제 19 회 한국정보처리학회 춘계학술발표대회 논문집 제 10 권 제 1 호 (2003. 5)

생존성을 위한 소프트웨어 재활기법의 모델

킨 미미 아웅*, 박종서*
*한국항공대학교 컴퓨터공학과
E-mail: maung@mail.hangkong.ac.kr, jspark@mail.hangkong.ac.kr

A Model of Software Rejuvenation for Survivability

Khin Mi Mi Aung*, Jong Sou Park*
*Dept. of Computer Engineering, Hankuk Aviation University, Seoul, KOREA.

요 약(Abstract)

The importance of Security measures by means of Physical Security, Network Security and Online/Internet Security. Lack of security is one of the primary obstacles in fielding many technologies in both commercial and DoD networks. Moreover, Internet censorship may be growing and is becoming increasingly sophisticated. In this paper, we will evaluate the Practice to Policy to Theory Approach for survivability by means of software rejuvenation models. These models would be assessed the effectiveness of proactive fault management in operational software systems and determined optimal times to perform rejuvenation.

1. Introduction

Security policies define the rules that regulate how our organization manages and protects its information and computing resources to achieve security objectives. Security policy has often been the most abstract and, for those untrained in advanced information security, the most obscure element of the information security field. Evidence based policy is more than just reading the results of research and applying those results to incidents because incidents have particular features that may make them different from the "average" incident studied in a trial. There are two types of differences. The first types of differences comprise those that affect probability (for example, the probability that actions will have the same absolute or relative effects as those measured in the trial). The second types of differences comprise those values (or utilities) that affect how much cost effective against the positive advantages of action.

Thus it is necessary for us to relate the results from a trial to get the good policies. A formal decision analysis provides an intellectual framework for developing an explicit decision making algorithm, which can be criticized and improved. Although, currently, time constraints make it unrealistic to conduct a separate decision analysis for each incidents and we have to find out the optimum one to overcome this problem. It is, however, feasible for decision analyses to be done for categories of incidents with similar symptoms.

An incident is a collection of data representing one or more related attacks. Attacker, type of attack, objectives, sites, or timing, may relate attacks. Based on these, we will evaluate the Practice to Policy to Theory Approach for survivability by means of software rejuvenation models. We will learn and implement the knowledge based opportunity from the real world evidences to be in tune with Rule Based policies and evaluate with the Software Rejuvenation Methodology. The relationship is also in the direction of practice to policy to theory for survivability. We are desired that the cost to be minimized. RFC 1244[4] contained an excellent of basic security policies and procedures in the table of contents. RFC 2196[1] is the current issue. Security Policy and Procedures range from high-level corporate policy to very detailed implementations. Our goal is to find out the specific ways which must be fulfilled the organization's requirements, efficiently and cost effectively.

Security invariant is the policy goal that we want to enforce in networks and the policy is the brain of the management platform.

2. Related Works

A number of security issues concentrate on Fault Avoidance, Fault free, Fault tolerant. Howsoever our methods of Software Rejuvenation applicable to Survivability, are the potential solutions to improve survivability instead of virtually impossible fault-free one. So that the system can reconfigure software and survive even under hacker's attack.

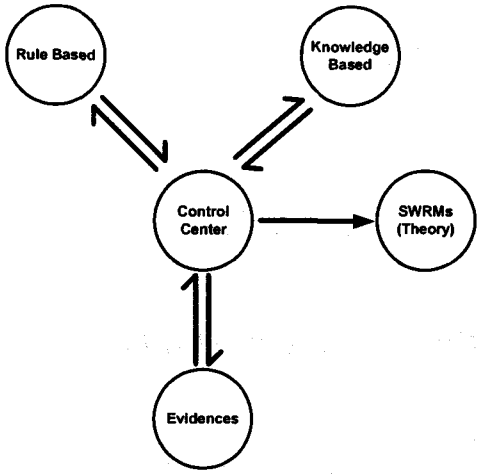

Fig. 1 A basic flow Diagram of Practic to Policy to Theory for Survivability

The ITUA project [3] relies on intrusion detection and randomized automated response. It provides middleware based on process replication and unpredictable adaptation to tolerate the faults that result from staged attacks. SITAR [2] is an intrusion-tolerant architecture for distributed services, and shares many of our goals and assumptions.

