American Journal of Computer Science and Engineering Survey
a Pubicon Open Journal

RESEARCH OPEN ACCESS

Modeling and Analysis of High availability Security Architecture for Whole of Government Systems

Pranita Upadhyaya[1], Tuan Anh Nguyen[2], Manish Pokharel[1] and Subarna Shakya[3]

[1]Department of Computer Science & Engineering /Kathmandu University, Kavre, Nepal
[2]Department of Computer Engineering, Korea Aerospace University South Korea
[3]Department of E & C and Computer, Pulchowk Campus /Institute of Engineering /Tribhuvan University, Pulchowk, Lalitpur, Nepal
Email Id:
pranitau@yahoo.com

Abstract

Failure in complex software or network based information server causes various problems. Recently, security is treated not only to prevent from intrusion but also as a Quality of Service (QoS) attribute. Software failure arises from random uncovering of faults during execution of successful inputs. We cannot predict all future intrusion and faults. To handle new type of attack based on application layer, Semi Markov Model is used to analyze & quantify the security attributes of the system.
In developing countries, e-government systems are in its infancy state, if attacks like DDoS occur citizens may be reluctant in using E-services. This paper compares various security architectures for whole of government systems and proposes the optimal solution. The model is analyzed for high availability using SHARPE software package .It focuses on impact rather than specific attack procedure.

Keywords- Security architecture, Whole of government, Semi markov model, Availability.

Introduction

Existence of weak legal protection of cyber world, threat to government services is increasing tremendously as organized crime all throughout the world. Maintaining relative anonymity & using various strategies, many are minting money illegally online. The proper remedy to it has yet to be discerned.

One of the most important responsibilities of any Government is to ensure the security and territorial integrity of the nation, including protecting the institutions that sustain confidence, good governance, and prosperity.
In order that this responsibility can be discharged, a Government requires its national security machinery to be well led, strategically focused, co-ordinated, cost-effective, accountable, geared to risk management, and responsive to any challenges that arise and to the needs of Ministers.

1.1 Related Work

E-government requires a great deal more than just a solid website that provides the right content. Behind every reliable and efficient application lies an extensive infrastructure of digital networks, Internet and application servers, databases and support services. The citizens expect high standards of services, instant access to information, efficient transactions and support, whenever and wherever they need it, but in a secure fashion. Rabah Alshboul *et al*[6] explored two main issues; vulnerability and security in e-government, discussed various important e-government security concerns. He investigated vulnerability and proposed some solution to achieve security. However, his ideas could not cater ever changing security threats.

WU Zhong[7] analyzed security technology and put forth an e-government security model based on PKI and SSL channel. According to him, the security requirements of e-government include confidentiality, integrity, non-repudiation & controllability. However, for ever changing pattern of attacks, this does not help either. In such case, Availability plays a major role. If no other counter measures are possible, we can at least increase the availability of that system.

Fengmin Gong *et al*.[8] presented a state transition model to describe the dynamic behavior of the intruder tolerant system. Even though they conduct case studies for various systems, they do not have analytical & quantitative assessment of operational security of any a prototype system and its evaluation through experimental measurements.

Felix Salfner *et al*.[9] contribute to software availability enhancement by offering a two-step strategy: Failure prediction (universal basis functions & similar event prediction) followed by maintenance actions (existing & adaptive recovery blocks) with the objective of avoiding impending failures or minimizing the effort of their repair. Finally, their effects on availability have been investigated. However, availability of the system as a whole did not pose very attractive results.

J. Grey *et al.*[3] explains ideas to assess high-availability system trends , provides the key concepts & techniques used to build high availability computer systems which are (1) modularity, (2) fail-fast modules, (3) independent failure modes, (4) redundancy, and (5) repair. According to him, these ideas apply to hardware, to design, and to software. They also apply to tolerating operations faults and environmental faults.

Here, we propose high availability security architecture suitable for whole of government systems. It is the design artifact that describes how the security controls (security countermeasures) can be positioned, and how they relate to the overall information technology architecture. These controls serve to maintain the system's quality attributes: confidentiality, integrity, availability, accountability and assurance services[1].

1.2 Security Architecture

Security architecture is a unified security design that incorporates the security requirements and the potential risks involved in a certain scenario or environment. It also stresses on when and where to apply security controls.

In security architecture, the design principles are reported clearly, and in-depth. Security control specifications are generally documented in independent documents. System architecture can be considered a design that includes a structure and addresses the connection between the components of that structure.

