Detecting Malicious Packet Dropping Using Statistical Traffic Patterns

Julian Benadit.P[1], Sharmila Baskaran[2] and Ramya Taimanessamy[3]

**[1] Computer Science and Engineering, Pondicherry University,
Pondicherry, India**

**[2] Computer Science and Engineering, Pondicherry University,
Pondicherry, India**

**[3] Computer Science and Engineering, Pondicherry University
Pondicherry, India**

Abstract

Internet is a global network where it is easily prone to be attacked by hackers. Packet loss exhibits temporal dependency. Many approaches have been implemented to provide secure route for the packets sent and finding out malicious packets. In this paper, we use a protocol and maintain log at each router to find out where the loss actually occurred. Our paper mainly focuses on where the packet has dropped or attacked.

Keywords: *Internet dependability, distributed systems, reliable networks, malicious routers.*

1. Introduction

THIS document details the approach, methodology and results of recent experimentation for of detecting packet loss in a network. In this paper, we propose an *operationally viable* approach to find out where the loss occurred. If an attacker gains control over a router, he could disrupt the communication by dropping or manipulating the packets sent. Traffic can be severely disrupted by routers refusing to serve their advertised routes, announcing nonexistent routes, or simply failing to withdraw failed routes, as a result of either malfunction or malice. The key idea behind detecting malicious packet loss is finding where the packet loss has occurred in the network using a protocol and maintaining log. The attackers may disrupt packet forwarding (i.e., the *data plane* of the network) by dropping packets routed to it by its neighbors. Authentication of the routing protocol messages is not sufficient to prevent the disruption of routing. Even though the Border Gateway Routing Protocol (BGP)[6] is central for Internet packet routing, it was designed for a trusted environment and provides relatively minimal security against an attacker. We need a way to securely detect and localize the source of packet forwarding misbehavior so that the problem can then be corrected by routing around the trouble spot.

2. Related Works

There are two threats posed by a compromised packet: The first is that it might be attacked by the hacker. The second is the malfunctioning of the router. Secure traceroute [15] is a link-level detection scheme that could conceivably be applied at the path level. However, this scheme may fail to detect attacks that target low-rate components of the aggregate traffic in a path or attacks that exploit the TCP mechanism. Other proposals, such as Listen [16] and Feedback-Based Routing [17], detect data-plane attacks by monitoring traffic at the TCP level. However, this scheme may fail to detect attacks that target low-rate components of the aggregate traffic in a path or attacks that exploit the TCP mechanism. The earliest work on fault-tolerant forwarding is due to Perlman [1], [2] developed a novel method for robust routing based on source routing, digitally signed route-setup packets, reserved buffers. However, many implementation details are left open and the protocol requires higher network level participation to detect anomalies. Assumptions were made that the network uses a single-path routing protocol [3] of some kind. Networks where, for example, all traffic is propagated by flooding can achieve robustness in the complete absence of identities and quite possibly in the presence of numerous malicious adversaries. But singlepath routing protocols have more difficulty dealing with individual misbehaving routers, since it is easier for the adversary to disrupt the forwarding of a stream of unreplicated packets along a common path. A mechanism to detect such misbehavior is therefore desirable. WATCHERS system detects disruptive routers passively via a distributed monitoring algorithm that detects deviations from a “conservation of flow” invariant [4], [5]. However, work on WATCHERS was abandoned, in part due to limitations in its distributed detection protocol, its overhead, and the problem of ambiguity stemming from congestion[5]. [8], [9] present a secure router routing a combination of source routing, hop by hop authentication, end-to-end reliability mechanisms, and timeouts. But, it still has a high overhead to be deployable in modern networks.

