Ad hoc Software Rejuvenation for Survivability

Khin Mi Mi Aung*, Jong Sou Park*
*Department of Computer Engineering Hankuk Aviation Univ.

ABSTRACT

We propose the model of Software Rejuvenation methodology, which is applicable for survivability. Software rejuvenation is a proactive fault management technique and being used in fault tolerant systems as a cost effective technique for dealing with software faults. Survivability focuses on delivery of essential services and preservation of essential assets, even systems are penetrated and compromised. Thus, our objective is to detect the intrusions in a real time and survive in face of such attacks. As we deterrent against an attack in a system level, the Intrusion tolerance could be maximized at the target environment. We address the optimal time to execute ad hoc software rejuvenation and we compute it by using the semi Markov process. This is one way that could be really frustrated and deterred the attacks, as the attacker can't make their progress. This Software Rejuvenation method can be very effective under the assumption of unknown attacks. In this paper, we compute the optimum time to perform an ad hoc Software Rejuvenation through intrusions.

I. INTRODUCTION

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. Survivability focuses on delivery of essential services and preservation of essential assets, even when systems are penetrated and compromised.

We have proposed the intrusion tolerance system or survivability with Software Rejuvenation Methodologies (SWRMS), which is the methodology that can really frustrate, and tolerance the attacks, as the attacker could not make their progress.

We will apply our proposed methodology on existing systems with proper architectures. While Firewall and intrusion detection system are doing their functions, the resolver, our proposed model will analysis concurrently. SWRMS contains schedule and adhoc software rejuvenation. The schedule s/w rejuvenation will perform regularly to improve the system's availability. For the adhoc s/w rejuvenation, we have to decide the optimum time to perform s/w rejuvenation for survivability. In the critical conditions it will really effective to prevent the occurrence of more severe exploits. In this work, we have to make a precise decision of when and how to perform our methods according to our collected effective information.

Many researchers have recognized the need of Intrusion tolerance or survivability. The general principle of survivability has been completed by [11]. Recently, survivable software architectures has been done by [22], [16] has accomplish survivability of networks, [23] has presented the survivable storage stystems and [12] are working one survivability in database system.

In this paper, we address the system's dependency degree versus Software Rejuvenation to determine the optimal time to execute ad hoc s/w rejuvenation for Intrusion

Tolerant System and we compute it by using the semi_Markov process. To counter act the attacks' attempts or intrusions, we perform the rejuvenation 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.

II. Proposed Framework

New attacks are being formulated everyday. The problem with detecting new attacks is that existing attack knowledge has to be aggregated with current behavior and events to determine that misuse is taking place. For the anomaly detection requiring multiple source pattern recognition associated with known misuse and the equivalent of instinct.

In term of statistical technique, the attacks are characterized within the short intervals by triggering the hypothesized number of different changes and every transient states could be evaluate the flow probability for each attack's features and network service density. Itis more for visualization that having any real basis and most similarity measures work about the same regardless of model. Rigorous formal model attempts to predict the probability that a given trace will be relevant to a given query. Ranks retrieved traces according to this probability of relevance (Probability Ranking Principle) and rely on accurate estimates of probabilities for accurate results. In this section, we analyze these intrusions according to their state changes in transient period, based on cumulative security aging events.

We perform Software Rejuvenation in two-stage method of the Semi Markov Process, one is the ad hoc and the rest is the schedule. In the first stage, amount of time by sojourn time distribution and in the second stage, the distribution functions are determined by the probability. The process would be performed on a standby component if the system has redundant components, or on the active component if it can be determined that it is a transient failure versus a hard failure.More specifically, after the initial fault at the hardware layer, the controller/drive might retry t-times to eliminate transient and simple seek errors; then it might adapt the skew to attempt to recover the requested block. Ideally, we will provide the ability to analyze trends in system performance as well as transient and recoverable events in order to predict faults.

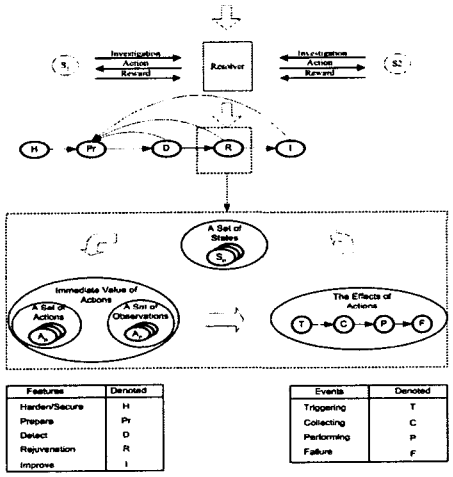

Fig.1:A framework of Ad hoc Rejuvenation of ITS

III.Software Rejuvenation intwo-stage method of the Semi Markov Process

We will work out the Rejuvenation time Vs. Interdependency by tracing the collected traces of data from that run of one day's traffic. Our system will be in monitoring mode that is genuine user traffic as it comes in and goes out. It also allows to sniff network packets and can be measured the response time and the packet loss percentages. It can also be viewed on TCP connection Establishment and Termination. We consider a system normal steady state asstate 1. It is initially in state 1 where it remains for a random amount of time having mean μ_1, then it goes to state 2, attack suspect state where it remains for a random amount of time

having mean μ_2, then it goes to state 3, attack penetration state where it remains for a mean time μ_3, then it goes to state 4, rejuvenation performing state, where it remains for a mean time μ_4, then back to state 1, and so on. The process is a cycle and completed each time the process returns to state 1.

