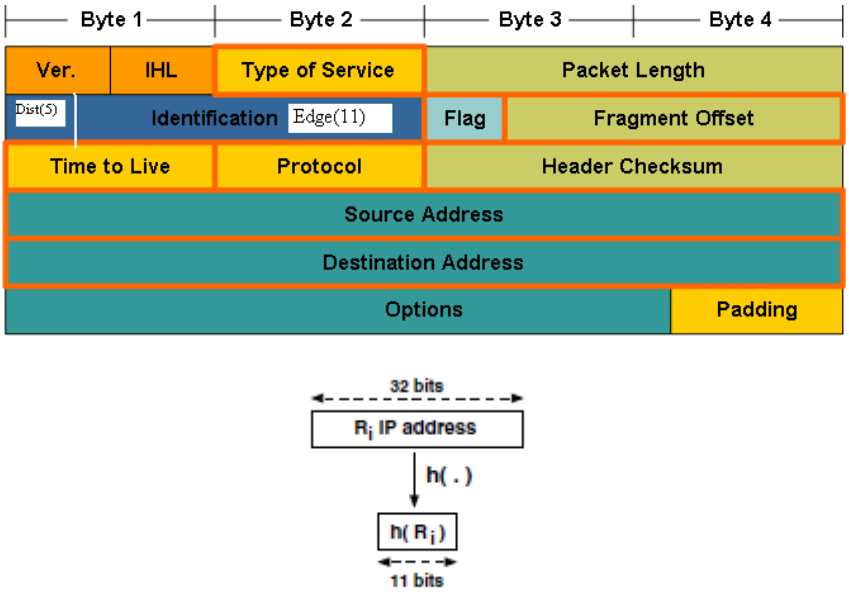


Figure 2 Encoding in Advanced Marking Scheme

In this method, two independent hash functions, h and h0,are used in the encoding of the routers' IP addresses. h and h0 both have 11-bit outputs. Every router marks a packet with a probability q when forwarding the packet. If a router Ri decides to mark the packet P, it writes h(Ri) into the edge field and 0 into the distance field in packet P. Otherwise, if the distance field is 0 which implies its previous router has marked the packet, it XORs h0(Ri) with the edge field value and overwrites the edge field with the result of the XOR. The router always increments the distance field if it decides not to mark the packet. The XOR of two neighboring routers encode the edge between the two routers of the upstream router map. The edge field of the marking will contain the XOR result of two neighboring routers, except for samples from routers one hop away from the victim. Because a xor b xor a = b; we could start from markings from the routers one hop away from the victim, and then hop-by-hop, decode the previous routers, as shown in figure 3. The reason to use two independent hash functions is to distinguish the order of the two routers in the XOR result.

Kopsidas, Zisiadis, and Tassiulas propose a method to secure a VoIP network called VIPSec (Voice Interactive Personalized Security). The VIPSec based application [21] comprises a rich set of network and security information elements. Additionally, the implementation achieves a high degree of user interactivity. It is an architecture design focusing on the network data model. In the implementation, users provide all relevant personal information like name, e-mail etc in order to register to a central database, which in turn provides directory services to the users of the service using their e-mail address or nickname as a search key. The central server provides information to the initiating client regarding called party's address. The client in turn connects directly to the other client over the Internet. After a successful VIPSec handshaking process, using random numbers as exchanged objects, the users are able to communicate securely.

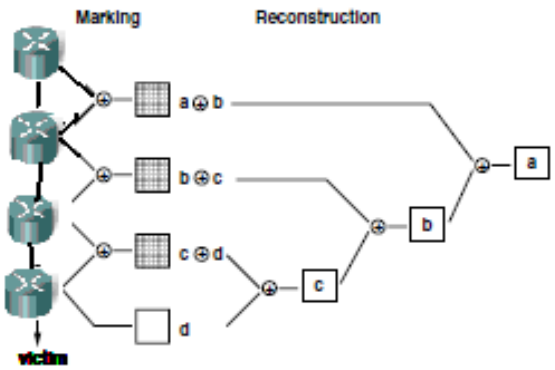


Fig. 3. XOR Encoding

Fig. 4 illustrates the communication procedure between two users. In the first step, Host A, and Host B sign into the directory server. A queries the server for B's IP address, and calls B directly. Next, a VIPSec handshaking procedure takes place in order to secure the communication channel. Finally, A and B confirm the numbers they sent to each other. If the confirmation is successful, then the key exchange has been completed successfully as well. As a result, the exchanged symmetric key is safe to be used for a secure channel establishment.