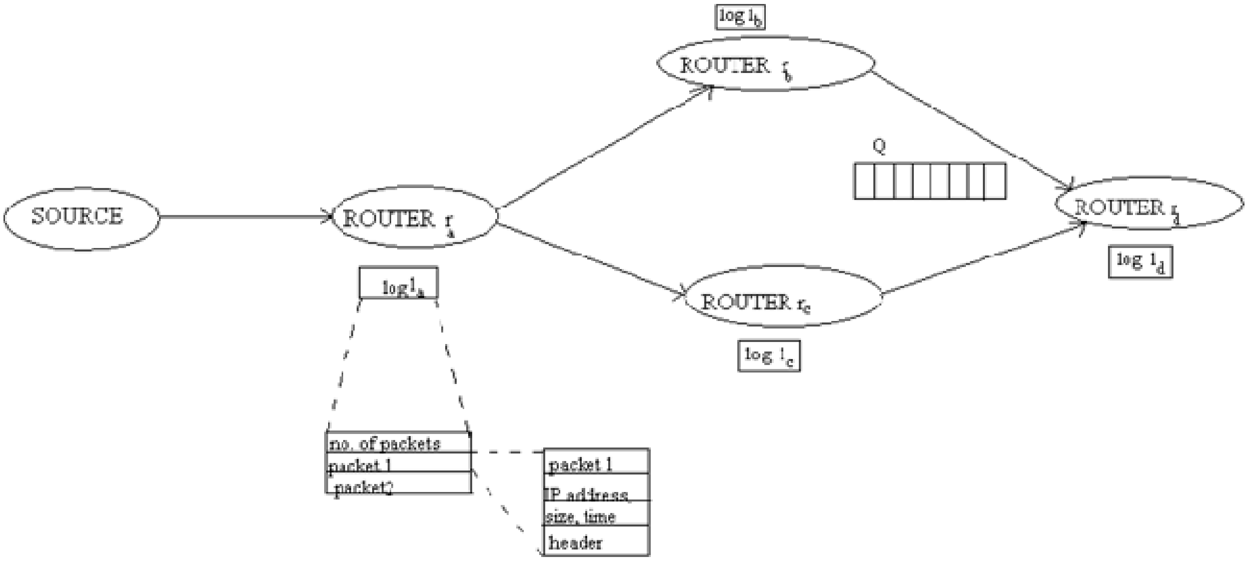


Fig a. Routing in a network

4.Assumptions

- Low rates of packet loss are assumed to be congestive, while rates above some predefined threshold are malicious.
- Packet loss rates are predicted as a function of traffic parameters and losses beyond the prediction are malicious.
- Individual packet losses are predicted as a function of measured traffic load and router buffer capacity. Deviations from these predictions are malicious

3.Protocol

The protocol used maintains log in each router stating the information about each packet that passes through it. If the actual behavior deviates from the predicted behavior, then a failure has occurred.

a. Packets Information

Consider a queue Q as shown in figure. The neighbor routers r_b, r_c feed data in queue Q. Each Q has the order by which the packets should enter along with its associated information. Each router maintains a log record.

Let Q_{in} be the traffic before entering the queue and Q_{out} be the traffic after leaving the queue.

At any instant time, the traffic is represented as $R(Q,q_p(t),I,F)$ where

1. $q_p(t)$ is the predicted state of queue at any time ‘t’.
2. I is the traffic before entering the queue by the information collected from neighbouring routers (r_b, r_c).
3. D is the traffic after leaving the queue, collected at router rd.

If $R(Q,q_p(t),I,F)$ is false and the routers are not protocol faulty, then the packets are dropped maliciously at time ‘t’. Each packet forwarded maintains a log record which includes:

- Header name P
- IP address (from where it was forwarded)
- Packet size(no. of routers it should traverse) ps
- Time at which it arrived at the router

Every log is compared with the previous one before it is forwarded. In our case, if loga ≠ logb, then r_b stops forwarding packets further- detect failure.

Three criteria could be used for predicting the state of the packet P:

- If P came from F, then the packet is leaving Q
- If P came from I (P traversed D), the packet is entering the Q and will exit at q_p+p_s
- If P came from I (P hasn't traversed D), the packet is entering the final Queue and is received at the destination.

To detect how the attack occurred, two conditions have to be satisfied :

- Buffer limit(B) is maintained at each router. If $B<q_p+p_s$, then the packet P is dropped due to congestion.
- Otherwise, the packet P is dropped due to malicious access.