3. Proposed Approach

In this section, we will discuss a vulnerability case study and map it with our basic flow. There has been extensive work reported on categorization of software vulnerabilities, system errors and flaws, and intrusions. Having a security policy document in itself is not enough. The contents must be implemented to be effective. Now we will discuss about the cost effective model with the real conditions. Practice-based policy has been defined as the use of current best evidence in making decisions about the incidents, the planning and implementation of security services and the development of security policy. This approach to survivability is particularly concerned with the use of mathematical estimates of probability and risk in making survivability decisions.

Case study: Exploiting the Cold-fusion web administration page hole of the forum. index.cfm?<act=adm> can be found easily for a cracker with lot of free time and patience. Worse than biased, there was no authentication required due to bad integration. The abuser used admin page to create/change/add/edit/modify the forum contents. Debug page issues a lot of useful info such as ODBC name, table names, and fields. Then, Database Server can be compromised.

From such kind of evidance, we got some practices. E.g to test well the overall health and security of the site before going on-line, to remove debug options before going on-line, user rights to be properly defined and Authentication and logging to be deployed strictly and properly etc. This vulnerability amazes a high security risk, specifically, compromise of confidentiality. Exploiting in this vulnerability could access all types of data. The mapping of this vulnerability to our Practice to Policy to Theory approach for survivability is shown in Figure 2.

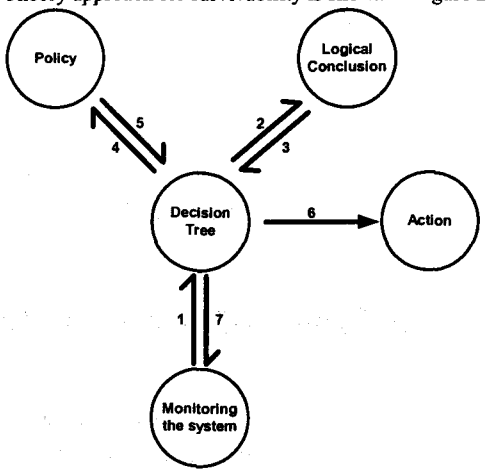

Fig. 2. A Model of Systematic Learning from Doing

From that model, we can get the various decision points and their consequences. After mapping (with their related probabilities and symptoms) to form a decision trees—a visual representation of the decisions available. Our aim is the logical reduction of a decision process into its individual decision points. Probabilities and symptoms are accumulated logically to determine the optimum policy (that is, the policy with the highest expected utility).

According to our practice to policy to Theory approach, we have been learned a framework for the survivability as Table1. And we determined to model with Schedule-based and Practice-based Software Rejuvenation Methodologies. To better assess the survivability of a system, the capability to adapt with decentralized managements instead of preventing attacks from penetrating the system is the more suitable approach. We intend our system to be able to detect and recover from intrusions automatically, to be augmented by measurements of the dynamic aspects of the system, to be effective and accurate and to be robust against malicious attackers. Although we may face with unknown attacks, we can develop intrusion triggers by focusing on only those events that pose a threat to the services under consideration instead of on arbitrary events. To perform this, we have to cope with the symptoms of corresponding Faults/Attacks first and then have to build a Framework by criticality, severity, or priority levels and finally we will propose scientific basis for probabilistically quantifying survivability.

At Figure 3, we perform a frame work of Software Rejuvenation Methodology. As a set of states, our model contains five states, Robust State, Monitoring State, Detecting State, Rejuvenating State and Failure State.