The reward is the amount of time we spend in state i during that cycle, and then the above is a renewal reward process. The proportion of time that the process is in state i, is given by

$$P_i = \frac{\mu_i}{\mu_1 + \mu_2 + \mu_3 + \mu_4}, i = 1,2,3,4 \quad (1)$$

Similarly, our system had a process, which could be in any of N states $1,2,...,N$ and which moved from state $1 \to 2 \to 3 \to 4 \to ... \to N-1 \to N \to 1$, then the long-run proportion of time that the process spends in state i is

$$P_i = \frac{\mu_i}{\mu_1 + \mu_2 + ... + \mu_N}, i = 1,2,...,N \quad (2)$$

where μ_i is the expected amount of time the process spends in state i during each visit.

A process can be in any one of N states $1,2,...,N$, and that each time it enters state i it remains there for a random amount of time having mean μ_i, and then makes a transition into state j with probability P_{ij}. It is a semi-Markov process.

And we can denote as:

$N \to S$ = Random Interval when the Normal State Changes to Suspect State

$S \to U$ = Unavailable Interval After the state becomes the Suspect State, Service Unavailable State may be occurred

$U \to R$ = Rejuvenating Interval when the Rejuvena -tion Process is performing

$R \to N$ = Renewing Interval after Rejuvenation Process

c = Rejuvenation Perform Time We suppose that $N \to S$ is observable during the operation.

From the above four states, we get the following Distribution Functions

μ_{ar} = Ad hoc Software Rejuvenation

With finite mean $\mu_{ar} > 0$

State 4 to State 1 $R \to N$ (The System becomes to Normal State again)

$F_{arfin}(t)$ = Rejuvenation Finishing Time.

With finite mean $\mu_{arfin} > 0$

After completing the rejuvenation, the system becomes as good as new. As a result, we define the time interval from the starting to the ending of the system operation as one cycle, and the same cycle is repeated again andagain. If we consider the time to software rejuvenation time as a constant c, then it follows

$$D_{arfin}(c) = \frac{\mu_{N\to S} + \int_{N\to S}^{c} \bar{F}_{S\to P}(t)dt}{\mu_{N\to S} + \mu_{P\to R} F_{S\to P}(c) + \mu_{arfin} \bar{F}_{S\to P}(c) + \int_{N\to S}^{c} \bar{F}_{S\to P}(t)dt} \quad (3)$$

$$F_{ar\min}(t) = \min\{N \to S + S \to P + P \to R, R + c\} \quad (4)$$

$$F_{ar\max}(t) = \max\{N \to S + S \to P + P \to R, R + c\} \quad (5)$$

Finite Mean times of Rejuvenating Vs $\mu_{P\to R}$ Finishing μ_{arfin} increase, the rejuvenation time decreases; the maximum dependency degree also decreases. Higher the mean repair time relative to the mean time to perform rejuvenation, we should rejuvenate

the system more frequently.

Finite Mean times of Unavailable Vs Suspect $\frac{\lambda_{S\to R}}{\mu_{N\to S}}$, Mean Time to Failure becomes larger for a fixed time $\mu_{N\to S}$ i.e. the system tends to be more reliable, Performing Rejuvenation less and less frequently.

Our framework has two operations, Schedule and Ad hoc Rejuvenation Operations. Schedule Operation rejuvenation means that are persisted and execute constantly while Ad hoc Operation rejuvenation will be performed when it has to take the advantages of security management reconfiguration, vulnerability scanning, damage assessment and trend analysis.

Since the intrusion detection system can't detect all malicious attacks, performing rejuvenation is more challenging for survivability. Connections for TCP are well defined, because establishing and terminating a connection plays a central part of the TCP protocol unlikely with UDP. This transmission is termed as a request, even if in fact that application protocol being used is not based on request or rely. There are a number of generic procedures associated with TCP connections. We are emphasized on the flags, which reports a set of additional binary state associated with the connection. Many of TCP's actions, in response to different types of segments arriving on a connection, can be summarized in a state transition diagram.

Ⅳ. Conclusion and Further Work

Based on Statistical and Transient analysis, the attacks are characterized within the short intervals and every transient states 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 promising result outcomes, we can compute a Rejuvenation time versus Interdependency and average rejuvenation time versus dependency degree in face with attacks. By performing schedule and ad hoc Software Rejuvenation methods in the optimum time to counter act attacks'attempts or intrusions, 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. Restricting intruder access through software rejuvenation actions is a short-term solution.

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. And the purpose is to model and analyzethe system, which can continue being useful in face of attacks and will be operated through attacks.

This proposed approach is 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 a real time.

Acknowlegement

This research was supported by University IT Research Center(ITRC) Project and "Internet Information Retrieval(IRC)" Regional Research Center(RRC) Program.

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, E. A. Strunk and K. Sullivan "Towards a Rigorous Definition of Information System Survivability" IEEE DARPA DISCEX'03, pages 78-89, 2003

[12] P. Liu;Architectures for intrusion tolerant database systems, Computer Security Applications Conference, 2002. Proceedings. 18th Annual, Page(s): 311 320, 9-13 Dec. 2002.

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

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

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

[16] D. Medhi and D. Tipper. Multi-layered network survivability models, analysis, architecture, framework and implementation: An overview. In Proceedings of the 2000 DARPA Information Survivability Conference & Exposi-tion, pages 173186, CA, June 2000.

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

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

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

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

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

[22] V. Stavridou. Intrusion tolerant software architectures. In Proceedings of the 2001 DARPA Information Survivability Conference & Exposition, CA, June 2001.

[23] J. J. Wylie, M. W. Bigrigg, J. D. Strunk, G. R. Ganger, H. Kiliccote, and P. K. Khosla. Survivable information storage systems. IEEE Computer, (8):6168, August 2000.