We use two tests:
Confidence value test
Detecting malicious router test

Confidence value test:
We introduce a term C_v which is the probability of an attack to occur. If a packet P is dropped at time t at queuelength qp, then C_v is raised. This is suitable only when a single packet is lost.

We use the following terms :
qs(t) – size of the queue for packet P
ps – size of the packet P
q_{lim} – max size of queue
X – new malicious packet inserted

C_v is calculated as below:

C_v = Prob(Packet P to be dropped)
= Prob(more space in the queue)
= $Prob(qs(t)+ps \leq q_{lim})$
= $Prob(X+q_p(t)+ps \leq q_{lim})$
= $Prob(X \leq q_{lim}-q_p(t)-ps)$
= $Prob(Y \leq (q_{lim}-q_p(t)-ps-\mu)/\sigma)$
Random variable $Y= (X- \mu)/\sigma$
$=Prob(Y \leq y)$
$y = (qlim-q_p(t)-ps- \mu)/ \sigma$
$= (1+erf(y/\sqrt{2}))/2$

Detecting malicious router test:
This is based on the well-known Z-test4 [10]. Let N be the no. of packets lost due to malicious access. For those N packets, let $\overline{qs(t)}$ be the mean of qs(t). Let $\overline{ps}$ be the mean of ps and $\overline{qp(t)}$ be the mean of $q_p(t)$. The packet loss occurs only when $X > q_{lim}-q_p$-ps. The Z test score is:

$Z = ((q_{lim}-\overline{qp(t)}-\overline{ps}-\mu)/(\sigma\sqrt{n}))$

4.1Protocol Faulty

A faulty router can also be protocol faulty. It can behave arbitrarily with respect to the protocol, by dropping or altering the control messages of X. We mask the effect of protocol faulty routers using distributed detection.

For a queue Q, the routers involved in the detection are:
r_b,r_c – which sends the traffic
r_d – the router to which Q's traffic is forwarded.

Each router in the network collects the following information at time t.
r_s : collects $R(r_s,Q_{in},(r_b,r_c),t)$
r_d: collects $R(r_d,Q_{out}(r_s,r_d),t)$

1. Let T be the max time to forward traffic information
a. If rd doesn't receive any traffic information within T, then it detects (r_b,r_d) and (r_c,r_d)
b. If rd has received the traffic information, it verifies $R(r_s,Q_{in}(r_b,r_c),t)$ to see whether it matches $R(r_d,Q_{in}(r_d),t)$. If so, it forwards the packet to the next router.
If not, it again detects (r_b,r_d) and (r_c, r_d). If a failure is detected, then it forwards its own copy of traffic information $R(r_d,Q_{in}(r_d),t)$.

2. a. If rd doesn't receive any traffic information after 2T, then it announces r_b and r_c as faulty.
b.After receiving the traffic information, it checks the integrity and the authenticity of the message. If it fails, it again detects (r_c,r_d) and (r_b,r_d).
c. After receiving all traffic information by rd, it calculates the predicated traffic R1. If $R1(r_d,Q_{in}(r_d),t)$ evaluates to false, it detects (r_c,r_d) and (r_b,r_d).

Fault detections Ia, Ib, IIa, and IIb are due to protocol faulty routers, and fault detection IIc is due to the traffic validation detecting traffic faulty routers.

4.2 Protocol Analysis

Computing Traffic Validation:

The time complexity of computing TV depends on the size of the traffic information collected and received from the neighbors that are within two hops Hence, it depends on the traffic volume of the network. If traffic information stores the packets in order of increasing time stamps, then a straightforward implementation of traffic validation exists.

Router:

Let M be the number of routers in the network, and L be the maximum number of links incident on a router. Our Protocol requires a router to monitor the path segments that are at most two hops away.