The architecture should be highly available so that it does not have the capacity to avoid security breach but tolerate and put the system into operation as far as possible to handle attacks like DDoS. (Refer figure 1)

1.3 High availability systems

According to Weibull[3], availability is defined as the probability that the system is operating properly when it is requested for use. In other words, availability is the probability that a system is not failed or undergoing a repair action when it needs to be used.

It is paradoxical that the larger a system is, the more critical is its availability, and the more difficult it is to make it highly-available. Building large systems involving thousands of modules and millions of lines of code is still an art. These large systems are a core technology of modern society, yet their availability is still poorly understood[3]. High availability is a system design approach and associated service implementation that ensures a prearranged level of operational performance will be met during a contractual measurement period. High-availability systems require fewer failures and faster repair. As the nines begin to pile up in the availability measure, it is better to think of availability in terms of denial-of-service measured in minutes per year. Availability class is the number of leading nines in the availability figure for a system or module[3]. Thus, our intension is to develop a whole of government security architecture which is highly available for use

even if there is an error. It is most effective for widely popular attack like DDoS, where availability of the system plays a major role.

We are proposing a security architecture which is suitable for Whole of Government (WoG) system. Now let us discuss the reason of our choice.

1.4 Whole of Government approach Details

The movement from isolated silos in public administration to formal and informal networks is a global trend driven by various societal forces such as the growing complexity of problems that call for collaborative responses, the increased demand on the part of citizens for more personalized and accessible public services, which are to be planned, implemented and evaluated with their participation, and the opportunities presented by the Internet to transform the way the government works for the people[2]. It enables governments to connect seamlessly across functions, agencies and jurisdictions to deliver effective and efficient services to citizens and businesses. It reduces security risk & better manage crisis & disaster. The absence of whole of government inhibits progress in many areas, notably in low income countries[2].

For such systems, security is the critical factor. Proper security architecture needs to be developed which caters the dynamic behavior of the intrusion. In other words, by knowing the present state we need to predict the future because we might not know the future attack patterns. Regarding it Semi Markov Model would be optimum.

1.5 Semi Markov Model

Markov chains are a relatively simple but very interesting and useful class of random processes. A Markov chain describes a system whose state changes over time. The changes are not completely predictable, but rather are governed by probability distributions. These probability distributions incorporate a simple sort of dependence structure, where the conditional distribution of future states of the system, given some information about present states. It depends only on the most recent piece of information. That is, it matters in predicting the future of the system from its present state, and not the path by which the system got to its present state. Markov chains illustrate many of the important ideas of stochastic processes in an elementary setting[4].

As in traditional approach, by applying a strategy just for avoiding the intruder would not help either. We have to make the system be tolerable even if the proper protection is breached. Intrusion tolerance is the ability of a system to continue providing (possibly degraded but) adequate services after a penetration[5]. It is an emerging approach to security. It aims to increase the likelihood that an application will continue to operate correctly in spite of malicious attacks. To evaluate the capability of a resilient system in surviving intrusions, quantitative evaluation models are needed. Furthermore, the evaluation models can be used to compare different intrusion tolerant

architectures. Most existing evaluation models assume that the intrusions are caused by random attacks. However, in many real cases the successful intrusions are caused by human intent which may not be a random event. Semi-Markov process is an extension of Markov chains and shows advantages in several aspects.
In Markov environment, the distributions for transitions between states have to be negative exponential functions. Instead in the semi-Markov case the distributions can be non exponential functions. This is the main advantage of semi-Markov processes. In addition, in Markov chain environment the time transition is given. But in the reality, the transition between two states in a security system sometimes happens after a random duration. This is one reason why the semi-Markov chain model is applicable real problems better than the Markov one[5].

This architecture is then analyzed using SHARPE software package. The numerical results showed very high availability.
The rest of the paper is organized as follows: Section 2 presents the proposed solution, and section 3 contains its analysis model. In section 4, numerical results are presented. Finally, we conclude the paper in section 5.

Proposed solution

The overall connected government security architecture is shown in Figure 1. Here, we cater the dynamic behavior of the intruder. Since all the attacks are not well known just focusing on the existing pattern and applying general protection mechanism is not enough. Based on the present scenario we need to perform future predictions. Thus, this model uses Markov Model to analyze future results. It tries to describe both known and unknown security exploits by focusing on impacts rather than specific attack procedures.