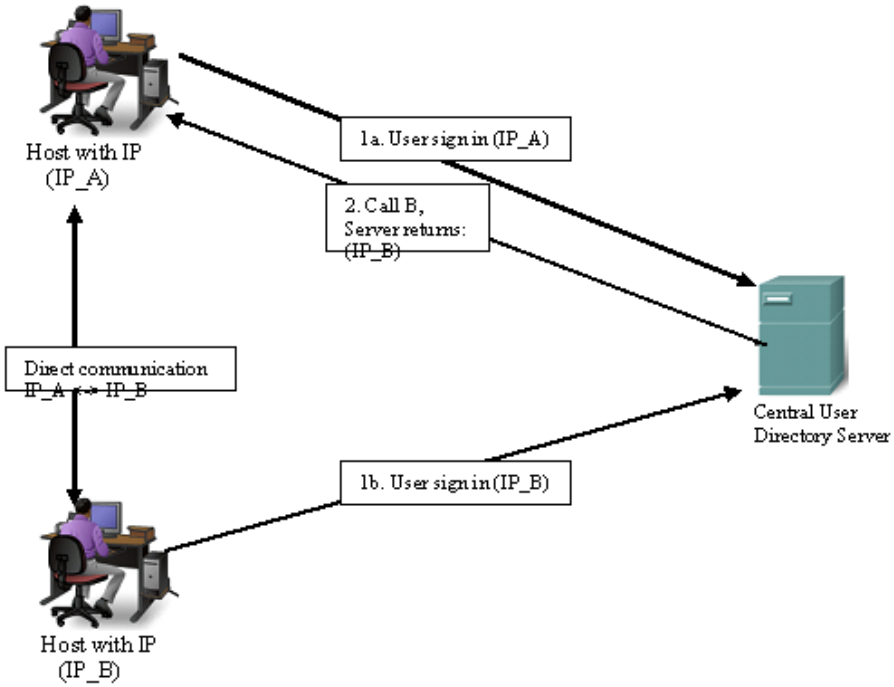


Figure.4. Communication Model of the VIPSec Based Application

In case that the confirmation of the numbers is unsuccessful, communication channel is considered to be compromised. The design and implementation of a VIPSec based application meet certain requirements, since there are specific procedures that should take place. The application architecture is depicted in fig 5.

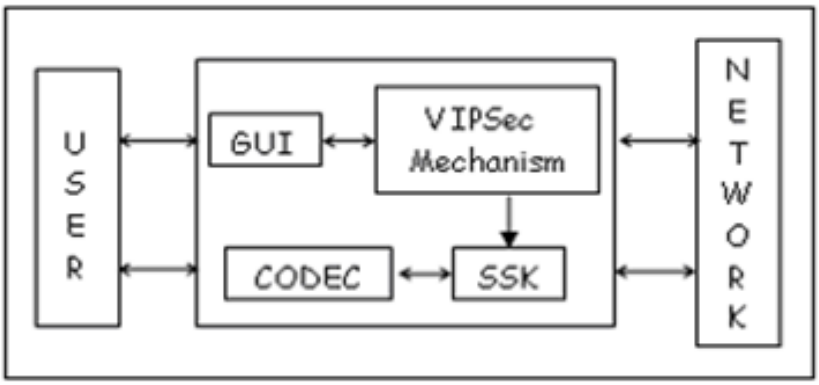


Figure.5. VIPSec Based Application Architecture

The covariance analysis model detects the attack on a statistics-based method [3], but the attacker will not be static and will not use the same IP address always. Different detection mechanisms were discussed by several authors [4-7] and a comparative analysis has been done in

2. The System Model

2.1 The Network Model

Voice information is sent over traditional IP networks. The VoIP server receives requests for making a voice call from clients. This request is routed over a normal IP network that forms the basis of Internet traffic. The topology of the network plays no role in determining the functioning of the protocol. When traffic arrives at a node, it's objective is to sample the packets in the event of an attack and if that packet is deemed to be coming from an attacker, it is dropped. If the traffic is legitimate it is passed on to the next router along the path. Sampling all the packets that arrive at a node puts an excessive load on any node. To prevent this, the packets are probabilistically sampled and only a portion of the malicious packets may be detected. The following nodes along the path continue to sample the packets and repeat the function of dropping packets from an attacker.

2.2 Preventing an Attack

As VoIP uses IP networks in relaying voice information, it is prone to a multitude of attacks from various sources and by widely differing methods. But due to the real-time nature of voice communication which puts constraints on more powerful techniques to identify and root out malicious entities in the network, a simpler solution which does not unduly overload the network and thereby maintaining a Quality of Service, while at the same time reducing and preventing the occurrence of a denial of service attack is necessary.