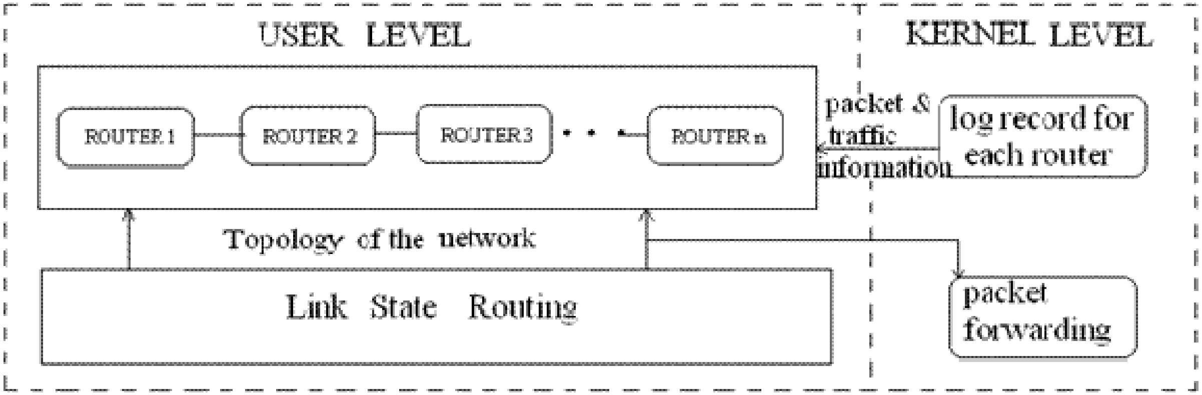


Fig b. System Architecture

5.Performance Evaluation

We evaluate mainly the performance according to the following metrics.

Control overhead: The control overhead is defined as the ratio between total number of packets to be sent to the total number of received data packets.

Average end-to-end delay: .The end-to-end-delay is the average time taken by data packets to reach from the sources to the destinations. This includes all the delays caused during route acquisition, buffering and processing at intermediate routers, etc.

Average Packet Delivery Ratio: This is the fraction of the data packets generated by the sources that are delivered to the destination. This evaluates the ability of the protocol to discover routes [12].

Router Reliability: The node reliability is calculated by the packet delivery ratio of that particular router. If the ratio is high means reliability is also high

6.System Architecture

In this paper, we have implemenented a router based protocol. The system architecture shown has 4 principal components.

6.1 Router

Routers are machines that direct traffic flow on any sizeable computer network. Every packet of data a router receives must be correctly forwarded to the next appropriate router, or *hop*. In order to do this, each router maintains a *routing table* listing the appropriate next hop for specific destinations and/or destination networks. Upon receiving a packet, a router parses that packet's *header*, extracting, among other things, the destination address. The router then checks its routing table, performs any maintenance or special instructions requested in the packet header, and sends the packet out on the correct network interface.. A router can be traffic faulty by maliciously dropping packets and protocol faulty by not following the rules of the detection protocol.

6.2 Log Record

Each router in the network maintains a log record[13] containing information about the number of packets sent and received (N), the size of each packet(ps), header of the packet(P), time at which the packet was received(t). This log record helps in detecting where the loss in packet occurred. Each router maintains a queue(Q) before it gets the particular packets. Attacks occur only when Q_{lim}(maximum size of queue) is not full. When Q_{lim}<ps+t, then the packet is dropped due to congestion or else some malicious attack has occurred. When a packet arrives at router r and is forwarded to a destination that will traverse a path segment ending at router x, r increments an outbound counter associated with router x. Conversely, when a packet arrives at router r, via a path segment beginning with router x, it increments its inbound counter associated with router x. Periodically, router x sends a copy of its outbound counters to the associated routers for validation. Then, a given router r can compare the number of packets that x claims to have sent to r with the number of packets it counts as being received from x, and it can detect the number of packet losses. The mechanism is already discussed in section 4 and 4.1

6.3 Link State Routing

The Routing Daemon, which is based on Zebra [14] in the current prototype manages link state announcements, shortest path computation and forwarding table calculation and installation. We define a path to be a finite sequence $(r_1,r_2\ldots r_n)$ of adjacent routers. Operationally, a path defines a sequence of routers a packet can follow. We call the first router of the path the source and the last router its sink; together, these are called terminal routers. A path might consist of only one router, in which case the source and sink are the same. Terminal routers are leaf routers: they are never in the middle of any path. In addition, we have modified the protocol to incorporate input from log record. When the router has received the traffic information, it verifies its log record with the previous router. If any changes are found, then the verification proceeds to see whether it was due to congestion or malicious accesss or router faulty, the loss has occurred as described in section 4.1