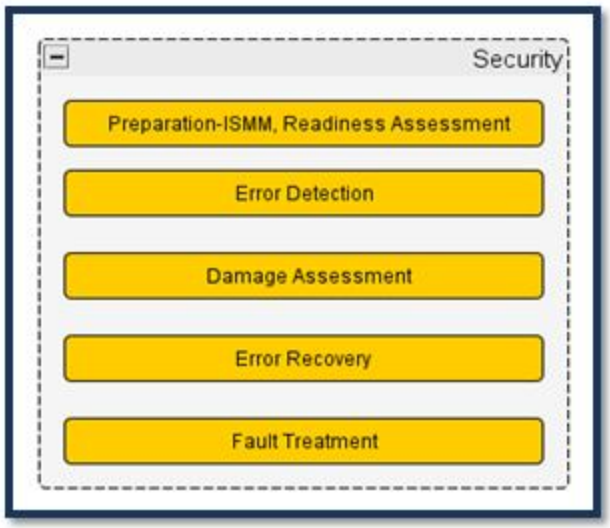

Fig 1: Security Architecture

As shown in the figure 1, there are total of five subsystems in the security architecture. It can also be taken as five levels. The first level is preparation. At this stage, as per the type of service whole of government systems provides to the customer, its security level is identified and provided accordingly[9]. Besides, the security levels has to be monitored on a regular basis i.e. security readiness assessment needs to be done[10]. In other words, vulnerability checking is done at this stage. The system is in vulnerable state if it allows user to read & modify information without authorization which is the violation of the security policy. If not recovered, i.e. exploiting vulnerability, system enters into active attack stage & damage may follow. The error needs to be detected. Thus, enters into the next level i.e. error detection stage. If the error cannot be detected and recovered; resistance to attack fails. Once, the system enters this stage damage may follow. At this stage, the system should be capable of increasing the ability of the system to tolerate even if it is attacked. If so, damage type needs to be assessed, for which the next level is reached i.e. damage assessment stage. Using various methodologies like, identifying multiple attack patterns and masking it, error recovery is performed. If this level is also violated, system enters into fault treatment stage which may be recovered by manual intervention.

Analysis Model

The analysis model for the above architecture is drawn using SHARPE software package in figure 2 and traditional approach is drawn in figure 3. Initially all the components (five) mentioned above in security architectures are in the up states. Nodes depicted as UUUUU. If all the preparative measures are not taken, the system becomes vulnerable and may be subjected to attack. The system thus reaches DUUUU state by failure rate □1. Software rejuvenation technique [11] is applied to the system. This state is named RUUUU. Here, some portion of the software gets rejuvenated by rate µ1 and the remaining portion which could not get recovered by software rejuvenation technique called FUUUU state FUUUU i.e. the first component failure state, gets repaired by a repair person by rate µ2. This is the state where vulnerability is breached. This has a very high probability of being attacked. If it gets attacked, the second component gets defeated and the state becomes FDUUU. If this goes for rejuvenation the system reaches FRUUU state which gets repaired by rate µ3, if not, reaches the state FFUUU. It can be repaired by a repair person by rate µ4. Suppose at this state, an error cannot be detected, then the type of damage needs to be assessed. On assessing, If this stage is detected the system reaches FFDUU state. If it goes for rectification by software rejuvenation technique, the state is called FFRUU and gets rejuvenated by rate µ5. If not system reaches FFFUU state where 3 of the components are down. It can be recovered by manual intervention by rate µ6.

 ISSN: 2349- 7238

Similarly, when the entire component goes down, the system reaches FFFFF state. This state can only be recovered by manual intervention.

Table: 1. Data sets for analysis

Params	Description	Value(min)
	Total Time assumed	6000
1/□1	Meantime for application failure	2000
1/δ1	Meantime for failure detection	0.5
1/δ2	Meantime for s/w rejuvenation failure	0.5
1/μ1	Meantime for vulnerability repair by s/w rejuvenation	13
1/μ2	Meantime for vulnerability repair by manual intervention	20
1/ □2	Meantime for vulnerability failure	1500
1/ μ3	Meantime for error repair by software rejuvenation	13
1/ μ4	Meantime for error repair by manual intervention	60
1/□3	Meantime for damage assessment failure	1500
1/ μ5	Meantime for damage assessment component repair by s/w rejuvenation	13
1/ μ6	Meantime for damage assessment component repair by Manual Intervention	100
1/ □4	Meantime for error recovery failure	1000
1/μ7	Meantime for error recovery repair by s/w intervention	13
1/ μ8	Meantime for error recovery repair by Manual Intervention	140
1/□5	Meantime for fault treatment failure	1000
1/μ9	Meantime for fault treatment failure recovery by s/w rejuvenation	13
1/μ10	Meantime - fault treatment failure recovery by manual intervention	200

Figure 3 shows the analysis model for traditional approach. It is the scenario of regular security architecture of e-government based on the principal of fault avoidance.

