Volume 4, Issue 9, September 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Creative Representation for Detecting Nasty Packet Losses

Siddireddy Parimala[1], K. J. Pavithran Kumar[2] , Bhima Sankaram Alladi[3], B. Sandhya Rani[4]
[1, 4]Department of Computer Science and Engineering, RGM College of Engineering &Technology, Nandyal, AP, India
[2]Department of Computer Science and Engg., Mother Theresa Institute of Engg and Technology,Chittoor Dt, AP, India
[3]Department of Information Technology, G.Narayanamma Inst of tech & science for Women, R.R Dist, Telangana, India

Abstract: In this paper, 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. 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. 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.

Keywords: Denial of service, Emulab, Object Modeling Technique (OMT), Class Responsibility Collaborator (CRC), Random Early Detection.

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, which demonstrated how Cisco routers can be compromised via simple software vulnerabilities. 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. Several researchers have developed distributed protocols to detect such traffic manipulations, typically by validating that traffic transmitted by one router is received unmodified by another. However, all of these schemes—including our own—struggle in interpreting the absence of traffic. While a packet that has been modified in transit represents clear evidence of tampering, a missing packet is inherently ambiguous: it may have been explicitly blocked by a compromised router or it may have been dropped benignly due to network congestion. In fact, modern routers routinely drop packets due to bursts in traffic that exceed their buffering capacities, and the widely used Transmission Control Protocol (TCP) is designed to cause such losses as part of its normal congestion control behavior. Thus, existing traffic validation systems must inevitably produce false positives for benign events and/or produce false negatives by failing to report real malicious packet dropping. In this paper, we develop a compromised router detection protocol that dynamically infers the precise number of congestive packet losses that will occur. Once the congestion ambiguity is removed, subsequent packet losses can be safely attributed to malicious actions. We believe our protocol is the first to automatically predict congestion in a systematic manner and that it is necessary for making any such network fault detection practical. In the remainder of this paper, we briefly survey the related background material, evaluate options for inferring congestion, and then present the assumptions, specification, and a formal description of a protocol that achieves these goals. We have evaluated our protocol in a small experimental network and demonstrate that it is capable of accurately resolving extremely small and fine-grained attacks.

Background

There are inherently two threats posed by a compromised router. The attacker may subvert the network control plane (e.g., by manipulating the routing protocol into false route updates) or may subvert the network data plane and forward individual packets incorrectly. The first sets of attacks have seen the widest interest and the most activity—largely due to their catastrophic potential. By violating the routing protocol itself, an attacker may cause large portions of the network to become inoperable. Thus, there have been a variety of efforts to impart authenticity and consistency guarantees on route update messages with varying levels of cost and protection. We do not consider this class of attacks

in this paper. Instead, we have focused on the less well-appreciated threat of an attacker subverting the packet forwarding process on a compromised router. Such an attack presents a wide set of opportunities including DoS, surveillance, man-in-the-middle attacks, replay and insertion attacks, and so on. Moreover, most of these attacks can be trivially implemented via the existing command shell languages in commodity routers. The earliest work on fault-tolerant forwarding is due to Perlman who developed a robust routing system based on source routing, digitally signed route-setup packets, and reserved buffers. While groundbreaking, Perlman's work required significant commitments of router resources and high levels of network participation to detect anomalies. Since then, a variety of researchers have proposed lighter weight protocols for actively probing the network to test whether packets are forwarded in a manner consistent with the advertised global topology. Conversely, the 1997 WATCHERS system detects disruptive routers passively via a distributed monitoring algorithm that detects deviations from a "conservation of flow" invariant. 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. Finally, our own work broke the problem into three pieces: a traffic validation mechanism, a distributed detection protocol, and a rerouting countermeasure. In and, we focused on the detection protocol, provided a formal framework for evaluating the accuracy and precision of any such protocol, and described several practical protocols that allow scalable implementations. However, we also assumed that the problem of congestion ambiguity could be solved, without providing a solution. This paper presents a protocol that removes this assumption.