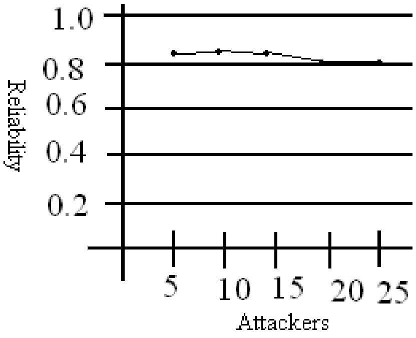
6.4 Packet Forwarding

A router maintains a distinct forwarding table for each detected suspicious path segment that the current router is in the middle. The database defines the criteria used to decide which forwarding table should be used to look-up a packet.

7.Results

7.1 ATTACKERS VS RELIABILTY RATIO:

Figure shows the reliability for misbehaving routers (5,10,…) Clearly, our protocol receives more reliability than the previous works.



Fig

7.2 QUEUE SIZE:

As queue size is empty, it is prone to be attacked by hackers and introduce malicious contents in the packet as shown in the figure 2a . When the queue size is maintained small as in figure 2b and when the queue is almost full, packet could not be as easily accessed by the hackers for malicious attack. When the queue is too small, then the packet could be lost as a result of congestion as shown in figure 2c.

Fig 2a

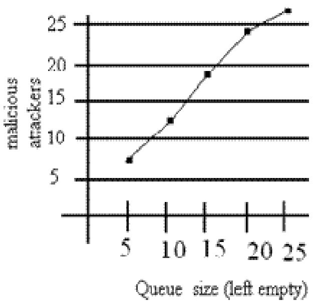


Fig 2b

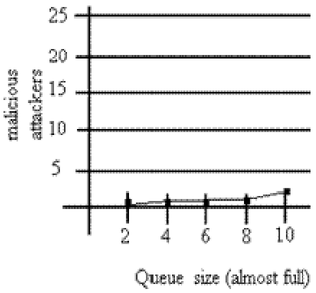
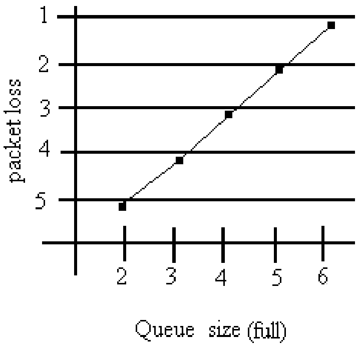


Fig 2c



7.3 QUEUE RESULTS:

Fig 3a shows the data dropped due to congestion in queue. That is when the packet size has exceeded the queue limt(Q_{lim}). The selected(circled) packets will be dropped since it has exceeded Q_{lim}. Fig 3b shows the malicious packet inserted(circled) since the queue is not full and it is prone to be attacked by hackers.

Fig 3a

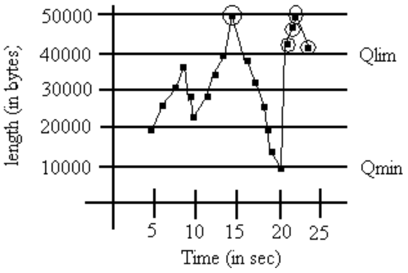
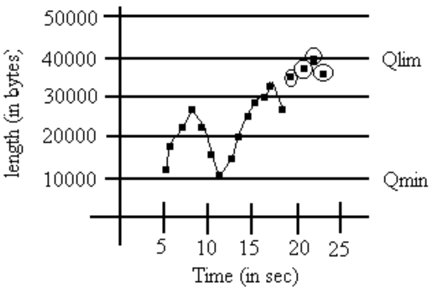


Fig 3b



8. Conclusion

We have implemented a compromised router detection protocol that dynamically infers, based on measured traffic rates and buffer sizes, the number of congestive packet losses that will occur. Subsequent packet losses can be attributed to malicious actions. Our protocol maintains log record and helps the user to know where the packet loss happened in the topology of the network. It also tells us whether it is due to malicious access or traffic congestion.

