네트워크 시스템 생존성 : 소프트웨어 재활기법을 이용한 TCP의 프레임워크

킨 미미 아웅* 박종서*

*한국항공대학교, 컴퓨터공학과

Network System Survivability: A Framework of Transmission Control Protocol with Software Rejuvenation Methodology

Khin Mi Mi Aung* Jong Sou Park*

*Department of Computer Engineering Hankuk Aviation Univ.

요 약

In this paper,we propose a framework of Transmission Control Protocol with Software Rejuvenation methodology, which is applicable for network system survivability. This method is utilized to improve the survivability because it can limit the damage caused by successful attacks. The main objectives are to detect intrusions in real time, to characterize attacks, and to survive in face of attacks. To counter act the attacks' attempts or intrusions, we perform the Software Rejuvenation methods such as killing the intruders' processes in their tracks, halting abuse before it happens, shutting down unauthorized connection, and responding and restarting in real time. These slogans will really frustrate and deter the attacks, as the attacker can't make their progress. This is the way of survivability to maximize the deterrence against an attack in the target environment. We address a framework to model and analyze the critical intrusion tolerance problems ahead of intrusion detection on Transmission Control Protocol (TCP).

I . Introduction

The Transmission Control Protocol (TCP) is intended for use as a highly reliable host-to-host protocol between hosts in packet-switched computer communication networks, and in interconnected systems of such networks [26]. Internet connectivity compounds the problem by exposing the network to a host of security concerns. Even though TCP protocol suite is widely used to connect to the Internet, it has many known security weaknesses within its fundamental specification. In this paper we address a framework of Transmission Control Protocol with Software Rejuvenation methodology, which is applicable for network system survivability.

Security mechanism must include not only the ability to prevent attacks but also the ability to survive from and operate through attacks. Next generation security mechanisms may focus on survivability. We extend our security mechanisms by rejuvenating the systems to be survived even under attacks. In general, there is no solution to security problems as we have formulated it. However, techniques can be used to raise the cost of attack to a discouraging level. Survivability focuses on delivery of essential services and preservation of essential

assets, even when systems are penetrated and compromised.

Firstly, in termof statistical technique, the attacks are characterized within the short intervals by triggering the hypothesized number of different changes. Short-term real time activities focus on specific tasks and recognize the accurate, abnormal behavior and up-to-date information in a timely fashion of short intervals, since we often need to update with new attack's response methods. In order to estimate and reduce the density, we second-hand a linear projection technique called "Principle Components Analysis (PCA)[10]. This statistical detection analysis provides most powerful features in intrusion detection [5,6,12]. It can be used for identifying trends in behavioral data and damage assessment.

We will discuss the effectiveness of performing software rejuvenation method with TCP and determining the optimal time to execute the rejuvenation, which is applicable for Network system survivability. To counter act the attacks' attempts or intrusions, we perform its functions such as killing the intruders' processes in their tracks, halting abuse before it happens, shutting down unauthorized connection, and responding and restarting in real time. These slogans will really frustrate and deter the attacks, as the attacker can't make their progress. This is the way of survivability to maximize the deterrence against an attack in the target environment. But most of the attacks are novel attacks with unlabeled constraints. That is, we shall investigate what can be done when all one has is a collection of models without knowing their complexities. So next sections will be the approached analysis trends with their respective experiments.

II. Related Work

W. Cleveland et. al [3] looks at the statistical properties of Internet traffic and the difficulites of handling the complex and very large bases that result from collecting packet headers. They utilized the S statistical package for analyzing the data. J. Frank [6] used data collected by NSM L. Heberlein et. al [8] from the Internet to classify network flows including, flow duration, packets from source, packets from destination, bytes from source, bytes from destination and intrusion warning. J. Cannady [1] describes a neural network trained to detect Network intrusions from packet header data. P. Porras et. al [17] looks for statistical anomalies in network traffic that might indicate intrusions. Current traffic is compared against a database of historical traffic characteristics. The database includes metrics for traffic intensity, typical port usage, and typical active hosts. They also check the content of some network packets against a list of attack signatures. R. Caceres et. al [2] and K. Thompson et. al [21] are researched on Internet Traffic consider link utilization, packet loss, packet/data volume, network service/port usage, time-of-day, packet size, inter arrival times, flow direction. Since web/http traffic accounts for most of the Internet traffic, much work has been done on analyzing web traffic. B. Mah [14] and J. Morul [15].

