Statistical Traffic Approach for Detecting Malicious Packets

D.Pavan Kumar[1], R.L.K.Prasanna[2], M.Mounika[3], Sk.A.T.Ahmed[4], Sk. Jaharunnisa[5]
[1]*Assistant Professor, Dept. of. CSE, Tirumala Engg College, Jonnalagadda, NRT, AP, India*
[2,3,4,5] *U.G. Students, Dept. of. CSE, Tirumala Engg College, Jonnalagadda, NRT, AP, India*

Abstract—
We consider the problem of detecting whether a compromised router is maliciously manipulating its stream of packets. In particular, we are concerned with a simple yet effective attack in which a router selectively drops packets destined for some victim. Unfortunately, it is quite challenging to attribute a missing packet to a malicious action because normal network congestion can produce the same effect. Modern networks routinely drop packets when the load temporarily exceeds their buffering capacities. Previous detection protocols have tried to address this problem with a user defined threshold: too many dropped packets imply malicious intent. We have designed, developed, and 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. Once the ambiguity from congestion is removed, subsequent packet losses can be attributed to malicious actions. We have tested our protocol in Emulab and have studied its effectiveness in differentiating attacks from legitimate network behavior. 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.*

I. INTRODUCTION

The Internet is not a safe place. Unsecured hosts can expect to be compromised within minutes of connecting to the Internet and even well-protected hosts may be crippled with denial-of-service (DoS) attacks. However, while such threats to host systems are widely understood, it is less well appreciated that the network infrastructure itself is subject to constant attack as well. Indeed, through combinations of social engineering and weak passwords, attackers have seized control over thousands of Internet routers. Even more troubling is Mike Lynn's controversial presentation at the 2005 Black Hat Briefings, Once a router has been compromised in such a fashion, an attacker may interpose on the traffic stream and manipulate it maliciously to attack others selectively dropping, modifying, or rerouting packets. 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.

II. RELATED DATA

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 dataplane 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.

III. EXISTING SYSTEM

Previous detection protocols have tried to address this problem with a userdefined threshold: too many dropped packets imply malicious intent. However, this heuristic is fundamentally unsound; setting this threshold is, at best, an art and will certainly create unnecessary false positives or mask highly focused attacks. The earliest work on fault-tolerant forwarding is due to Pearlman who developed a robust routing system based on source routing, digitally signed route-setup packets, and reserved buffers.

Static Threshold: Low rates of packet loss are assumed to be congestive, while rates above some predefined threshold are deemed malicious.

Traffic modeling: Packet loss rates are predicted as a function of traffic parameters and losses beyond the prediction are deemed malicious.

.

Traffic measurement: Individual packet losses are predicted as a function of measured traffic load and router buffer capacity. Deviations from these predictions are deemed malicious.

IV. PROPOSED SYSTEM

In contrast, protocol X can detect such malicious behaviors because it measures the router's queues, which are determined by the dynamics of the network transport protocol. Protocol can report false positives and false negatives, but the probability of such detections can be controlled with a significance level for the statistical tests upon which is built. A static threshold cannot be used in the same way. To summarize, these protocols are designed to detect anomalies between pairs of correct nodes, and thus for simplicity, it is assumed that a terminal router is not faulty with respect to traffic originating from or being consumed by that router.

A. 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

B. Modules

1. Create Network Environment
2. Packet Collection Operation.
3. Packet forwarding using Static Threshold.
4. Selection of congested area
5. Compromised router Detection Protocol.

In first module, at first we create an environment. The environment setup can be in rectangular area. The nodes in the environment can be aligned in Random Access method. It means each node consists of four neighbor nodes. It can be fixed through mesh topology. In second module packet collection process was done between the aggregated nodes and the member nodes. Using the coverage distance and timing events.

In third module described about forwarding the packets using buffer size. It defines the node number, size of the packet and the packet loss using the static threshold method.

V. CONCLUSIONS

To the best of our knowledge, this paper is the first serious attempt to distinguish between a router dropping packets maliciously and a router dropping packets due to congestion. Previous work has approached this issue using a static user defined threshold, which is fundamentally limiting. Using the same framework as our earlier work, we developed a compromised router detection protocol x 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. Because of no determinism introduced by imperfectly synchronized clocks and scheduling delays, protocol x uses user defined significance levels, but these levels are independent of the properties of the traffic.

VI. REFERENCES

[1] Chieh-Jen Cheng, Chao-Ching Wang, Wei-Chun Ku, Tien-Fu Chen , and Jinn- Shyan Wang, "Scalable High- erformance Virus Detection Processor Against a Large Pattern Set for Embedded Network Security" Commun. vol. 51, pp. 62–70,2011.

[2] O. Villa, D. P. Scarpazza, and F. Petrini, "Accelerating real-time string searching with multicore processors," Computer, vol. 41, pp. 42–50,2008.
[3] D. P. Scarpazza, O. Villa, and F. Petrini, "High-speed string searching against large dictionaries on the Cell/B.E. processor," in Proc. IEEE Int. Symp. Parallel Distrib. Process., 2008, pp. 1–8.
[4] D. P. Scarpazza, O. Villa, and F. Petrini, "Peak-performance DFA based string matching on the Cell processor," in Proc. IEEE Int. Symp. Parallel Distrib. Process., 2007, pp. 1–8.
[5] L. Tan and T. Sherwood, "A high throughput string matching architecture for intrusion detection and prevention,"in Proc. 32nd Annu. Int. Symp. Comput. Arch., 2005, pp. 112–122.
[6] S. Dharmapurikar, P. Krishnamurthy, and T. S. Sproull, "Deep packet inspection using parallel bloom filters," IEEE Micro, vol. 24, no. 1, pp.52–61, Jan. 2004.
[7] R.-T. Liu, N.-F. Huang, C.-N. Kao, and C.-H. Chen, "A fast string matching algorithm for network processor-based intrusion detection system," ACMTrans. Embed. Comput. Syst., vol. 3, pp. 614–633, 2004.
[8] F. Yu, R. H. Katz, and T. V. Lakshman, "Gigabit rate packet pattern matching using TCAM," in Proc. 12th IEEE Int. Conf. Netw. Protocols, 2004, pp. 174–178.intrusion detection system," ACMTrans. Embed. Comput. Syst., vol. 3, pp. 614–633, 2004.
[9] R. S. Boyer and J. S. Moore, "A fast string searching algorithm,"Commun. ACM, vol. 20, pp. 762–772, 1977.
[10] V. Aho and M. J. Corasick, "Efficient string matching: An aid to bibliographic search," Commun. ACM, vol. 18, pp. 333– 340, 1975
[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.