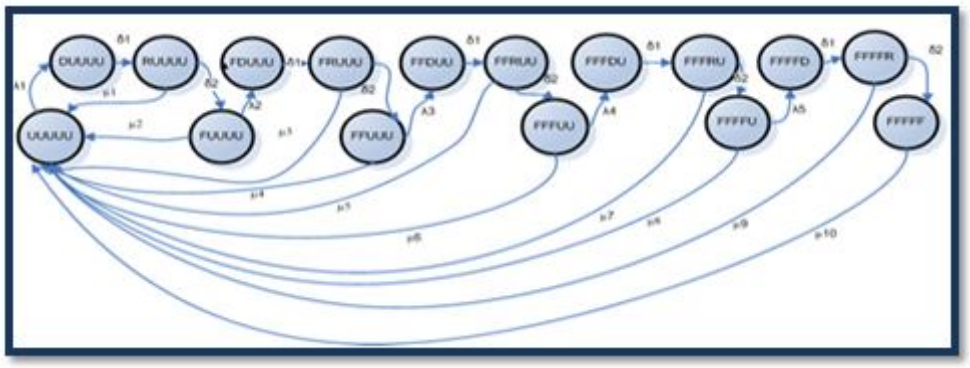

Fig 2: Analysis Model for Security Architecture

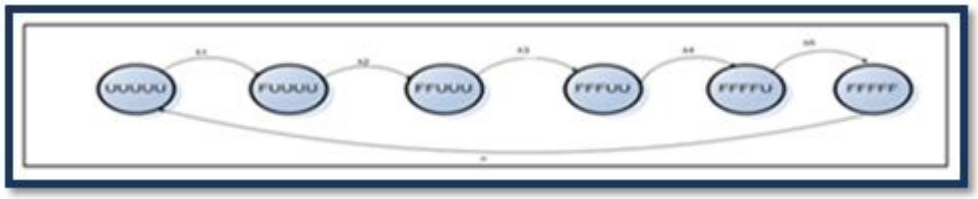

Fig 3: Analysis model for the traditional approach

Numerical Results

Based on the analysis model presented in figure 2 & 3, we illustrate the evaluation of the security attributes. It is considered that a successful attack that brings the system from vulnerable state to an active attack state is less likely than detection of the attack and bringing the system back from vulnerable to good state. The system spends more time in the good and vulnerable states than in the active attack state A. Thus, system stays in good state half of the total time, 1/3rd and 1/4th of the time in active attack state, damage assessment state & 1/6th in the remaining stages[15]. The time difference between 1st and the second repair (manual intervention), 3rd and 4th and so on increases by 40 mins each time[14]. Based on the above mentioned collected data sets[14,15] table 1 is prepared .Table 2 shows the output that is obtained using SHARPE software package for two different scenarios; 1. Using of software rejuvenation technique 2. Using Traditional approach. The graphical representation is shown in figure 4. Software rejuvenation is a primary method to fight software aging. Among the various

methodologies of rejuvenation, virtualization is considered to be one of the effective techniques. It involves stopping the software application, cleaning its internal state and/or its environment and then restarting it. By removing the accrued error conditions and freeing up or decrementing operating
System (OS) resources, this technique proactively prevents unexpected future system outages. It is widely understood that this technique of rejuvenation provides better results, resulting in higher availability and lower costs[15].

Table: 2. Outputs from SHARPE

Steady stage availability	Output from SHARPE
With the use of software rejuvenation technique	9.9999 e-001
With traditional approach(applying fault avoidance only)	4.9864 e-001