III. Statistical Analysis

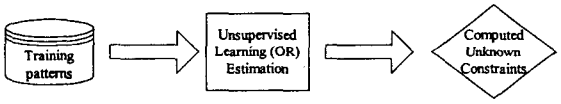

Fig.1: Basic Flow of Statistical Analysis.

In this section, we apply PCA for unsupervised learning of the normal runs and for extracting the symptoms of new attacks by minimizing their statistical profiles (Fig.1.). From the statistical analysis, we achieve a Rejuvenation time versus Interdependency by tracing the collected traces of data from that run of one day's traffic and as a result, we report average rejuvenation time versus dependency degree with attacks.

We computed unknown constraints by training the patterns from the first standard corpora for evolution of computer network intrusion detection systems, which has collected and distributed by MIT Lincoln Laboratory, under Defense Advanced Research Projects Agency (DARPA ITO) and Air Force Research Laboratory (AFRL/SNHS) sponsorship. These

training patterns are the first formal, repeatable and statistically significant evaluations. Such evaluation efforts have been carried out in 1998 and 1999.

As the basic idea of software rejuvenation is occasionally shutdown the service or cleaning the internal states, the dependency degree versus system availability becomes an important issue.

addition, since TCP connection gives up after sending retransmissions of a packet about 12 times, attackers can easily terminate the TCP service by dropping retransmissions.

For the best-effort services, our packet dropping "resolver" is a congestion management mechanism implemented at each intermediate node. And it decides, proactively drop the packets to reduce congestion and free up precious buffer space in face of attack.

Ⅳ. Experiment

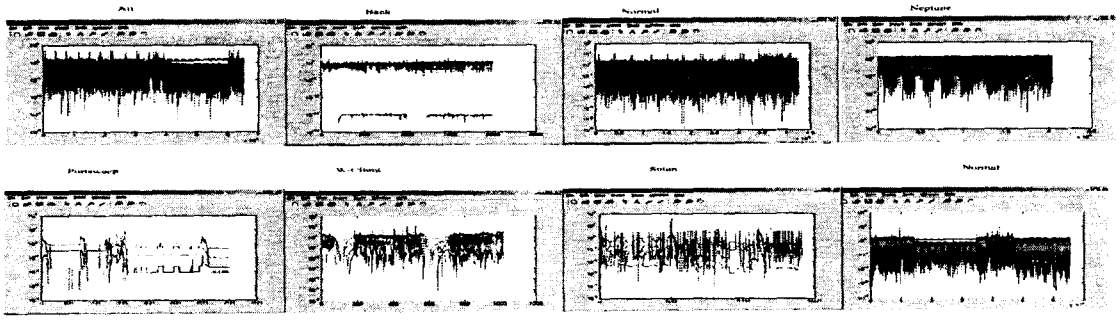

Fig. 2: Extraction the attacks and Normal from Trained Data Set.

After analyzing the features of intrusions, we have found that the various Internet services have distinctive statistical signatures and it is possible to identify certain classes of service without content analysis. Our method is based on these results. Even though the TCP protocol is widely used to connect to the Internet, it has many known security weaknesses. Attackers always drop the retransmissions of a specific packet. [5]

In this section, we present that our resolver can detect retransmissions of a specific packet. When a retransmission packet is lost, TCP goes back to slow start phase and exponentially back offs its retransmission timeout value (RTO) upon every packet loss, with an upper limit of 64 seconds. We can infer that after a few consecutive retransmissions being dropped, the sender has to wait for a long period of idle time before performing a new retransmission. Normally, no packets are sent out during this idle period. Thus, in [23], through NS2 simulation, retransmission packet dropping attacks can degrade the TCP's performance greatly by dropping only a few packets. In

With regard to TCP, these sequence number comparisons determine whether a given sequence number is in the future or a retransmission.