In the robust state we have to prevent to resist by various policies and offer proactive managements. For the Detecting State of rejuvenation performing stage, we need to be able to weigh the risk of policy with further damage against the policy of shutting the system in an emergency stage. At Monitoring State, the rejuvenation triggering stage, we will carry out with a various events[5] such as

SSR[1]	PSR[2]	Strategies	Detecting	Based On
Periodic Diagnostics and Automatic Error Log Schedule tasks (Checking Routine) Based on experiences to assess the approximate frequency of unplanned outages due to resource exhaustion Monitor server subsystems and software processes to ascertain common trends accompanying regular failures Error Logging and Alerts (Error Logging Controls) Essentially involves occasionally termination an application or a system, cleaning its internal state and restarting it. [1][Schedule-based Software Rejuvenation (SSR)]	Software Trace Service Aids package (monitoring selected system events) System Dump Facility Audit Event selection: per process and per object. Security Policy Events Subject Events, Object Events, Import/Export Events, Accountability Events, General System Administration Events, Security Violations (potential) [2][Practice-based Software Rejuvenation (PSR)]	How to organize online & anti hacking policies How to react to attacks and exploits How to communicate securely using all kinds of encryption technologies Trusted Computing Base responsible for enforcing the information security policies of the system, kernel, configuration, privilege Available Safety Features Maintain the security of information resources on a computer system. The detection and prevention of security violations on a system Managing Protected Resources with Access Control, addresses the privacy, integrity, and availability of information	The means to record security-relevant information, which can be analyzed to detect potential and actual violations of the system security policy. The auditing subsystem has three functions: event detection, information collection, and information processing	Basic Security Advanced Security Accounting Auditing Trusted Computing Base (TCB) OS Specific Security TCP/IP-Specific Security TCP/IP Command Security Trusted Processes The Network Trusted Computing Base (NTCB) Data Security and Information Protection

Table.3 Software Rejuvenation for Survivability: a Frame Work

monitoring selected system events, Audit Event selection per process and per object, Security Policy Events, Accountability Events and General System Administration Events. Based on those constraints and the systemtic learnings from doing, the system survivability must be improved.

As a consequence, the methodologies of how to organize online & anti hacking policies, how to react to attacks and exploits, and how to survive using all possibilities of software rejuvenation technologies are evaluated. Finally, Software Rejuvenation Methodologies are reviewed and synthesized. The main strategies are occasionally stopping the executing software, "cleaning" the "internal state" and restarting. These are the effectiveness of proactive managements and degrading mechanisms. We have to estimate the schedules, check pointing and expect the downtime cost, also.

The effects of the actions of our framework, an analysis phase, the symptoms of corresponding Faults and attacks are checked and analyzed as probabilistic information. The probabilistic approach provides a unified mathematical framework for correlating alerts that match closely but not perfectly, where the minimum degree of match required fusing alerts is controlled by a single configurable parameter.

