AUTOMATED CONTROL FOR RISK ASSESSMENT ON UNIX OPERATING SYSTEM -I

[1]Dr. Prashant Kumar Patra & [2]Padma Lochan Pradhan

[1]Dept. of CSE, College of Engineering & Technology, BPUT, Bhubaneswar-751003
[2]Dept. of CSE, Sikha 'O' Anusandhan University. Bhubaneswar, Orissa, India

ABSTRACT

Control and Risk are the two parts of the coin. Risk management is the process of identifying vulnerabilities and threats to operating system resources to achieving business objectives and deciding what counter measures to take in reducing the lowest level of risk. The increased use of computer & communications system by IT industries has increased the risk of theft of proprietary information. Cryptographic control (Encryption) is a primary method of protecting system resources. Automated Control is probably the most important aspect of communications security and becoming increasingly important as basic building block for computer security. Automated Control is inversely proportional to the risk & mean while control is directly proportional to the quality of standard (S). Automated Control provides accountability for individuals who are accessing sensitive information on application, system software, server and network. This accountability is accomplished through access control mechanisms that require identification, authentication, authorization, non-repudiation, availability, reliability & integrity through the audit function. We have to develop java automated control for risk optimization based on unix operating system.

Pyramid Diagram :

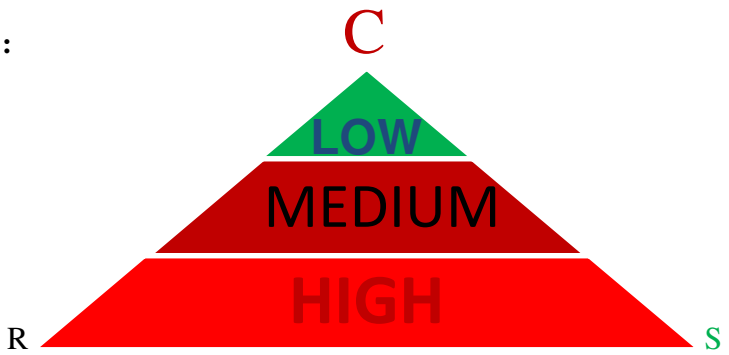

Fig: 1 [Red Color: High Risk, Purple Red: Medium Risk, Green: Low Risk]

Keyword

AC-Automated Control, PC- Preventive Control, SSH- Secure Shell, AES-Advance Encryption Standard, CBC: Cipher Block Chain, CERT-Computer Emergency Response Team, DC-Detective Control, CC-Corrective Control, CMDB-Change Management Database, CKM-Cryptographic key management.

Introduction

Risk assessment is the first process of the risk management methodology. Organizations use risk assessment to determine the extent of the potential threat and the risk associated with an IT system and sub system throughout its life cycle as the steps follow. The output of this process helps to identify appropriate controls (PC, DC &CC) for reducing or eliminating risk during the risk mitigation process, as discussed in risk mitigation model & method. Risk is a function of the likelihood of a given threat-source's exercising a particular potential vulnerability, and the resulting impact of that adverse event on the organization. To determine the likelihood of a future adverse event, threats to an IT system must be analyzed in conjunction with the potential vulnerabilities and the controls in place for the IT system. Impact refers to the magnitude of harm that could be caused by a threat's exercise of vulnerability. The level of impact is governed by the potential mission impacts and in turn produces a relative value for the IT assets and resources affected (e.g., the criticality and sensitivity of the IT system components, devices and data). The risk assessment methodology encompasses eleven primary steps, which are described as follows:1.System Characterization, 2.Risk Identification, 3.Vulnerability Identification,4.Risk Mitigation,5.Risk Analysis, 6.Control Analysis,7.Likelihood Determination, 8.Impact Analysis,9.Risk Determination,10.Control Recommendations,11.Results Documentation. [12]

Operating System control is a step by step process of securely configuring a system to protect it against unauthorized access, while also taking steps to make the system more reliable. Generally anything that is done in the name of system. Preventive control ensures the system is secure, reliable and high available for high IT culture. Operating system control is the process to address security weaknesses in operation systems by implementing the latest OS patches, hot fixes and updates and following procedures and policies to reduce attacks and system down time men while increase the throughput of the system. Preventive control of the operating systems is the first step towards safeguarding systems from intrusion. Workstations, applications, network and servers typically arrive from the vendor, installed with a multitude of development tools and utilities, which although beneficial to the user, also provide potential back-door access to the systems.
Security is a process, not a result. It is a process which is difficult to adopt under normal conditions; the problem is compounded when it spans several job and application running simultaneously under complex based web infrastructure which using million of user accessing the same piece of devices and information around the globe. All the system level security in the world is rendered useless by insecure web-applications.

