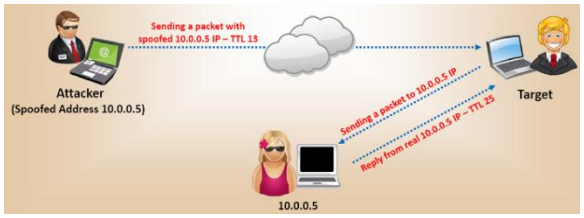
Internal Self IP Trace back System for tracing IP Spoofing inNetworks

VURITI SIREESHA[1] P.K.SAHU[2]
[1](Final Year M.Tech Student,Dept. of CSE, Aditya Institute of Technology and Management(AITAM), Tekkali, Srikakulam, Andhra Pradesh , vsireesha.vuriti@gmail.com)
[2](Assoc Professor, Dept. of CSE, Aditya Institute of Technology and Management (AITAM), Tekkali, Srikakulam, Andhra Pradesh,promod_sahu@yahoo.com)

ABSTRACT:

The efficient way of communication from source and destination networks in hybrid system is always a challenge since years. One of the existing approach is whenever the data loss over (attacks and spoofing) the networks the path will be re established for the repetitive communications. This is totally burden to the hybrid networking system in lot of perspectives like time and router frame work. We know that none of the peers does not have direct interaction. Due to frequent spoofing among hybrid networks our approach is to do check the best feasible path internally by router before transmitting from source to destination. The router will deliver the packets to destination router at the destination network in hybrid system. So the time will be saved for transmission instead of reconstructing the whole path. Here the router will take an active part to send the information about the sender to the destination router. This is to find out whether the packets are coming from trusted/valid source.

Once the packets are received by the router which is having the best feasible internal path in their routing tables, So the packets will be discovered / delivered to the right destination. The target router will maintain the log information for all the communications so that if any spoofing is found the transmission will be repeated only from source router (which is having the shadow of the transmission



Introduction:

IP spoofing is a technique used to gain unauthorized access to computers, where by the attacker sends messages to a computer with a forging IP address indicating that the message is coming from a trusted host.

Attacker puts an internal, or trusted, IP address as its source. The access control device sees the IP address as trusted and lets it through.

Related Work:

In Previous paper we have discussed about the IP Trace back system which having the capable of identifying the spoofing in network over the transmission of packets. Here in any transmission, if there is any attack or spoofing was occurred the routers were alerted to resolve the problem with

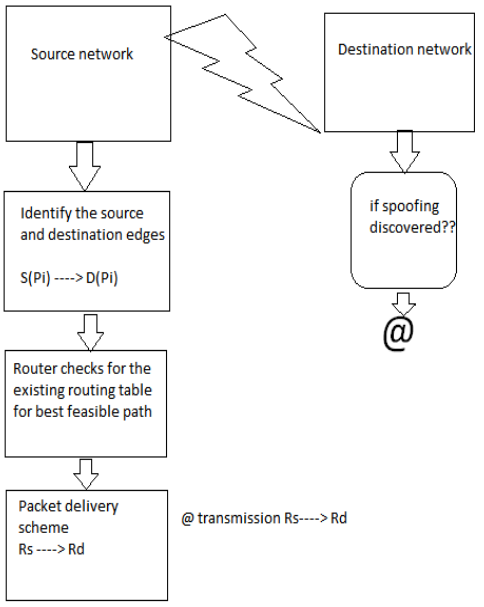
Attack and reconstruct the paths to change the direction to transmit the packets .To overcome this by using path reconstruction or re establishment for repetitive

communications. It is also not a good solution. But it is time taken to routers framework.

To overcome the Problems in Previous work we have proposed a best schema that takes less time and show best performance in resolving the problems from attackers. So herepeers in the Hybrid network

Do not have a direct path. Due to this router have to maintain best feasible path internally in between source and destination. Packets delivery is done by the router to destination router so time is saved. No need of again path reconstruction of new path.