In a Denial of Service attack, an attacker sends multiple requests to the server. If the number of requests crosses a certain threshold, the server is unable to handle the requests and experiences downtime. In a real-time application like VoIP, it is very impractical if the server goes down due to an attack. Our work aims to develop an effective mechanism to detect and prevent an attack of this sort while at the same time maintaining a quality of service that is within acceptable standards.

This protocol borrows from the concept of packet sampling. This approach involves the scanning header information from packets that arrive at a node and based on some information that is passed to the node from the protocol determining if the packet is arriving from an attacker. Since it is impractical to sample all the packets that arrive at a given node, only a portion of the packets are sampled to determine whether they are malicious or not. If a packet is deemed to be malicious, the packet is simply dropped by that node and is not forwarded. Since all the packets cannot be sampled by one node alone, the load of detecting malicious information is distributed across the network. The packet sampling approach is not unique to our work and has been used before in various security mechanisms.

2.3 Detecting an Attacker

The concept of trust is introduced at the server to identify any potential attackers. The server consists of a five tuple consisting of IP address, port number, username, requests and trust. The IP address and port number are those of the client that sends the requests to the server. For every request that a client sends to the server an entry is made with the server, figure 6. The username that is used to authenticate the client is also stored along with the IP address and port number.

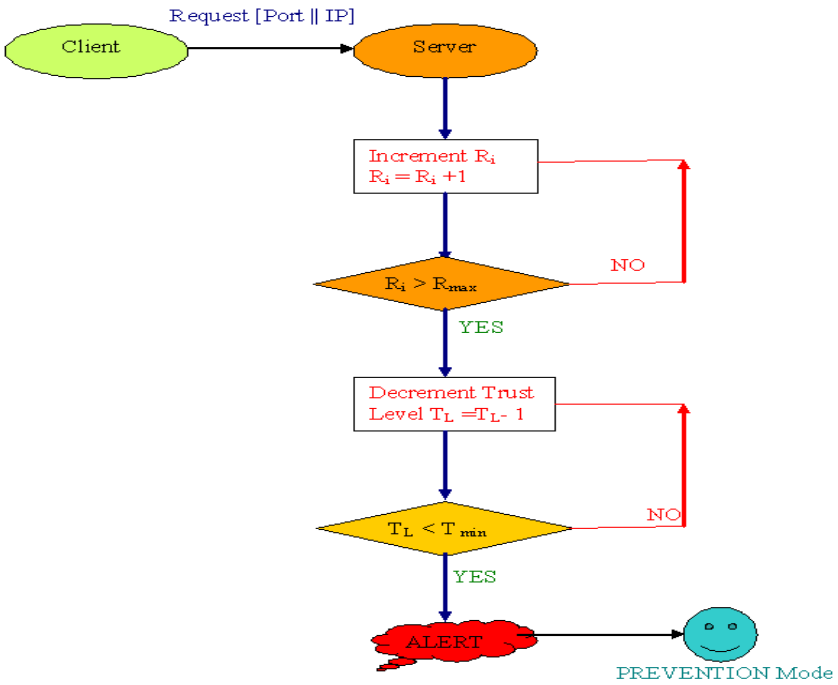


Figure 6. Alerting to Prevention Mode upon Detection of an Attacker

For every request, *i*, that the server gets from this triple, the requests field, R_i, corresponding to this triple is incremented. If the value of the requests exceeds a certain pre-defined threshold value, the trust, T **,** that is associated with that tuple is decremented (1).

$$R_i = \begin{cases} R_{i+1} \; ; & 1 \le i \le R_{\max} \\ T(L) \; ; & i > R_{\max} \end{cases} \qquad (1)$$

Where $T(L)$ is defined in equ.(4). The number of requests that a client issues to the server is not enough to determine exclusively that the client is performing an attack on the server. The trust field is used as a buffer in case the client is genuinely making extra requests due to a busy schedule, known as Flash Events. Therefore, for each time interval that the client exceeds the request threshold value, the trust associated with that triple is decremented.

If the trust decreases below the minimum trust required for any triple, that IP address is deemed to be an attacker. The server then issues an ALERT message to all the other nodes in the network. This message is unique to our protocol and contains information that the nodes require to identify an attacker or in the case of a distributed attack, it is capable of identifying multiple attackers. With this information, the nodes go into an prevention mode. The nodes then start sampling packets that arrive to detect if it comes from the IP address of the attacker. If the packets are deemed to be malicious, the packets are dropped immediately and no further forwarding of the packet is done.