Literature Survey: [5,12,16,17]

The technical literature survey in IS Security area is very critical & tedious task to collect the actual data and evidence in the real life. It is one of the on going processes in a continuous manner. It is a very time consuming to investigate & judge the information. There are many text book & reference books help to us to find out the real issue. The reference books like: Applied Cryptography by Bruce Schneier & Cryptography & Network Security by William Stalling is very much help full to expand our idea. The object oriented java programming is very help to make programming for cryptographic key management issue. The Sun Micro-system UNIX sun solaris system administration guide: Vol 1 & Vol 2. & O' Reilly, Essential of System Administration is very helpful to collect the basic data.

In our past experience, we observed on operating system as well as network server, there are many system parameters are defaults and are not developed and mis-organize many more issue which is very high level risk to the Information System. This issue is high lighted in our action plan and proposed method.We have to find out some method to make the more efficient, secure, high available & robust high end operating system. No where develop the detail methods in Graphically as well as Mathematically about the risk management of the OS . There are many issue are not develop till now like : System Characterizations, Risk Identification, Risk Analysis, Vulnerability Identification,Risk Mitigation, Control Analysis, Impact Analysis, Risk Determination, Control Recommendations & risk results in the operating system level. We have to develop Risk Identification, Risk Analysis, and Risk Mitigation in both analytical & graphical way. There many documents are available in general sense of risk identifications, risk analysis, risk mitigation, but operating system level the classification & categorization of risk is not available on today itself. We have to focus on the system specific like OS & system software to develop automated control based on java programming.

Preventive Controls [16,17]:

Preventive controls are implemented to stop the loss related to a risk from occurring when the risk situation presents itself, the preventive control kicks in and prevent the loss. Preventive techniques are the most complete form of stop loss control, because the loss is prevented by their nature. There are costs associated with preventive control that must be considered to get the full picture of the impact to the business. Prevention can mean the continuous close examination of each case, performing on analysis for the risk condition and stopping the risk whenever it is identified. This can be more expensive way to control than simply enabling the process to perform, identifying errant exceptions after they have occurred and taking them out of the process stream for corrective action in due course of time. While attempting to prevent errors whenever it is cost effective to do so, many production lines in the manufacturing sector also use detective technique to weed out errors, which is a more cost effective way of dealing with all of the possible permutations of error conditions that may exist in the process. The alternative of building preventative controls for each scenario would be cost prohibitive. The monitoring & management of the preventive controls also will need to be considered when determining what is best for the business.

Detective Controls [5]:

Detective controls are used in situations where it is more important to understand that some things has happened that it was to prevent from happening. In some cases, a detective control will ensure that a desirable event did indeed occur, providing feedback that the process is working as intended. Evaluation of the detective controls require proving that the detection occurs with a high degree of accuracy and reliability. When it is important to detect that an action has occurred, it will be equally important to rely on the control to not miss any valid occurrences where that detection should be taking place and to flag only those valid occurrences of predefined interest. To assess these controls, we will need to understand the trigger event and the mechanism used to identify it. The risk associated with detective control are the risk of not knowing a situation or event has occurred .If this failure to detect happens regularly , the control cannot be assessed as defective. When evaluating the cost-benefit for this control type , we must review what happens to the process if the events or situation is not detected and then assess the costs of this scenario

against the cost of developing, implementing and maintaining the control. All system based logs automated generate on the development & production server.

Corrective Controls [5,16]:

A corrective control fixes errant situations or events as they are identified. It assumes some amount of detection is inherent in its mission of fixing out-of-bonds conditions .These controls are useful when simple corrections are easily found and fixed a process without lot of risk and complexity. The risk of not finding and fixing these items must be considered when assessing the total cost and benefit of such a control. It will need to be determined that corrective actions are possible and performed accurately to the satisfaction of the process in order to draw conclusions that these kinds of controls are effective. Determining what is acceptable in terms of corrective actions will be part of this process. Those situations that are not caught and fixed that does not require attention will need to be identified and examined for false positive and false negative implications. Comparing this control to one that prevented the need for correction in the first place may be valid assessment when evaluating whether the right kinds of controls are employed to mitigate risk in a process. The cost to fix along with cost to identified or cost to prevent all will now need to be part of the cost benefit analysis.