II. RELATED WORK

Project Description:

I need software developed to detect malicious packet losses using Random Early Detection (RED) Algorithm within an office of 10 networked computers. This project considers the problem of detecting malicious packet losses in a computer networks. This project is concerned with a simple yet effective attack in which movement of packet in a network is being tampered with. For example, When a packet (data) is sent across in a computer from a particular point e.g node A(source address) and its supposed to be taking to node D(destination address) but unfortunately, the packet was dropped along the way at node C to be precise and was unable to get to its proper destination so as to deliver its packet. This means the packet has been maliciously dropped. But it is quiet abnormal to attribute every packet loss to a malicious action because normal network congestion can as well cause packet loss. In other words, computer network tend to drop packet when the load of data's being processed (sending of files from one computer to the other in a computer network) exceeds their buffering capacities and this is where the function of the RED Algorithm comes in helping to remove the over ambiguity(overload) of traffic congestion in a network so that the network can be free of traffic congestion and once the overload from traffic congestion is removed, subsequent packet losses can be attributed to malicious actions.

Existing System:

The earliest work on fault-tolerant forwarding is due to Perlman who developed a robust routing system based on source routing, digitally signed route-setup packets, and reserved buffers. While groundbreaking, Perlman's work required significant commitments of router resources and high levels of network participation to detect anomalies. Since then, a variety of researchers have proposed lighter weight protocols for actively probing the network to test whether packets are forwarded in a manner consistent with the advertised global topology. Conversely, the 1997 WATCHERS system detects disruptive routers passively via a distributed monitoring algorithm that detects deviations from a "conservation of flow" invariant. 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.

Proposed System:

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 Emu lab and have studied its effectiveness in differentiating attacks from legitimate network behavior.

Network routers occupy a unique role in modern distributed systems. They are responsible for cooperatively shuttling packets amongst themselves in order to provide the illusion of a network with universal point-to-point connectivity. However, this illusion is shattered – as are implicit assumptions of availability, confidentiality, or integrity – when network routers are subverted to act in a malicious fashion. By manipulating, diverting, or dropping packets arriving at a compromised router, an attacker can trivially mount denial-of-service, surveillance, or man-in-the-middle attacks on end host systems. Consequently, Internet routers have become a choice target for would-be attackers and thousands have been subverted to these ends.

III. SYSTEM ANALYSIS

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

In building a traffic justification protocol, it is necessary to explicitly resolve the haziness around packet losses. Should the absence of a given packet be seen as malicious or benign? In practice, there are three approaches for addressing this issue:

- 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.

Single Packet Loss Test:

If a packet with fingerprint fp and size ps is dropped at time ts when the predicted queue length is qpredðtsÞ, then we raise an alarm with a confidence value csingle, which is the probability of the packet being dropped maliciously. The mean _ and standard deviation _ of X can be determined by monitoring during a learning period. We do not expect _ and _ to change much over time, because theyare in turn determined by values that themselves do not change much over time. Hence, the learning period need not be done very often.

Distributed Detection:

Since the behavior of the queue is deterministic, the traffic validation mechanisms detect traffic faulty routers whenever the actual behavior of the queue deviates from the predicted behavior. However, 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 _. We mask the effect of protocol faulty routers using distributed detection.

Given TV , we need to distribute the necessary traffic information among the routers and implement a distributed detection protocol. Every outbound interface queue Q in the network is monitored by the neighboring routers and validated by a router rd such that Q is associated with the link hr; rdi. With respect to a given Q, the routers involved in detection are

- rs_ , which sends traffic into Q to be forwarded.
- r, which hosts Q.
- rd, which is the router to which Q's outgoing traffic is forwarded.

Traffic Information Collection

Each router collects the following traffic information during a time interval t:

- rs_ : Collect Tinfoðrs_;Qin; hrs_; r; rdi; _Þ.
- r: Collect Tinfoðr;Qin; hrs_; r; rdi; _Þ. This information is used to check the transit traffic information sent by the rs_ routers.
- rd: Collect Tinfoðrd;Qout; hr; rdi; _Þ.