But as mentioned earlier, the nodes cannot sample all the packets that traverse through it. To enforce this, we associate a budget with each node. For each packet that is sampled, there is some load on the node. As the number of packets, **n**, sampled by the node, $\mathbf{N_i}$, increases it reaches the maximum amount of packets, $\mathbf{n}_{\mathbf{max}}$, that the node is allowed to sample,**S**, in a particular time interval, **t**. Once this limit is reached, the nodes cannot sample any more

packets that arrive and it simply forwards the packet to the next node along the route. If this node has any remaining budget it will continue probabilistically sampling the packets till it reaches its limit. In this way the load of detecting any malicious packet is distributed across the network as given in (2).

$$S_{Ni} = \begin{cases} N_i \;\; ; \;\; 1 \le n \le n_{\max} \\ N_{i+1} \; ; \;\; otherwise \end{cases} \tag{2}$$

The probability of sampling the packets, $P_n(S)$, that arrive at a node can be varied depending on whether the node is in the path of an attacker or not, as is in (3),

$$P_n(S) = \begin{cases} P_n(S)+1 \;\; ; \;\; if \;\; n \in n_a \\ P_n(S)-1 \; ; \;\; if \;\; n \in n_a \end{cases} \tag{3}$$

Where $\mathbf{n_a}$ is the attacker node, **n** is the node in the traversing path.

If the node is in the path of an attacker, then most of the packets that arrive at it are bound to be from the attacker. It thus makes sense to increase the probability of sampling at that node. But if the node does not fall under a path that the attacker packets reach the server, it is not necessary for the node to continue sampling packets at an extremely high rate, thereby wasting resources. In this case the probability of sampling the packets is decreased to a more acceptable level.

When a new user is created, he is assigned a trust level, **T(L)**. Depending on the number of requests**, Ri**, he makes to the server, the trust level associated with him is increased or decreased. If the user makes a large number of requests such that there is reason to suspect an attack, his trust level is decreased. If the user continues to operate in a manner that falls within acceptable standards, his trust level is increased. Of course, this makes it possible that a user can start a long term attack that involves slowly building up trust and once enough trust is acquired, he launches an attack. Due to the large trust value, it takes time for the attack to be detected while the trust decreases to below the minimum, $T(L)_{min,}$ required before an attack is deemed to be taking place. To prevent this from occurring, a maximum trust value is established beyond which the trust of any user never increases.

$$T(L) = \begin{cases} T(L)-1 \;\; ; \; if \;\; R_i < R_{\max} \\ T(L)+1 \;\; ; \; if \;\; R_i > R_{\max} \\ T(L) \;\;\;\;\; ; \; if \;\; T(L) < T(L)_{\min} \end{cases} \tag{4}$$

Since this protocol involves storing client information including IP addresses for certain time periods, it becomes easy to incorporate an authentication mechanism for each user. In case an attacker gets hold of a user's id and password, he can generate an attack or even simply make free phone calls at that user's expense. To prevent this from happening, whenever a user makes a request to the server, the server checks the IP address from which the request is generated. If the IP address is not similar to those that the client normally issues requests from, there is cause for suspicion. In this case, the server does not immediately authorize the request, but asks for certain additional information from the user to authenticate him.

If the user is unable to provide the information needed to validate the request, no further requests are entertained from that user. The application running at the server sends a message to the original user informing him of the possibility that his account may be compromised. The server then requests for the additional information from the original user. Upon correct receipt of the validation information the server will enable the user to change his password or transfer his credit to a new account or another account of his choosing.

Two new packet formats have been introduced in our protocol. One is the ALERT packet that the server sends to the other nodes in the network depending on if it detects an attack. The ALERT packet contains information about the state of the attack, that is, whether an attack has just started or if the threat has been minimized due to the attack prevention mechanism employed by the nodes. It also contains information about the IP address and port number of host the server has deemed to be an attacker. This information is necessary for the nodes to sample and drop the packets. Only if the state of the attack is true will the nodes sample the packets.

The other packet developed is added by the client which contains user information and is used by the server to authenticate a user and also identify the user from which malicious requests are originating.

The receive ALERT algorithm is stated as follows:

PROCESS ALERT

An ALERT message can be of two types such that the server is on the verge of DoS and ther server has been recovered. In the earlier type, the server is flooded with large number of packets and on the verge of DoS, indicated by a value of 1.In the later type, server has been rescued form flooding, which is indicated by a value of 0. Whenever an ALERT message is received by a node:

1. Test out the value v (ALERT), present in the ALERT message.
2. If v (ALERT) = 1
   (i) Assign v (ALERT) =1, for the node
   (ii) Examine the attacker's IP address and port number from the ALERT message and write the attacker IP and port number to the attacker information that is maintained by each node...
3. Else, reset v (ALERT) = 0.
4. If the node has sent less than 3 ALERT packets with the same information, the node re-sends the same packet to its neighboring nodes thereby spreading the ALERT information to all the nodes in the network but not so far as to flood the network.