DATA COLLECTION BASED ON EXISTING CONTROL: (BASIC DATA)[7,8,18,19]

There are number of hardening and control methods developed as per requirement of the secure computing to achieve the highest level of business objective. There is a few method developed based on UNIX server (UNIX operating system programming).

SN	SYSTEM FILES	ACTION PLAN	REMARKS
1	/etc/system	implement the kernel & n-bit processer	Can be improve the system performance
2	/etc/hosts	Develop the scripts: allow/disallow as per policy, chmod 000= /etc/marshmdisallow	preventative control
3	/etc/services	Disable the third parties services. Remove the ftp, http, telnet, port no, printer, IP services. Those services are not required.	preventative control
4	/use/bin/rsh, etc/pam.conf	Disable all remote services: chmod 000 /usr/bin/rsh, rksh,rcp, ruser,rlogin, uptime.	preventative control
5	/var/adm/message	Date & time stamp	Internal audit purpose

6	/etc/rc.conf script	Run level script Run level script have to develop as per requirement. /etc/init.conf,rc2.d example:httpd_flags="NO"	preventative control
7	/etc/init.d	OS services, run level	preventative control
8	etc/ssh/sshd_config **Automated Control**	Cryptography enable through ssh implementation AES: 256 bits cipher blowfish-CBC,aes256-CBC, aes256-chr.ssh-key gen -b 1024 -f /etc/ssh_host_key -n " chmod - - - /etc/ssh/ssh_config	preventative control n=1024, 2048, 4096 chmod r w x (i. e. 4 2 1) – blank is nothing

Table: 1Unix file system have to be develop as per business requirement: (DERIVED DATA)

Problem in existing control: [7,8]

The problem is compounded when it spans several jobs and applications running simultaneously under complex based web infrastructure which using million of user accessing the same piece of devices and information or data on the around the clock (24 x 7 x 52). All the system level security in the world is rendered useless by insecure web-applications. The ssh key 1024 & AES key 256 is limited shorter key size. The system throughput became slow down, slow down the network resources & loss of communication system. There is no balance ratio among the processor, memory & time slot of the multi-tire system.

Clients & Sever (Master ~Slave) Many to One concept. Increasing the hackers as well as

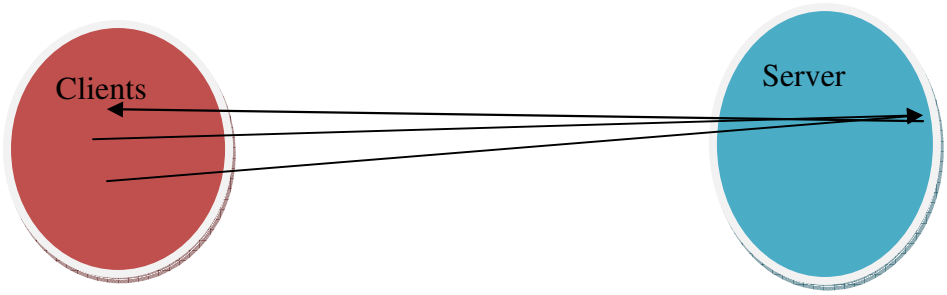

Fig:2experiences hackers in client server environment.

External Intruders

Internal Intruders

Why we use stronger (larger) key sizes? Because, we have to develop automated control based on java based system programming.

Problem in existing control: due to shorter keys size
That's why we are using stronger key: AES 512, 1024, 2048 & so on.
Increasing the business (large volume of data, Information, data warehousing & data mining).

- Increasing the million of users.
- Increasing the hackers as well as experiences hackers.
- Increasing the hardware & software capabilities (n-th bits processor & no of CPU, Memory).

Our proposed cryptographic key management (encryption) control will be help to :

- Business continuity planning & disaster planning
- Internal & external system audit.
- Keep the system balance among devices, sub-systems, resources & users need.
- Improve the throughput, interoperability, CPU utilization, total cost ownership (TCO) & Return on Investment (ROI)
- The least cost & best fit approach.