9.References

[1] R. Perlman, “Network Layer Protocols with Byzantine Robustness,” PhD dissertation, MIT LCS TR-429, Oct. 1988

[2] R. Perlman, Interconnections: Bridges and Routers. Addison Wesley Longman Publishing Co. Inc., 1992.

[3] “ Securing Routing in Open Networks Using Secure Traceroute “, Gaurav Mathur, Venkata N. Padmanabhan ,Daniel R. Simon,*Microsoft Research*, July 2004

[4] S. Cheung and K.N. Levitt, “Protecting Routing Infrastructures from Denial of Service Using Cooperative Intrusion Detection,” Proc. Workshop on New Security Paradigms (NSPW ’97), pp. 94-106, 1997.

[5] K.A. Bradley, S. Cheung, N. Puketza, B. Mukherjee, and R.A. Olsson, “Detecting Disruptive Routers: A Distributed Network Monitoring Approach,” Proc. IEEE Symp. Security and Privacy (S&P ’98), pp. 115-124, May 1998.

[6] R. White. Deployment Considerations for Secure Origin BGP (soBGP), draft-white-sobgp-bgp-deployment-01.txt. Draft, Internet Engineering Task Force, June 2003

[7] J.R. Hughes, T. Aura, and M. Bishop, “Using Conservation of Flow as a Security Mechanism in Network Protocols,” Proc. IEEE Symp. Security and Privacy (S&P ’00), pp. 131-132, 2000.

[8] I. Avramopoulos, H. Kobayashi, R. Wang, and A. Krishnamurthy, “Highly Secure and Efficient Routing,” Proc. INFOCOM Conf.,
Mar. 2004.

[9] I. Avramopoulos, H. Kobayashi, R. Wang, and A. Krishnamurthy, “Amendment to: Highly Secure and Efficient Routing,” amendment,
Feb. 2004.

[10] R.J. Larsen and M.L. Marx, Introduction to Mathematical Statistics and Its Application, fourth ed. Prentice Hall, 2005

[11] J. Black, S. Halevi, H. Krawczyk, T. Krovetz, and P. Rogaway, “UMAC: Fast and Secure Message Authentication,” LNCS, vol. 1666, pp. 216-233, 1999.

[12] Kimaya Sanzgiri, Bridget Dahill, Brian Neil Levine, Clay Shields and Elizabeth M. Belding-Royer, "A Secure Routing Protocol for Ad Hoc Networks", in proc. of 10th International Conference on Network Protocols, pp: 78-87, 12-15 November 2002.

[13] Alper T. Mizrak, Stefan Savage and Keith Marzullo, “Detecting Malicious Packet Losses” IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 20, NO. 2, FEBRUARY 2009

[14] GNU Zebra, http://www.zebra.org, 2006.

[15] V. Padmanabhan and D. Simon. Secure traceroute to detect faulty or malicious routing. In *Proc. ACM SIGCOMM
HotNets Workshop*, Oct. 2002.

[16] L. Subramanian, V. Roth, I. Stoica, S. Shenker, and R. Katz. Listen and Whisper: Security mechanisms for BGP. In *Proc. Symposium on Networked System Design and Implementation*, Mar. 2004.

[17] D. Zhu, M. Gritter, and D. Cheriton. Feedback based routing. In *Proc. ACM SIGCOMM HotNets Workshop*, Oct.
2002.

Julian Benadit.P is currently working as a lecturer in Rajiv Gandhi College of Engineering and Technology, Pondicherry. He received B.Tech degree in computer science and engineering from Pondicherry University. He also received M.Tech degree in computer science and engineering from Anna University. He is a life member in ISTE (International Society for Technology in Education) ,CSI,IETE,SSI and Instituion of Engineers. He has published paper in IEEE Explore. He has attended International Conference on Wireless Communication and Sensor Computing.

Sharmila.B is currently pursuing her final year of B.Tech degree in computer science and engineering from Rajiv Gandhi of College and Technology, Pondicherry. She has presented various papers in college symposiums.

Ramya.T is currently pursuing her final year of B.Tech degree in computer science and engineering from Rajiv Gandhi of College and Technology, Pondicherry.