The SEND ALERT algorithm is issued by the server if an attack is underway or if it has just recovered from an attack. This is necessary so that the nodes know when to start and stop the packet sampling process.

SEND ALERT

The server sends an ALERT message with the value of 1 if it detects an attack and a value of zero when it has recovered from an attack.

```
1. If v (ALERT) = 1
   (i) Set the ALERT information in the ALERT message to 1.
   (ii) Write the attacker's IP and port number to the ALERT message.
2. Else v (ALERT) = 0.
3. Disseminate the message to its neighboring nodes.
```

3. Frame Formats

3.1 The Alert Packet

A new message format known as the Alert packet has been proposed (Fig.7). This packet is sent by the server to its neighboring nodes whenever an attack is occurring and the server is close to its capacity. The nodes that receive this packet then forward it to the other nodes in the network.

This message contains information about the state of the attack that is whether the attack has just started or if the server has recovered from the attack. This is specified by the AlertState field of the packet. The server also sends information about the IP address and port number of the attacker. Once a node receives this packet, it adds the attacker information to a table that it maintains which consists of all attackers IPs. When a node receives a packet from one of the attackers, it immediately drops the packet and no forwarding of the packets is done.

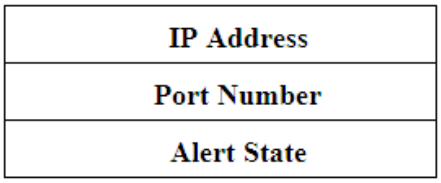


Figure 7. Alert Packet Header

3.2 The User Information Packet

Since a VoIP server requires each client to create an account before it starts to access the VoIP service, each user has a unique user identification number (Fig. 8). Whenever a client makes a request, the client application adds its own header information to the packet which will enable the server to uniquely identify each client.

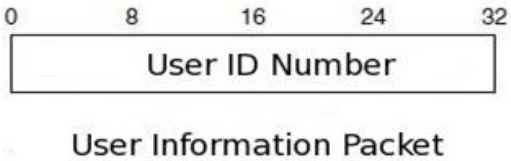


Figure 8. User Information Packet

A new message format has been introduced which will enable a client to send user information. The application at the server end is capable to deciphering this information and identify if a single user is causing a denial of service account from multiple IPs. This also helps to authenticate the user if his account has been hacked and requests are coming from IP addresses which are different from that normally used by the client to access the VoIP service.

4. Experimental Evaluation

The proposed protocol was simulated using the NS3 simulator. Two sets of experiments were conducted to assess the performance of our protocol. The parameters used in the simulation are depicted in Table 1 followed by the experimental values in Table 2.

Table 1 Simulation parameters used in NS3.

Parameters	Value
Channel Helper	Yans Wifi channel Helper
Channel Propagation Delay	Ns3:Constant Propagation Delay Model
Physical medium helper	Yans Wifi channel Helper:: Default
MAC type	Adhoc Wifi MAC
Position allocator	Ns3::GridPositionAllocator
Min X	0
Min Y	0
Delta X	120
Delta Y	120
Grid Width	5
Layout type	Row First
Mobility Model	Ns3::Static Mobility Model

Table 2 Parameters for Experiment 1.

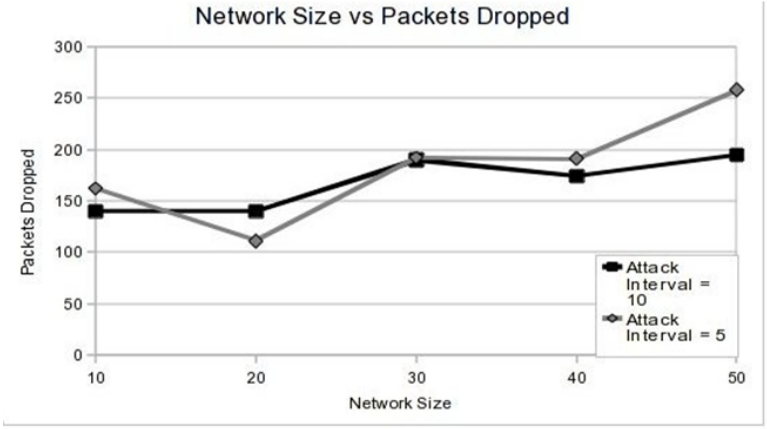
Parameters	**Value**
Initial Trust	2
Max Trust	5
Min Trust	0
Max Requests	10
Attackers	2
Packets Sent by Attackers	400
Attack Interval	0.4s