PROPOSED AUTOMATIC CONTROL:

CRYPTOGRAPHIC ENCRYPTION CONTROL (ANALYTICAL METHOD)

Cryptographic Key Management (CKM). Cryptographic keys must be securely managed when cryptographic functions are implemented in various other controls. Cryptographic key management includes key generation, distribution, storage, and maintenance purpose. This is a measure preventive control in security world around the globe. As a brief, in the operational security domain, preventive controls are designed to archive two things: to lower the amount and impact of unintentional errors their are entering the system and to prevent unauthorized intruders from internally or externally accessing the system. An example of these controls might be renumbered forms or a data validation and review procedure to prevent duplications. How the operating system maintaining ratio & proportion among various sub systems like server key, encryption

key, processor & memory capability, availability and efficienciency. This issue is high lighted in our action plan. We have to find out some method, to make the more efficient, secure, high available & reliable the robust high end operating system. The SSH key in the existing Unix based operating system support only up to 1024 etc /ssh/sshd_config (Cryptography enable through ssh implementation AES: 256, bits chipper, chipper blowfish-CBC,aes256-CBC, aes256-chr.). The existing system supporting only 1024 in SSH key & 256 key size in AES Level. These AES-256 and SSH-1024 is not sufficient for high end processor, CPU, Memory, instruction pipe lines (SIMD, MISD, MIMD) . But, the propose control will be facilitate and resolve the various problems when it spans several jobs and applications are running simultaneously under heterogeneous complex based web infrastructure & mobile computing environment, which using million of user accessing the same piece of data around the clock(24 x 7 x 52).

Fig:3 Flow diagram of detail work: system program (proposed method)

- Prevent: Preventive controls focus on preventing security breaches from occurring in the first place.
- Detect and Recover: These controls focus on detecting (DC) and recovering (CC) from a security breach.
- Corrective Control: These controls focus on corrective control, that' why we call automated control. (CC)

Mathematically: AC : (PC+DC+CC), ACα1/R, AC α S. AC=k. 1/R, AC=k. S

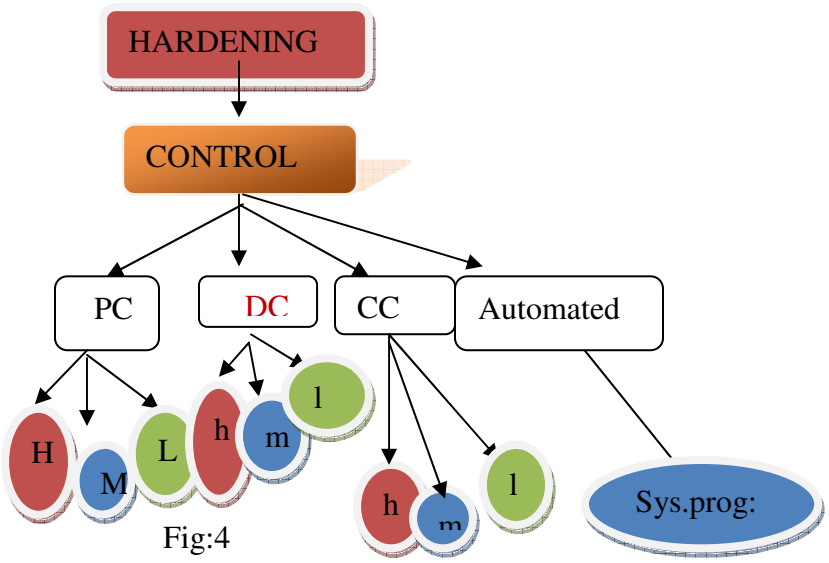

Fig:4

Proposed Control (Cryptography based system control): Our Action Plan

Aim and Objective of the Thesis:

a).To investigate the preventive control and technique to safe guard our operating system that should be free from risk.
Our objective is that, to find out the maximum policy, procedure, tools, commands & technique how to safe guard our assets from internal and external hacker as well as misuse of the critical

infrastructure components like (N-tire architecture) network components, operating system , server, data and file, database, middleware, application, scanner, printer and storage devices. Therefore, the following sub systems have to protect as per business requirement. Prevention is the first step of the risk assessment of the critical IT infrastructure. We are going to develop many more hard core prevention methods to protect the multi-tire architecture for web based application, which is facilitating million of customers. We have to maintain the integrity & privacy of the sub system for multi-core operating system as well as current virtualization technology. There are four level of prevention control methods are require to minimize the risk of any web based multi-tire infrastructure.