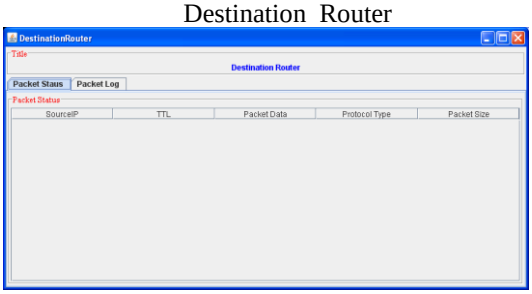
Traffic Validation Correctness

Any failure of detecting malicious attack by TV results in a false negative, and any misdetection of legitimate behavior by TV results in a false positive. Within the given system model of Section 4, the example TV predicate in Section 5.1 is correct. However, the system model is still simplistic. In a real router, packets may be legitimately dropped due to reasons other than congestion: for example, errors in hardware, software or memory, and transient link errors. Classifying these as arising from a router being compromised might be a problem, especially if they are infrequent enough that they would be best ignored rather than warranting repairs the router or link.

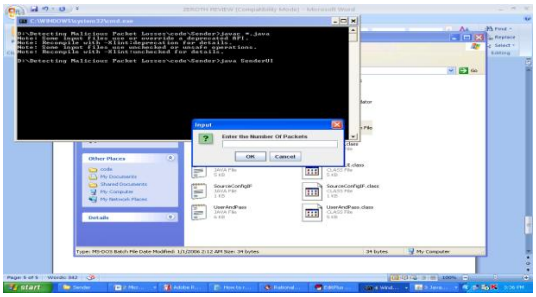
V. RESULT AND ANALYSIS

Source Router

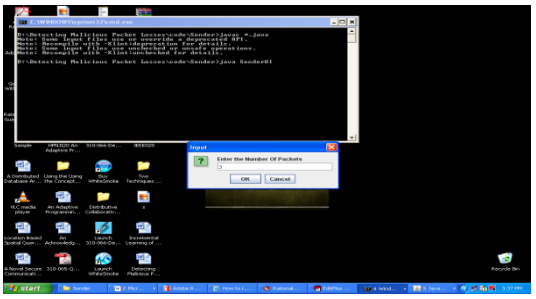
Destination Router

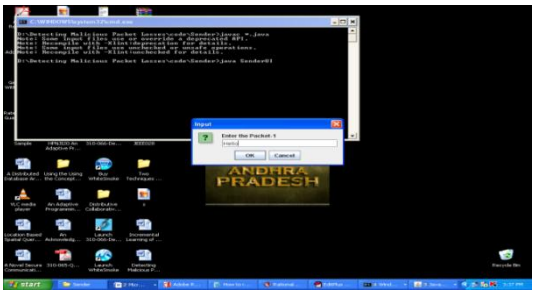


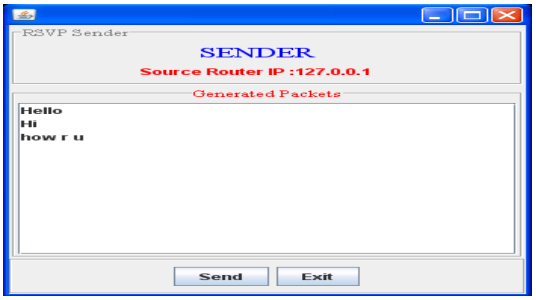
Enter No of Packets

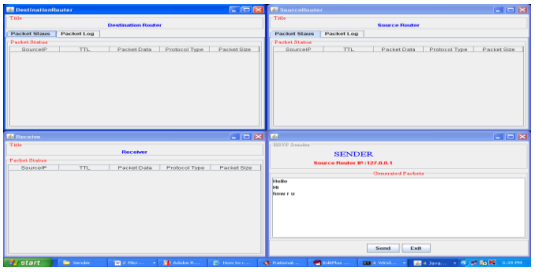


Packet Information

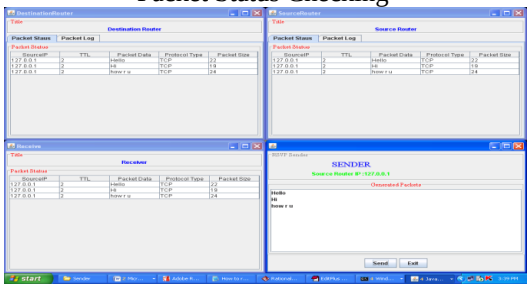


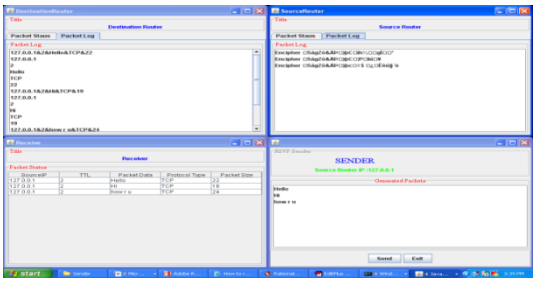


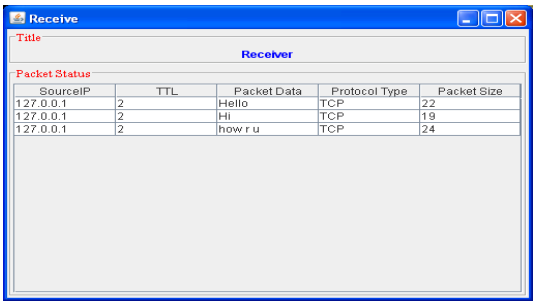




Packet Status Checking







VI. CONCLUSION

The designed, developed, and implemented system is 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 Emu lab and have studied its effectiveness in differentiating attacks from legitimate network behavior. 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 (which is based on a static user-defined threshold), we developed 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. Because of no determinism introduced by imperfectly synchronized clocks and scheduling delays, protocol _ uses user-defined significance levels, but these levels are independent of the properties of the traffic. Hence, protocol _ does not suffer from the limitations of static thresholds. We evaluated the effectiveness of protocol _ through an implementation and deployment in a small network. We show that even fine-grained attacks, such as stopping a host from opening a connection by discarding the SYN packet, can be detected.