4.1 Varying the Network Size

In this experiment, we studied the performance of the protocol by varying the network size. The figure shows the number of packets that were dropped by the network. As the size of the network increases, it can be seen that more packets are dropped on an average. This is due to the fact that since a budget is associated with each node, the smaller networks stop sampling packets after a while due to them crossing their budget. Due to this, many attacker packets can find their way to the server. But if the number of nodes in the network increases, even if the budget of earlier nodes is crossed, the nodes that come later in the path to the server can continue sampling the packets. By the time these nodes exhaust their budget, the timer would have expired and the earlier nodes will be able to sample the packets once again. In this way, the performance of VoIP Dos increases as the size of the network increases. For example, it can be seen from the graph that when the number of nodes in the network is 10, about 140 packets are only dropped. But as the nodes increase in number from 10 to 30, we

find that the number of packets that are dropped rises to 190 and a 50 network node drops about 195 packets.

Another interesting fact to note from the graph is the effect of the timer interval value. If the interval value is decreased, we find that on an average, more packets are detected and dropped. This is due to the fact that a smaller attack interval causes many packets to be sent to the server, probably exceeding the server capacity for the attack interval. By reducing the attack interval, it is possible to detect an attack earlier, since the number of packets received in that interval is smaller and if it exceeds the maximum number of requests allowed per time interval, we immediately can alert the network to the possibility of an attack. For example, for the same network size of 50 nodes, a timer interval of 10 is able to detect and drop 195 packets, whereas with a timer interval of 5; 258 packets are detected and dropped.

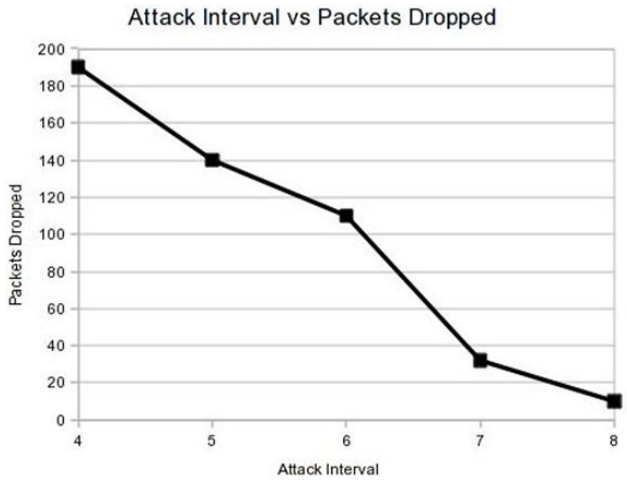


We cannot always have a very small interval value, since that would put an unnecessary load on the server, which can increase the chances of a denial of service. Therefore a compromise has to be struck when it comes to setting the timer interval value.

4.2 Effect of Attack Interval on the Occurrence of an Attack

The more time the attacker takes to send packets, the easier it is for the server to recover from such attacks. In some cases a denial of service may not even occur and hence no packets will be dropped. This happens in the case when (timer interval/attack interval) < maxRequests, where timer interval is the frequency at which the server checks for an attack, attack interval is the frequency with which the attacker is sending packets, and maxRequests is the maximum number of requests the server can handle in the timer interval.

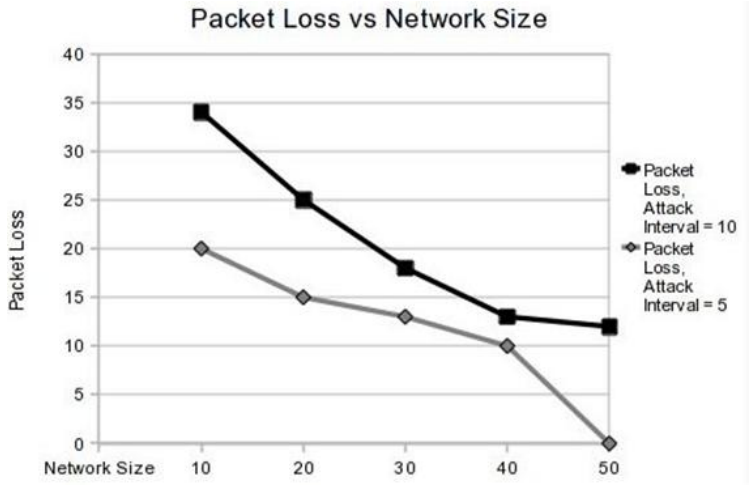
The figure below clearly shows that the slower the attacker sends his packets, the less there are chances for a denial of service to occur, thereby decreasing the number of packets that are to be dropped. From this we see that to successfully simulate a DDoS attack, it is not enough that a large number of packets be sent, but also that they are sent in quick succession.