System Architecture:



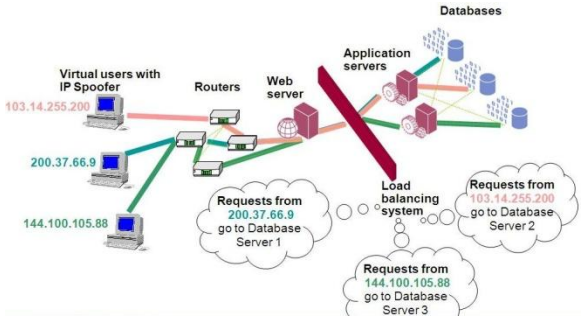
System Model:

In Hybrid Networks, normally the communication will be from source network to Destination network. The source network and Destination networks are of two different and unknown infrastructures to each other.

Two different networks are connected with Routers. The packet Transmission will be done through these Routers. In this system every router has to be monitored with itself while transmission. Theestablishingmonitors dedicated at spoofed/untrusted routers. Monitors evaluate the paths and update the hash routing table for trusted paths. The Threshold will be maintained once the monitor evaluates the entire path. so it suppress reconstruction. Somonitoringhas feasible paths for trusted communication from source to destination. Casting buffers are maintained only at centralized routers.

Transmission may be regular and also casting. Single are multi casting is always accountable for MANET monitors.Casting is useful in Security Maintenance in Regular Transmission.

The MANET will always maintains buffered or Storage nodes if casting the destination node is not in receiving state .so buffer or storage node will maintain the data to be transmitted in post communication. Buffer log will be maintain by monitor and monitor will evaluating on best path which is identified by monitor for regular transmission fluctuations..



Routing Tables:

Routers have to be monitored while transmission of packets from source to destinations. Each router have maintained with following tables.

Nodes Information:

Number of nodes	Nodes IP Addresses	Rate of Transmission
1	192.168.0.102	100mbps
2	192.168.0.100	80mbps

Log Tables:

At Destination Side we have to maintain Log tables for identifying the Transmission weather it was trusted or not. Log Tables have the fields like below. By having Log tables we can know the complete information regarding Source Information, PacketInformation, Traveling path Information.

Source	Traveled Path	Packet Size	Time of Delivery

NOTATIONS:-

1) M $\leftarrow \sum_{0}^{n-1} m$(M-> available peers(n) in the manet –M)
2) L_i <- level of i
3) $W_{(i,j)} \leftarrow$ weight of the path between I and j
4) t ← type of protocol
   a. $t_t \rightarrow$ tcp
   b. $t_u \rightarrow$ udp
5) $p_i \leftarrow$ packet buffer
6) L $\epsilon (w_i, j)$
7) $R_t \leftarrow$route stack [input n(peers)

SourceAddress	Destination	Feasible Paths	Protocol	Payload of transmission
192.168.0.102	192.168.0.100	6	udp	30mb
. .	. .	. .	. .	. .

output is manet M]

8) H←hash function

Framing MANET:

INTILIZATION:-

step 1: M← 0

(initially no MANET, no node so count =0)

count←0

step 2: loop: for each nodes in m

M⇐m_i

(if node is present inMANET increase count by 1)

count++

Nodes at(M, m_i)

$W_{(I, i+1)}$**← weight (l)**

(weight on the path I,i+1 is W(I,i+1),weight(l)=weight on trusted path)

L ϵ $W_{(I, i+1)}$

(so each trusted path having some weight)

end loop

step 3: if $M \rightarrow t_t$

(if the MANET is using TCP, call TCP monitor)

call monitor (M, t_t)

Else**if**

$M \rightarrow t_u$

(if the MANET is using UDP, call UDP monitor)

call monitor (M, t_u)

Framing Monitor:

MONITOR:

input→M (MANET)

output→S_M (security monitor)

count←0 (count initialized to 0)

if M is Tt (if monitor is TCP)

start

for each peer have 0 to n-1

(one peer having 0 to n-1 nodes)

Loop: C ← cast (l,f)

(Every casting will noted by count)

O_S ← Source packet

$\sum_f^l Od$ ← Destination cast peer

(l,f are nodes in casting)

R
t← c

(after packet is received by destination it is placed on routing stack Rt)

count++

(After receiving packet by destination count incremented)

end loop

M ϵ **S_M** ϵ **R_t**

(MANETs have security monitors security monitors have routing table)