Operating system control: (Sun Solaris, AIX, LINUX, HPUX, Microsoft OS)
Network control: (Load balancing, Clustering, DNS, Firewall, Proxy, Squid, SSH, SSL, FTP, LDAP)

Database control:
Application Control :(cryptography, mod-ssl, mod-security, mod_autho, vintila)

b). Minimize the Risk: (PC=K/R)
Meanwhile we have to investigate and audit the system security for detection, prevention & corrective action for minimize the various level of risk: like : High, Medium & Low. Risk is never eliminated, only we can able to minimize into lowest level. We can minimize the risk on IT infrastructure in the three ways of doing continues ways process of protection, detection and risk assessment method. But, risk never be eliminated. Risk can be transferable and minimize. We should always optimize the risk at the minimum level to protect our critical IT infrastructure and assets as well as sub systems (OS : processor, multi-plexier, memory and other related components).

In this paper, we are focusing only **operating system level** : (Advanced Cryptography Encryption control)

Automatic Control is directly proportional to AES-M. (PC=k .AES-M) [Where M=512, 1024 …2048..]

where k is the constant factor, then automatically reduced the risk. But, there is some limitation of architecture of processor & memory (instruction pipe lines: SIMD, MIMD, MISD) capability of the operating system for large based key generation of AES & CKM.

PROPOSED ALGORITHM [DEFINE & DESIGN STAGE] FOR AUTOMATED CONTROL:

Advanced key expansion algorithm: etc/ssh/sshd_config
The advanced key expansion algorithm scheme takes as input a 4-word (N), (16-byte) key and produces a linear array of 44 words (176 bytes). This is insufficient to provide a 4-word round key for the initial add round key stage and each f 10 rounds of the chipper. The following pseudo code describes the expansion. Let us consider N = 4, 8, 16, 32……
Advanced key expansion (byte key {16 }, word w [44])

```
{
 word temp
 for ( i = 0;    i  < N; i++ )
 w[i]  =  ( key [ N*i],  key [N*i+1] ) ,  key [N*i+ 2] , key [N*i+ 3] );
 for  ( i = N;  I < 44;  i++ )
{
          Temp = w [ i   - 1 ];
           if  ( i mod  N = 0 )    temp   =  SubWord ( RotWord  (temp ) )   Rcon⊕      [ i/N ];
           w[ i ] = w [ i - N ] = temp
}
}
```

The key is copied into the first four words of the expanded key. The reminder of the expanded key is filled in four words at a time. Each added word w{i} depends on the immediately proceeding word, w[i – 1], and the word four positions back, w[i – N]. In three out of four cases, a simple XOR is used. For a word whose position in the w array is a multiple of 4 (i.e N), a more complex function is used.

CODING & DEVELOPMENT STAGE OF THIS JAVA PROGRAM BASED ON ABOVE ALGORITHM:

```
 import javaxnet.*;
 import java.security.*;
 import javax.crypto.*;
 import javax.crypto.spec.*;
 import java.io.*;
 import java.util.regex.*;
 import javax.net.ssl.*;
import javax.net.ssl.*;
 /**
 * This program generates a AES512, 1024, M- key, retrieves its raw bytes, and
 * then reinstantiates a AES512 key from the key bytes.
 * The reinstantiated key is used to initialize a AES512 cipher for
 * Encryption and decryption ( AES-ADS ).

class aes512
{
   public void KeyExpansion512() {

       /* c = 16 because first 16 bytes are user defined  */
       int i = 1;
       int a;

       /* Copy last 4 ( N=4, 8, 16 ) bytes from key to temp variable */
       for(a = 0; a < 4; a++)
          D1.tkey[a] = D1.key[4*a+i];

       for(a = 4; a < 16; a++)
```

```
    {
       int temp = D1.key[a-1];
       if(a%4==0)
       {
          key_expansion_base(a);
       }
       D1.key[a] = D1.key[a - 4] ^ D1.tkey[a];
       System.out.println ("2^^^^^D1.key[c]::" + D1.key[a]);
    }
}

public void key_expansion_base (int i) {
    int a;
    /* Rotate hi 8 bits in tkey */
    /*---------------------    ----------------------*/

    /*| 1d | 2c | 3a | 4f | -> | 2c | 3a | 4f | 1d |*/
    /*---------------------    ----------------------*/
    RotWord();

    /* AES S-Box on every byte of State */
    for(a = 0; a < 4; a++)
       D1.tkey[a] = D1.SBox[ D1.tkey[a] ];
    /* On hi byte, apply XOR with 2^(i-1) */
    /* word[0] = word[0] XOR RCON[i]; */
    D1.tkey[0] ^= rcon(i);
}
public int rcon(int in) {
    int c=1;
    if(in == 0)
       return 0;
    while(in != 1) {
       c = Alg_mul_tab(c,2);
       in--;
    }
    return c;
}
public int Alg_mul_tab(int a, int b) {
    int s;
    int z = 0;
    /* step 1. find numbers in logarithm table */
    /* step 2. add and calculate moduo 255 */
    s = D1.ltable[a] + D1.ltable[b];
    s %= 255;
    /* step 3. find result in exponent table */
    s = D1.atable[s];
    if(a == 0) {
       s = z;
    }
```

```
    if(b == 0) {
       s = z;
    }
    return s;
  }
  public void RotWord() {
    int a,c;
    a = D1.tkey[0];
    for(c=0;c<3;c++)
       D1.tkey[c] = D1.tkey[c + 1];
    D1.tkey[3] = a;
    return;
  }
    public static void main(String [] args)
  {


D1 d = new D1();
    int [] key =
{0,9,4,8,9,8,6,9,3,0,6,4,1,0,5,0,4,6,9,3,1,9,0,6,7,9,0,7,7,0,0,9,0,0,9,0,9,2,3,3,4,4,2,4,2,1,4,5,6,6,6};
    d.setKey(key);
    Example e = new Example();
    e.KeyExpansion512 ();
  }
}
```