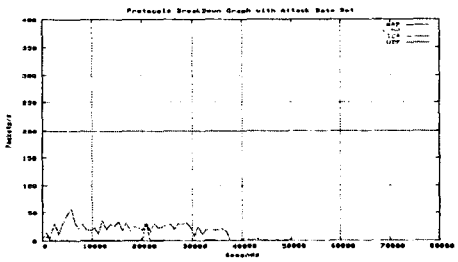

Fig. 3: Protocols Breakdown Graph with Attack Data Set.

Generally, packet dropping attacks can impact a network service on the following several aspects: Delay: e.g., dropping the retransmissions of packets in a FTP connection will drastically increase the total file transfer time. While the primary goal is to avoid or combat congestion, its designs can significantly affect application throughput, network utilization, performance fairness, and synchronization problems with multiple Transmission Control

Protocol (TCP) connections.

Proactive discard packet does not discriminate between the packets belonging to multiple TCP

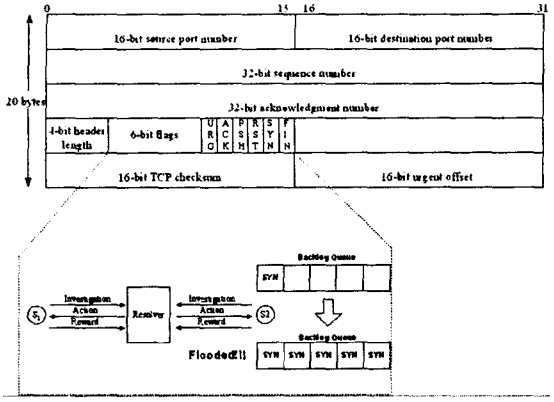

Fig. 4: TCP "resolver".

connections. This may lead to a situation where one TCP connection finishes transmitting a packet and is able to increase its window size, which in turn may cause the cell of another TCP connection to be dropped due to buffer overflow. The second TCP connection, upon a timeout, is then forced to reduce its congestion window. Thus, the first TCP connection is able to obtain an unfair share of the bandwidth. Therefore, the conditions under which a cell is dropped are (1):

(Buffer > Threshold) and (a weight W(x) > Z)

$$W(x) > Z \frac{B - Threshod}{C - Threshold} \qquad (1)$$

Where B is the buffer size and C is the total amount of cells in the buffer.

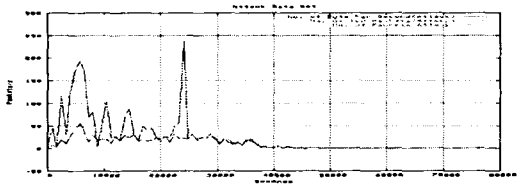

Fig. 5:Packet vs. Time (s) with Attack Data Set

V. Conclusion and Further work

Based on Statistical analysis, our experiments have been shown that the attacks are characterized within the short intervals and could be evaluate the flow probability of each attack's features. And it is possible to categorize Internet traffic flows without content analysis. According to these result outcomes, we achieve a Rejuvenation time versus Interdependency and average rejuvenation time versus dependency degree with attacks. By performing schedule and ad hoc Software Rejuvenation methods to counter act attacks' attempts or intrusions, in the optimum time, we found out the way of survivability to maximize the deterrence against an attack in the target environment. Software rejuvenation does not remove bugs resulting from software aging but rather prevents them from manifesting themselves as unpredictable whole system failures. Periodic rejuvenation limits the state space in the execution domain and transforms a no stationary random process into a stationary process that can be predicted and avoided. This survivability modeling is based on Resistance, ability of a system to repel attacks, Recognition ability to recognize attacks and the extent of damage and Recovery ability to restore essential services during attack, and recover full services after attack.

This proposed approach is a preliminary stage. The ongoing work in Software Rejuvenation for survivability has led to discovery of next more novel methodologies, the ability to dynamically trade-off security to aware with changes in their environment in real time. We will also find out the new policies based on prediction that we can solve with the temporal rejuvenation methods by mean of an error forecasting. Our approach is only suitable yet with some attacks especially concerned with internal states, garbage collection, memory defragmentation, operating system kernel tables, and reinitializing internal data structures. Then our aim and future work is about Security Rejuvenation Methodologies that would bring the system to Provide 100% of critical functionality when under sustained attack.