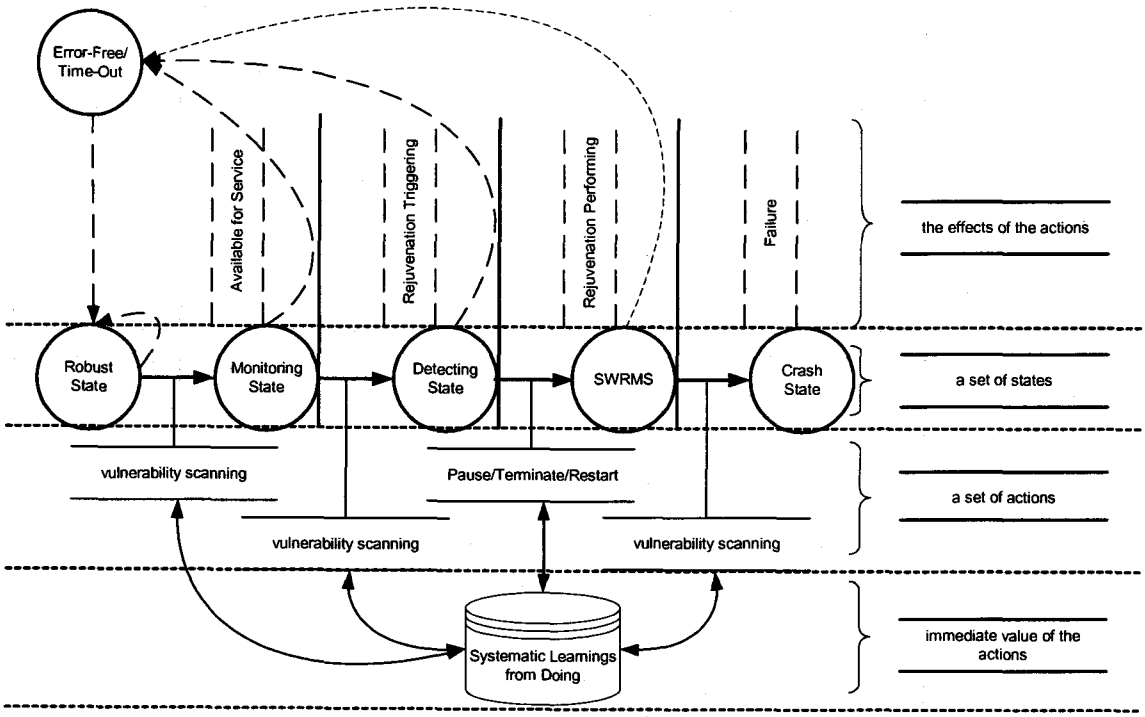

Figure. 3 Software Rejuvenation Methodology: A frame work.

According to the Markov's Decision Process, if the process is in state I at time n and action a is chosen, then the next state of the system is determined according to the transition probabilities state of the system is determined according to the transition probabilities $P_{ij}(a)$. If we let X_n denote the state of the process at time n and a_n the action chosen at time n, then the above is equivalent to stating that $P\{X_{n+1} = j | X_0, a_0, X_1, a_1, \ldots, X_n = I, a_n = a\} = P_{ij}(a)$.

4. Conclusion

Our plan is to incrementally grow the platform in each of the main components. In the security policy space, we will cover application level policies to move closer to higher level needs of security policy administration. Decision examination depends on probabilities and values, neither of which can be measured with assurance. It is impossible to incorporate and consider several components of a decision simultaneously. If evidence based policy is to be seen through to its logical conclusion and if both experiential evidence and the real world are to be incorporated into decision making, then this duality (the explicit collection of data v its implicit use) must be expressed.

Our ongoing work in Software Rejuvenation for survivability has led to the discovery of novel methodologies that are aware of changes in their environment can participate in their own defense, recognizing the symptoms of attacks and improving the detection and responses of security and defense mechanisms surviving on their behalf.

References

[1] B. Fraser, "Request for Comments: 2196", Site Security Handbook, September 1997.

[2] F. Wang, F. Gong, C. Sargor, K. Goseva-Popstojanova, K. Trivedi, and F. Jou. SITARIn Second IEEE SMC Information Assurance Workshop, 2001.

[3] M. Cukier, J. Lyons, P. Pandey, H. V. Ramasamy, W. H. Sanders, P. Pal, F. Webber, R. Schantz, J. Loyall, R. Watro, M. Atighetchi, and J. Gossett. "Intrusion tolerance approaches in ITUA" In Fast Abstract Supplement of the 2001 International Conference on Dependable Systems and Networks, pages B-64, B-65, July 2001.

[4] P. Holbrook, J. Reynolds, "Request for Comments: 1244", Site Security Handbook, July 1991.

[5] Real Secure server sensor policy guide version 6.5, December 2001.