VII. FUTURE ENHANCEMENT

Similar to the specification that we have defined in, we cast the problem as a failure detector with accuracy and completeness properties.

. a-**Accuracy**: A failure detector is a-Accurate if whenever a correct router suspects ð_; _Þ, then j_j _ a and some router r 2 _ was faulty in _ during _. We use the term traffic faulty to indicate a router that drop packets from transit traffic and the term protocol faulty to indicate a router that behaves arbitrarily with respect to the detection protocol. A faulty router is one that is traffic faulty, protocol faulty, or both. As before, we will add the phrase "in _" to indi cate that the faulty behavior is with respect to traffic that transits the path _. The a-Accuracy requirement can result in a detection if a router is either protocol faulty or traffic faulty. . a-Completeness: A failure detector is a-Complete if whenever a router r is traffic faulty at some time t, then all correct routers eventually suspect ð_; _Þ for some path segment _ : j_j _ a such that r was traffic faulty in _ at t, and for some interval _ containing t. Distinguishing between protocol faulty and traffic faulty behavior is useful because, while it is important to detect routers that are traffic faulty, it is not as critical to detect routers that are only protocol faulty: routers that are only protocol faulty are not dropping packets maliciously. Given an accurate and complete TV predicate, protocol _ is 2-accurate and 2-complete.

REFERENCES

[1] X. Ao, Report on DIMACS Workshop on Large-Scale Internet Attacks, http://dimacs.rutgers.edu/Workshops/Attacks/internet-attack- 9-03.pdf, Sept. 2003.

[2] R. Thomas, ISP Security BOF, NANOG 28, http://www.nanog. org/mtg-0306/pdf/thomas.pdf, June 2003.

[3] 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.