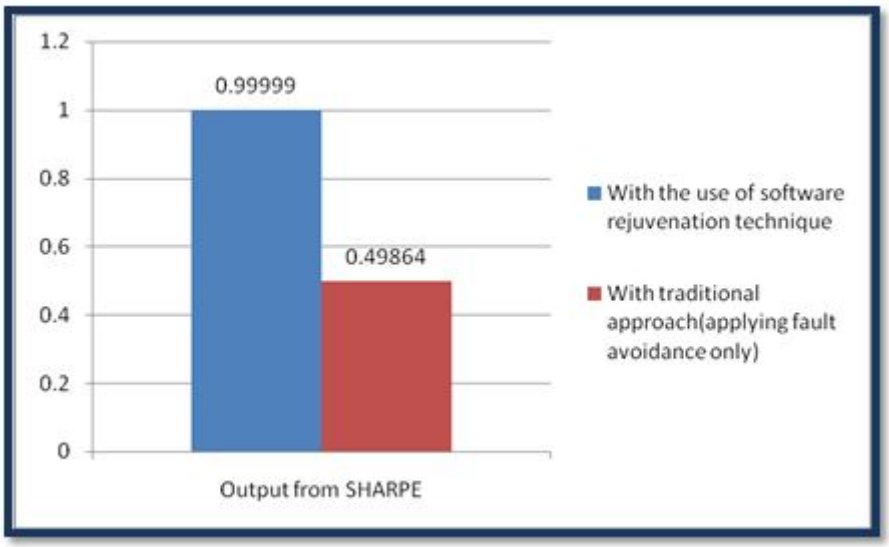

Fig 4: Graphical Representation of output from SHARPE

Conclusion

Detection and response research has so far mostly concentrated on known and well identified attacks. To cater ever changing pattern of attack needs to be dealt with a different security approach which incorporates dynamic

behavior of the system. In this model, security architecture provides architecture which can define vulnerability and threat by focusing on the attack impact rather than specific attack procedures. Incorporating the known security related faults and predicting the future probability of attack by using semi Markov model, and by the use of software rejuvenation technique, we deduced a high availability of security system architecture which handles a complete fault space of the present and future faults cost effectively.

Yet a detailed analysis of applying this security architecture as a subsystem to whole of government system architecture is lacking. In future, we propose to conduct further research on implementation of this architecture on whole of government system measure the total availability.

References

1. Definitions: "I T Security Architecture". Security Architecture. org, Jan, 2006
2. "United Nations E-Government Survey 2012. E-Government for the people. "
3. J. Gray and D. P. Siewiorek, ''*High-Availability computer Systems*,'' Computer 24, No. 9, 39–48, 1991.
4. J. Chang, "Markov Chains", March 30, 1999.
5. Alex Hai Wang *et al*.,"A Semi-Markov Survivability Evaluation Model for Intrusion Tolerant Database Systems", 2010 International Conference on Availability, Reliability and Security.
6. Rabah Alshboul *et al*, "Security and Vulnerability in the E-Government Society ", *Contemporary Engineering Sciences*, Vol. 5, 2012, no. 5, 215 – 226.
7. Shun-Chieh Lin *et al*, "Constructing detection knowledge for DDoS intrusion tolerance, *Expert Systems with Applications*", 27 (2004) 379–390
8. Felix Salfner *et al*, "Prediction-Based Software Availability Enhancement", springer link Self-star Properties in Complex Information Systems , *Lecture Notes in Computer Science* Volume 3460, 2005, pp 143-157
9. Pranita *et al*, "Information Security Framework for E-Government Implementation in Nepal ", VOL. 3, NO. 7, July 2012 ISSN 2079-8407 , *Journal of Emerging Trends in Computing and Information Sciences*
10. Pranita *et al*., "E-government Security Readiness Assessment for developing countries. Case study: Nepal Govt. Organizations", 978-1-4673-2590-5/12©2012 IEEE
11. DOMENICO COTRONEO *et al*," A Survey of Software Aging and Rejuvenation Studies", *ACM Journal on Emerging Technologies in Computing Systems*, Vol. V, No. N, Article A, Pub. date: January YYYY.
12. Rabah Alshboul, " *Security and Vulnerability in the E-Government Society Contemporary Engineering Sciences* ", Vol. 5, 2012, no. 5, 215 – 226

13. Kiejin Park *et al*, "Availability analysis and improvement of Active/Standby cluster systems using software rejuvenation" , *The Journal of Systems and Software* 61 (2002) 121–128
14. Dong Seong Kim *et al*, "Availability Modeling and Analysis of a Virtualized System ", 2009 15th IEEE Pacific Rim International Symposium on Dependable Computing.
15. Thandar Thein *et al*., "Improving Fault Tolerance by Virtualization and Software Rejuvenation", Second Asia International Conference on Modelling & Simulation. 978-0-7695-3136-6/08 © 2008 IEEEDOI 10.1109/AMS.2008.75