4.3 Quality of Service – Packet Loss against Network Size

The efficiency of any protocol can be gauged only by considering its quality of service. The packets lost as a result of server downtime seriously degrades the quality of service and as a result the protocol must ensure that it guards against packet loss. In case of VoIP, these lost packets can reduce the quality of audio signals from one caller to another. We plot the packet loss against network size for various timer intervals for our protocol. We find that the overall packet loss due to server downtime can be reduced by reducing the timer intervals. This causes the server to re-investigate the trust levels of the various clients much more frequently. But if we reduce the timer interval to a shorter duration, the server spends a considerable amount of its resources checking for an attack. Therefore, we need to optimize the attack interval so that packet loss can be minimized, while at the same time ensuring that the server does not spend much time detecting an attack.

Another interesting fact that can be noted from the graph below is that, as the size of the network increases, the packet loss due to server downtime decreases. This is due to the fact that many more nodes scan and drop packets which results in an overall decrease in the number of malicious packets that arrive at the server. Therefore as the size of the network increases, we see an improvement in teh quality of service. This makes it very suitable for larger networks consisting of many nodes across the network.



5. Conclusions

Table 3 : Comparative Analysis of Detection Techniques

Detection Technique	Type of Attack	Test Data	**Detection Results**
Activity Profiling	TCP, UDP flood	6 publicly available data	**12,000 DoS attacks on 5,000 distinct victims**
Change-point detection	TCP SYNC flood attack – Constant rate	3 private network data sets	**All attacks detected**
Wavelet Analysis	DoS floods		**90% anomalies detected**
Flow-based anomaly detection	BYE DDoS, RTP Traffic	monitoring traffic IPFIX 1/10Gbps	-
Entropy based Traffic mode Detection	DoS/DDoS	sample data from Lincoln Lab	**99.2% of TCP SYN flooding detected**
Volume based Detection Techniques	DoS		**Unable to detect Short-term attacks**
Distance based Detection technique	Flooding based DDoS attacks	NS-2	**Hard to exploit in Real situation**
Network Traffic Based Detection	DDoS	Data from Zaozhuang University with Sniffer Software	**1 out of 2 attacks were detected**
Covariance Analysis Model	SYN Flooding, DDoS	NS-2	**500 SYN packets/sec**
Integrated Victim-based Approach	DoS attacks	NS to simulate a wired IP network	-
TRUST BASED	**DDoS Attacks, Flash crowds**	**NS-3**	**99% anomalies detected**

In this paper the proposed protocol has a method to prevent the occurrence of a Denial of Service Attack and Flash Crowds in a VoIP service. The concepts of sampling and packet dropping along with a trust field are incorporated to optimizing the identification of a real threat and decreasing the possibility of false positives. Two new frame formats such as ALERT packet and User Identification packet were suggested to facilitate the functioning of our protocol. Our experiments have been performed using NS-3 to test the Normal and Flooding behaviour of the VoIP network. Our results have been found to be satisfactory in preventing a DDoS attack. Table 3 presents a comparative view of the performance analysis.

Future enhancement can be of implementing and testing in a real life environment with a large network and a more widely distributed attack.

Acknowledgements

The authors are thankful to the VIT University management, Director of SITE and Director (Academic Research) for their encouragement. They also wish to express their heart felt gratitude to the DC members Dr.P.S.Avadhani (Professor, Department of CS&SE, Andhra University, Visakhapatnam), Dr.V.Nagalakshmi (Professor, Department of CS, GITAM University, Visakhapatnam) and Dr. Balakrushna Tripathy (Senior Professor, SCSE, VIT University, Vellore) for their valuable suggestions.