[4] A.T. Mizrak, Y.-C. Cheng, K. Marzullo, and S. Savage, "Detecting and Isolating Malicious Routers," IEEE Trans. Dependable and Secure Computing, vol. 3, no. 3, pp. 230-244, July-Sept. 2006.

[5] L. Subramanian, V. Roth, I. Stoica, S. Shenker, and R. Katz, "Listen and Whisper: Security Mechanisms for BGP," Proc. First Symp. Networked Systems Design and Implementation (NSDI '04), Mar. 2004.

[6] S. Kent, C. Lynn, J. Mikkelson, and K. Seo, "Secure Border Gateway Protocol (Secure-BGP)," IEEE J. Selected Areas in Comm.,vol. 18, no. 4, pp. 582-592, Apr. 2000.
[7] Y.-C. Hu, A. Perrig, and D.B. Johnson, "Ariadne: A Secure On- Demand Routing Protocol for Ad Hoc Networks," Proc. ACM MobiCom '02, Sept. 2002.
[8] B.R. Smith and J. Garcia-Luna-Aceves, "Securing the Border Gateway Routing Protocol," Proc. IEEE Global Internet, Nov. 1996.
[9] Alper T. M_zrak, Detecting Malicious Packet Losses, Member, IEEE, and Keith Marzullo, Member, IEEE.
[10] M.T. Goodrich, Efficient and Secure Network Routing Algorithms, provisional patent filing, Jan. 2001.
[11] R. Perlman, "Network Layer Protocols with Byzantine Robustness," PhD dissertation, MIT LCS TR-429, Oct. 1988.
[12] V.N. Padmanabhan and D. Simon, "Secure Traceroute to Detect Faulty or Malicious Routing," SIGCOMM Computer Comm. Rev, vol. 33, no. 1, pp. 77-82, 2003.

Author's Profile

Ms. Siddireddy Parimala Post Graduated in Computer Science Engineering (**MCA**) From **Kasturbha Gandhi College for Women's**. She is working as Assistant Professor in Department of Computer Science & Engineering in **RGM college of Engineering &Technology**, Nandyal, Kurnool Dt., AP, India. She has 4+ years of Teaching Experience. Her research interests include Network Security, Cloud Computing & Data Warehousing and Data Mining.

Mr. K.J. PAVITHRAN KUMAR, Post Graduated in Computer Science & Engineering (M.Tech), SRI VENKATESWARA UNIVERSITY COLLEGE OF ENGINEERING, 2013 and Graduated in Computer Science & Engineering (B.Tech), Sri Krishna Devaraya University, 2007. He is working presently as an Assistant Professor in the Department of Computer Science & Engineering in **Mother Theresa Institute of Engineering &Technology**, Palamaner, Chittoor Dt., A.P, INDIA. He has 5 years Experience. He has interest in Cloud Computing, Data Mining, Image Processing, Parallel and Distributed computing and Computer Networks.

Mr. Bhima Sankaram Alladi, Post Graduated in Computer Science & Technology (**M.Tech**) From **Andhra University**, Visakhapatnam in 2010 and Graduated in Information Technology (B.Tech) form **JNTU**, Hyderabad, 2008. He is working as an Assistant Professor in Department of Information Technology in **G.Narayanamma Institute of Technology & Science for Women, Shaikpet, R.R Dist, Telangana, India**. He has 4+ years of Teaching Experience. His Research Interests Include Network Security, Cloud Computing & Data Warehousing and Data Mining.

Ms. B. Sandhya Rani, Post Graduated in Computer Science & Engineering (M.Tech), **St. Johns college of Engineering &Technology** and Graduated in Computer Science & Engineering (B.Tech), **ALFA college of Engineering & Technology**. She is working presently as an Assistant Professor in the Department of Computer Science & Engineering in **RGM College of Engineering &Technology**, Nandyal, Kurnool Dt., A.P, INDIA. She has 3+ years Experience. She has interest in Cloud Computing, Data Mining, Image Processing, Parallel and Distributed computing and Computer Networks.