Acknowledgements.

This work is supported by the Korea Science and Engineering Foundation (KOSEF) through the Internet Information Retrieval Research Center at Hankuk Aviation University.

References

[1] J. Cannady "Artificial Neural Networks for Misuse Detection" NISSC, October 1998.

[2] R. Caceres, P. Danzig, S. jamin, and D. Mitzel "Characteristics of Wide Area TCP/IP Conversions" ACM SIGCOMM, September 1991.

[3] W. Cleveland and D. Sun "Internet Traffic Data" Journal of the American Statistical Association pages 979-985, September 2000.

[4] R. O. Duda, P. E. Hart, D. G. Stork "Pattern Classification" ISBN 0-471-05669-3, 2000.

[5] X. Z. Ericsson, S. F. Wu, Z. Fu, T. Wu "Malicious Packet Dropping: How it Might Impact the TCP Performance and How We Can Detect It" Proceedings of IEEE ICNP'00 page, 263-272, 2000.

[6] J. Frank "Artificial Intelligence and Intrusion Detection: Current and Future Directions" Proceedings of the 13th National Computer Security Conference, 1994.

[7] J. D. Huba, et al., "On the Role of the Lower Hybrid Drift Instability in Substorm Dynamics" J. Geophys. Res., 86, 5881, 1981.

[8] L. Heberlein, G. Dias, K. Levitt, B. Mukherjee, J. Wood, and D. Wolber "A Network Security Monitor" IEEE Symposium on Research in Computer Security and Privacy, 1990.

[9] Y. Huang, C. Kintala, N. Kolettis and N. D. Fulton "Software Rejuvenation: Analysis, Module and Applications" Proc. Of FTCS-25, Pasadena, CA, Jun. 1995.

[10] I. T. Joliffe "Principal Component Analysis" Springer-Verlag, 1986.

[11] J. Knight, K. Sullivan, M. Elder, and C. Wang "Survivability architectures: Issues and approaches" In Proceedings of the 2000 DARPA ISCE, pages 157 - 171, CA, June 2000.

[12] W. Lee and D. Xiang "Information-theoretic measures for anomaly detection" In Proc. 2001 IEEE Symposium on Security and Privacy, Oakland, CA, May 2001.

[13] A. Mena and J.Heidemann "An Empirical Study of Real Audio Traffic" IEEE Infocom, March 2000.

[14] B. Mah "An Empirical Model of HTTP Network Traffic" IEEE Infocom, pages 592-600, April 1997.

[15] J. Morul "The case for persistent-connection HTTP" ACM SIGCOMM, pages 299-313, August 1995.

[16] E. Oja. "Neural networks, principal components, and subspaces" International Journal of Neural Systems, 1(1); 61-68, 1989.

[17] Porras and A. Valdes "Live Traffic Analysis of TCP/IP Gateways" Networks and Distributed Systems Security Symposium, March 1998.

[18] R. S. Pressman "Software Engineering A Practitioner's approach" ISBN0-07-365578-3, 2001.

[19] D. W. Scott "Multivariate Density Estimation: theory, practice, and visualization" John Wiley & Sons, Inc., New York, 1992.

[20] S. Sekar, M. Bendre, and P. Bollineni "A fast automaton-based method for detecting anomalous program behaviors" In Proc. 2001 IEEE Symposium on Security and Privacy, Oakland, CA, May 2001.

[21] K. Thompson, G. Miller, and R. Wilder "Wide-Area Internet Traffic Patterns and Characteristics" IEEE Network, November 1997.

[22] G. R. Wright, W. R. Stevens "TCP/IP Illustrated" ISBN 0-201-63354-X, March 1996.

[23] T-Li. Wu "Securing Internet QoS: Threats and Countermeasures" Ph.D. Thesis, North Carolina State University, 1999.

[24] www.software-rejuvenation.com

[25] http://www.ll.mit.edu/IST/ideval/data/data_index.html

[26] ftp://ftp.rfc-editor.org/in-notes/rfc793.txt