Loop r ←0

for each r in R_t

H ←S_r ←O_s

H←S_r ($\sum_f^l Od$)

end loop

Framing Hash Function:

Loop r←0

For each r in Rt

(Each router information is stored in Rt)

H←Sr←Os

(Some router/original source has Hash function H)

H←Sr ($\sum_f^l Od$)

(That hash function is applied to all nodes from source Sr to
destination Od i.e., l to f all internal nodes)

End loop

Feasible Path construction:

F P C B T (with out spoofing) λ=65%

Feasible path construction before transmission

step 1:

for each l in H

(each path in the hashing table Sr-source router,Od-Original destination)

Loop: b1 $\Leftarrow S_{r(\sum_f^l Od)}$

(b1 – first best path finding path increment c)

$s \leftarrow S_r$

c= count ++

if trans (s,b1)≤λ

p←(b1,s)

(1 feasible path is found)

if b1, C $S_{r(\sum_f^l Od)}$

(if b1 is one of the best path in between Sr,Od)

$t \leftarrow S_{r(\sum_f^l Od)}$

(any connection is existing between Sr,Od)type of protocol

if trans (s,t)≥λ (

transfer packet from S to T if ≥λ

p←(s,t) (packet is send from source to destination)

count++ (count incremented)

end if

end if

SPOOFING TECHNIQUE:

BEFORE PATH CONSTRUCTION:

STEP1:

Router/server allots the unique IP vector and routing table.

STEP2:

Always the IPs of individual peers will be identified by router

STEP3:

These IPs are fixed before path reconstruction/finding of shortest path.

AFTER PATH CONSTRUCTION/ SHORTEST PATH:

STEP4:

Router may allot the new IPs and the IP vector will be updated.

STEP5:

And the IP vector updated the routing table updated

STEP6:

Now the server/router will check the ip vector and If router finds an IP (forged) then the spoofing alert has to be updated.

Experiment Results:

In our experimental environment, the prototype system can trace packets to their sources. For tracers in real-world environments, however, especially on backbone networks where network traffic is very high, more memory will likely be needed. Moreover, it is difficult to deploy

tracers and monitoring managers all over the Internet at the same time. One way to enable traceback in real-world environments is to introduce these components into one administrative domain (such as a corporate intranet) and enable traceback within that domain first. If the adjacent domain also introduces the tracing function, the domains can trace beyond their network boundaries by exchanging trace information between monitoring managers.

Conclusion:

IP spoofing is related to IP packet structure and hence it is a difficult problem to be tackling with. There are several ways for exploring IP packets. As we know that intruders or hackers hide their identity with IP spoofing and it will be a huge way back for them to make several attacks on network. In this paper we have provided some of the proactive and reactive methods at the nodes for the over whelming menace where there is no easy solution. We also used routers in the network to help detect a spoofed packet and trace it back to its originating source

References:

[1] S. Savage et. al. "Practical Network Support for IP **Traceback**," Proc. 2001 ACM SIGCOMM, vol. 30, no. 4, ACM Press, NewYork, Aug. 2001, pp. 295-306; available on line athttp://www.cs.washington.edu/homes/savage/traceback.html.

[2] S. Bellovin, M. Leech, and T. Tylor, "ICMP Traceback Messages," Internet draft, work in progress, Oct 2001; available online at http://www.ietf.org/internet-drafts/draft-ietf-itrace-01.txt

[3] R. Stone, "CenterTrack: An IP Overlay Network for Tracking DoS Floods," Proc. 9th Usenix Security Symposium, Usenix Association, Berkeley, California, Aug 2000; available online athttp://www.usenix.org/publications/library/proceedings/sec2000/stone.html

[4] H.Y. Chang et. al., "DecIdUous: Decentralized Source Identification for Network -Based Intrusions," Proc. 6th IFIP/IEEE International Symposium. Integrated Network Management, IEEE Comm. Soc., New York, May 1999, pp. 701-714.