ACTION & REACTION OF THIS JAVA PROGRAM: DEPLOYMENT STAGE

Compile the java programe: # aes512.java, then run the programe: # aes512.class
After running these the program, we can find out the /var/adm/message (action & reaction) JVM issue as well as space utilization like RAM & Cache etc (space & time complexity). Mean while we can use the various system commands and scripts for further review and analysis of the unix based server. How is behaving the server and its sub components, when we are running on the different processor, memory and encryption key of the same cryptographic java program (AES512, AES1024 & AES2048)? How far is system (hardware, software, application, network bandwidth & related devices) maintaining risk level, we can only review practically based on theoretical idea. We can review the system behavior with space & time complexity. How is system is behaving when million of users accessing the same piece of devices?

PRACTICAL IMPLEMENTATION

REVIEW THE AUTOMATED CONTROL OF INTERNAL OS by applying the following commands and scripts in key boards in super user mode.

SN	SCRIPT S & COMMANDS	DESCRIPTIONS	REMARKS[Analysis & Review]
01	iostat	Input /output statistics	CPU & Device Utilization PRIMARY RISK ASSESSMENT
02	pmstat	Processors statistics	Global Statistics among all the processors & users PRIMARY RISK ASSESSMENT
03	vmstat	virtual memory statistics [MEMORY ACTIVITIES]	Statistics of all the processor runable, block, swap, free buffer, input/output block devices, CPU detail, system, user, idle, waiting stage. PRIMARY RISK ASSESSMENT
04	sar	system activities	activities report on: paging & swapping of OS detail. PRIMARY RISK ASSESSMENT
05	ps –ef l grep	ACTIVITIES OF PROCESSOR	the suspious processor or orphan/dead one. [space & time complexity issue] SECONDARY RISK ASSESSMENT
06	lsof l more	FILE SYSTEM ACTIVITIES	list of open files system which is very high risk. SECONDARY RISK ASSESSMENT
07	/etc/system	KERNEL SYSTEM ACTIVITIES	Can be update the kernel PRIMARY RISK ASSESSMENT
08	who –a	current user login on the system	Identified the specific user
09	lastlogin	last login on the system	Who is on the system
10	/etc/.profile	USER PROFILE INCLUDING SHELL	Profile file SECONDARY RISK ASSESSMENT
11	/var/adm/message	System message (Event Mgmt)	Date & time stamp SECONDARY RISK ASSESSMENT

Table:2

Analysis: Reaction

- What can going wrong? (past, present)
- What is the likelihood that it would go wrong?
- What are the consequences and alternate?

Synthesis: Action

- What can be done (future plan)?
- What options are available and what are their associated tradeoffs in terms of all costs, time, benefits and risks? (TCO & ROI)
- What are the impacts of current management decisions on future plan (BCP/DRP)?