References

[1] Stefan Savage, David Wetherall, Anna Karlin, and Tom Anderson, "Practical network support for IP Traceback," in Proceedings of the 2000 ACM SIGCOMM Conference, August 2000
[2] Bin Xiaoa, Wei Chenb, Yanxiang Hec "An autonomous defense against SYN flooding attacks: Detect and throttle attacks at the victim side independently", Journal of Parallel and Distributing computing, 2008.
[3] Shuyuan Jin, Daniel S. Yeung, "A Covariance Analysis Model for DDoS Attack Detection",IEEE communications society, 2004.
[4] Glenn Carl and George Kesidis, Richard R. Brooks, suresh Rai,"Denial-of-Service Attack-Detection Techniques", IEEE Internet Computing, 2006, pp82-89.
[5] Hemant Sengar, Haining Wang, Sushil Jajodia , "Fast Detection of Denial-Of-Service Attacks on IP Telephony", IEEE, 2006,pp199-207.
[6] Yonghua You, Md.Zulkernine,Anwar haque , "Detecting Flooding-Based DDoS Attacks", IEEE Communication Society, 2007, pp.1229-1234.
[7] Muhali li, Ming Li, Xiuying Jiang, "DDoS Attacks Detection Model and its Applications",WSEAS Transctions on computers,Vol.7, 2008, pp.1159-1168.
[8] Mohamed Nassar, Saverio Niccolini, Radu State, Thilo Ewald ," Holistic VoIP intrusion detection and prevention system", In the ACM Proceedings of the 1st international conference on Principles, systems and applications of IP telecommunications, 2007.
[9] Sven Ehlert, Yacine Rebahi, Thomas Magedanz, "Intrusion Detection System for Denial-of-Service flooding attacks in SIP communication networks, International Journal of Security and Networks , Volume 4 Issue 3, 2009.
[10] Tsz-Yeung Wong, Man-Hon Wong, and Chi-Shing (John) Lui" A Precise Termination Condition of the Probabilistic Packet Marking Algorithm", IEEE Transactions On Dependable And Secure Computing, Vol. 5, No. 1, January-March 2008.
[11] D. K. Y. Yau, J. C. S. Lui, F. Liang, and Y. Yeung, "Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles," IEEE/ACM Transactions on Net-working, vol. 13, no. 1, pp. 29-42, Feb. 2005.
[12] 2008Li Li, SHEN Su-bin, "Packet track and traceback mechanism against denial of service attacks", in Science Direct, 15(3):51-58, September 2007.
[13] Stefan Savage, David Wetherall, *Member, IEEE*, Anna Karlin, and Tom Anderson "Network Support for IP Traceback", IEEE/ACM Transactions On Networking, Vol. 9, No. 3, June 2001.
[14] P. Ferguson and D. Senie, "Network ingress filtering: Defeating denial of service attacks which employ ip source address spoofing," RFC 2267, January 1998.
[15] Z. Gao, N. Ansari, "A Practical and Robust Inter-Domain Marking Scheme for IP Traceback", Computer Networks 51 (3), 2007,pp. 732-750.
[16] T. Ho, R. Koetter, M. Medard, M. Eros, J. Shi, D. Karger, "A Random Linear Network Coding Approach to Multicast", IEEE Transactions on Information Theory 52 (10), 2006, pp.4413-4430.
[17] J. Mirkovic, S. Dietrich, D. Dittrich, P.Reiher, "Internet Denial of Service: Attack and Defense Mechanisms", Prentice Hall Professional Technical Reference, 2004.
[18] O. Spatscheck and L. Peterson. "Defending Against Denial of Service Attacks in Scout". In proceedings of the USENIX/ACM Symposium on Operating System Design and Implementation, 1999
[19] D. Dean, M. Franklin, and A. Stubblefield, "An Algebraic Approach to IP Traceback," ACM Trans. Information and System Security, vol. 5, no. 2, pp. 119-137, 2002.
[20] D.X. Song and A. Perrig, "Advanced and Authenticated Marking Schemes for IP Traceback," Proc. IEEE INFOCOM '01, pp. 878-886, Apr. 2001.
[21] Leandros Tassiulas, "Design and Implementation of a VIPSec Based Application", Fourth Annual International Conference on Mobile and Ubiquitous Systems Networking & Services (MobiQuitous), 2007

[21] K. Park and H. Lee, "On the Effectiveness of Probabilistic Packet Marking for IP Traceback under Denial-of-Service Attacks," Proc. IEEE INFOCOM '01, pp. 338-347, 2001.
[22] NS3: Network Simulator 3, http://www.nsnam.org/

Authors

Prof. N Jeyanthi is a faculty cum Research Scholar in VIT University. She received her M.Tech in Information Technology with Networking as Specialization from VIT University, India in 2006 and B.E. in Computer Science and Engineering from Madurai Kamaraj University, Madurai, India in 1999. Her current research interest is on Network Security in Real-Time applications. A life member of Indian Society of Technical Education.

Dr.N.Ch.S.N. Iyengar (M.Sc,M.E,Ph.D) is a Senior Professor at the School of Computing Science and Engineering at VIT University, Vellore, Tamil Nadu, India. His research interests include Agent based Distributed Computing, Cryptography and Network Security, Intelligent computing and Bio informatics. He has authored several textbooks and had more than 100 research Publications in International Journals. He chaired many IEEE international conferences and delivered invited/ technical lectures/ keynote addresses besides being International programme committee member.