[5] K. Ohtaet. al., "Detection, Defense, and Trackingof Internet Wide-Illegal Access in a distributed Manner," Proc., INET2000, Internet Society, Reston, VA, July 2000;

[6] CERT, "TCP SYN flooding and IP spoofing attacks," Advisory CA-96.21, September 1996.

[7] Vern Paxson, "An analysis of using reflectors for distributed denial-of-service attacks," Computer Communication Review, 31(3), 2001.

[8] Mike Kristovich, "Multi-vendor game server DDoS vulnerability,"

http://www.pivx.com/kristovich/adv/mk001/, November 2002.

[9] CERT, "IP spoofing attacks and hijacked terminal connections," Advisory CA-1995-01

www.cert.org/advisories/CA-1995-01.html, February 2001.

[10] L. Joncheray, "Simple active attack against TCP," www.insecure.org/stf/iphijack.txt , February 20

[11] ADLER, M. 2005. Trade-offs in probabilistic packet marking for IP traceback. J. ACM 52, 2, 217–244.

[12] ALBRIGHTSON, B., GARCIA-LUNA-ACEVES, J., AND BOYLE, J. 1994. EIGRP—A fast routing protocolbased on distance vectors. In Proceedings of the Networld/Interop.

[13] AURA, T. AND NIKANDER, P. 1997. Stateless connections. In Proceedings of the International Conferenceon Information and Communication Security, Y. Han et al., Eds. Lecture Notes in ComputerScience, vol. 1334. Springer, 87–97.

[14] BAKER, F. 1995. Requirements for IP Version 4 routers. RFC 1812.

[15] BAKER, F. AND SAVOLA, P. 2004. Ingress Filtering for Multihomed Networks. RFC 3704.

[16] B.Al-Duwari andM. Govindarasu, "Novel hybrid schemes employing packet marking and logging for IP traceback," *IEEE Trans. Parallel Distributed Syst.*, vol. 17, no. 5, pp. 403–418, May 2006.

[17] A. Appleby, Murmurhash 2010 [Online]. Available: http://sites.google.com/site/murmurhash /

[18] A. Belenky and N. Ansari, "IP traceback with deterministic packet marking," *IEEE Commun. Lett.*, vol. 7, no. 4, pp. 162–164, Apr. 2003.

[19] A. Belenky and N. Ansari, "Tracing multiple attackers with deterministic packetmarking (DPM)," in *Proc. IEEE PACRIM'03*, Victoria, BC, Canada, Aug. 2003, pp. 49–52.

[20] S. M. Bellovin, M. D. Leech, and T. Taylor, "ICMP traceback messages," *Internet Draft: Draft-Ietf-Itrace-04.Txt*, Feb. 2003.

[21] H. Burch and B. Cheswick, "Tracing anonymous packets to their approximate source," in *Proc. USENIX LISA 2000*, New Orleans, LA, Dec. 2000, pp. 319–327.

[22] CAIDA's Skitter Project CAIDA, 2010 [Online]. Available: http://www.caida.org/tools/skitter/

[23] K. H. Choi and H. K. Dai, "A marking scheme using Huffman codes for IP traceback," in *Proc. 7th Int. Symp. Parallel Architectures, Algorithms Networks (SPAN'04)*, Hong Kong, China, May 2004, pp. 421–428.

[24] C. Gong and K. Sarac, "A more practical approach for single-packet IP traceback using packet logging and marking," *IEEE Trans. Parallel Distributed Syst.*, vol. 19, no. 10, pp. 1310–1324, Oct. 2008.

[25] A. Hussain, J. Heidemann, and C. Papadopoulos, "A framework for classifying denial of service attacks," in *Proc. ACM SIGCOMM '03*, Karlsruhe, Germany, Aug. 2003, pp. 99–110.

[26] W. John and S. Tafvelin, "Analysis of internet backbone traffic and header anomalies observed," in *Proc. IMC '07: 7th ACM SIGCOMM Conf. Internet Measurement*, San Diego, CA, Oct. 2007, pp. 111–116.

[27] W. John and T. Olovsson, "Detection of malicious traffic on backbone links via packet header analysis," *Campus-Wide Inform. Syst.*, vol. 25, no. 5, pp. 342–358, 2008.