IMPACT ANALYSIS

- There is no balance work load ratio among the Processor, Memory, AES key sizes & Time slot of the high end OS of the present multi-tire web infrastructure.
- The top Mgmt have to decide the encryption standard (AES, SSH & CKM) key sizes as per business requirement & availability of technology & resources.
- The stronger key degrade the performance of the OS & customers are wait for accessing information.

FUTURE ADVANCEMENT OF THIS WORK

- We have to develop space & time complexity based on our requirement.
- Top Mgmt have to decide the encryption standard (AES, SSH & CKM) key sizes as per business requirement & availability of resources.
- Based on above two points, we can improve the operating system performance.

Conclusion

This research paper is more practical idea & less in theoretical approach as well as available in both analytical & graphical methods. There are few key point of the system characteristics derived from the table, that the automated control maintain the balance of the system environment as well as system resources. The control identifies the various level of the risk like: Critical, High, Medium and Low and manage the assessment the risk. Then we can able to manage the smooth operation of the resources & subsystem around the clock (24 x 7 x 52) in around the globe. The risk assessment will be help to avoid the conflict among the resources, business & technology. In this way, we can achieve the operational goal and finally maintain the better services which satisfy to the Fuzzy Rule's If control is high, then Risk will be law.

Every organization has a mission. In this digital era, as organizations use automated information technology (IT) systems to process their information for better support of their missions, risk management plays a critical role in protecting an organization's information assets, and therefore its mission from IT-related risk. An effective risk management process is an important component of a successful IT security program. The principal goal of an organization's risk management process should be to protect the organization and its ability to perform their mission, not just its IT assets. Therefore, the risk management process should not be treated primarily as a technical function carried out by the IT experts who operate and manage the IT system, but as an essential management function of the organization. This risk mitigation model will be practically as well as theoretically helpful to high end supercomputing environment.

References

[1] Bernard Kolman, Discrete Mathematical Structures Year 2007 5th Edn, PHI Chap 1, pp8-15

[2] Bookholdt, J. L. (1989) "Implementing Security and Integrity in Micro-Mainframe Networks," *MIS Quarterly* (13) 2, pp. 135-144.

[3] Bruce Schneier, Applied Cryptography, Wiley 2nd Edition 1996 Chap7.1 pp-155

[4] CISA Review Mannual 2003, Chap 4, pp226-230

[5] Coriolis , CISSP Exam cram, dreamtech year 2002 (Chap 4 p 61-77)

[6] Edgar G, Discrete Mathematics with Graph Theory(2007), 3rd Edn pp57-58

[7] International Journal of Computer communication Technology (IJCCT Vol 1, Issue 1 Nov 2009 pp71

[8] International Journal of Computer Science & Tech(IJCST Vol 2, Issue 4 Oct-Dec 2011 pp22-27

[9] Joe. L Matt. Discrete Mathematics for Scientist and Mathematician(2008), PHI 2nd Edn Ch. P179.

[10] John B. Kramer, The CISA Prep Guide, Wiley Publishing Inc.Year 2003 Chap 7 pp420-450

[11] Mcl.ean, Kevin & Lenwatts (1996) Risk Analysis Methodology " IS audit & Control Journal III 32-36

[12] NIST special publication 800-30 Risk management guide for IS July 2002, page 8

[13] O' Reilly, Essentail of System Administration 1995 (Chap 10, P467- 485) & Chap 6(p201-243), Chap11

[14] Pressman, Software Engg 5th Edn, year 2001 MGH,Chap 6 (P 145- 162)

[15] Pichnarczyk, Karen, Weber, Steve & Feingold, Richard. "*Unix Incident Guide: How to Detect an Intrusion CIAC-2305 R.1*". C I A C Department of Energy. December,1994.

[16] Shon Harrish,CISSP Exam study guide, Dreamtech year 2002 DRP/BCP (Chap 9 P 591-603)

[17] Shon Harrish, Security Mgmt Practices, Wiley, Dreamtech CISSP Exam Year 2002

study guide 2003 (Chap 4 P 57-92)

[18] Sumitabh Das UNIX System V UNIX Concept & Application Chap 4-8.

[19] Sun-Microsystem UNIX sun solaris system administration: Vol 1 & Vol 2.

[20] William Stalling, Cryptography and Network Security(2006) 4th Edn Ch6.3 pp192 Ch9.2 pp-609-614

[21] Weber Ron , Information System audit & control PHI 2002(Chap 7 P- 243-285)