 ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)

Dynamic Semi-Group SFK Pattern on Complex Operating System for Optimizing the Risk

Padma Lochan Pradhan
Dept. of CSE, Central Institute of Technology, New Raipur, India

Abstract
Now-a-days,increasing the importance of business and resources over a complex RTS and growing the external risks is a very common phenomenon. The system risks put forward to the senior management focus on complex risk on RTS. The senior management has to decide whether to accept expected losses or to implement into security mechanisms in order to minimize the down time of risk on complex infrastructure. This paper contributes to the development of an optimization model that aims to determine the optimal cost to be invested into UFS mechanisms that,the allocation& distribution of measure components on operating system and relevant resources (i.e. Shell, File and Kernel). Our SFK patternshould be design in such way, the file systems, shell and kernel automatically protected, detected & corrected all the time. Wehave to reduce the system risk by implementing SFK pattern based on semi-group structure, mean while improving the highest access control on the File, Memory and Processor & Kernel system. Finally, we have to maximize the performance, reliability, fault tolerance& minimize the cost, time of the RTOS over a complex web application. Our objective is that fix up the risk at optimal level with minimal cost and time.

Keyword
Shell File & Kernel;Unix File system; Preventive Detective Corrective Control; Cryptographic Key Management; Advanced Encryption Standard;Total Cost Ownership; Risk Mitigation.

I. Introduction

The complex operating system is a collection of hardware, software & application that manages system resources and provides common services for resources, program, application & users. The real time operating system is an essential component of the system software (shell, file & kernel) in computer system. The high level languages (application programs) usually require an operating system function and accountability.
The time-sharing operating systems schedule & reschedule tasks for efficient use of the internal utilities that may also include auditing system software for resource & cost allocation of processor and memory time, mass storage, printing and other resources.
The real time operating system is a multitasking, time sharing & distributed operating system that aims at executing real-time applications. The real-time operating systems often use specialized scheduling algorithms so that they can achieve a deterministic nature of behavior. The main objective of real-time operating systems is their quick and predictable response to events. They have an event-driven or time-sharing design and often aspects of both. An event-driven system switches between tasks based on their priorities or external (resources) events while time-sharing operating systems switch tasks based on clock interrupts.

There are various kinds of systemcontrol available and implemented on operating system to protect our IT assets for external & internal hackers. The PDC model & Mechanism traditionally prevent the core components of RTOS. The processor & memory is the core component of any type operating system. The processor and kernel is fully functional dependency on each other, but file and shell is the communication components of the RTS. We can improve the performance of RTOS by updating the kernel time to time. Kernel is the Nucleus of the operating system.

Architecture of the Operating System:

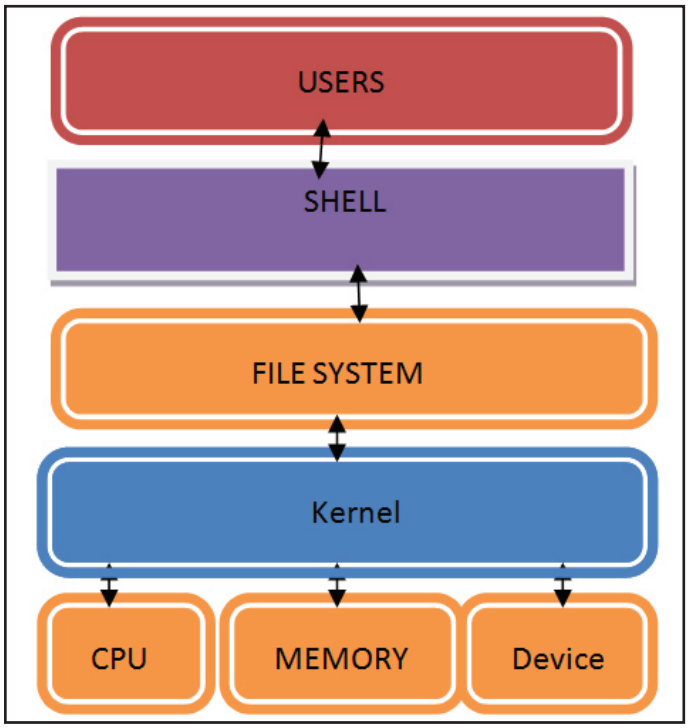


Fig. 1: Architecture of the RTOS

The real time operating system is consists of five measure devices and components such as, file, shell, kernel, memory & processor. The processor & memory capability help to forecasting of business & resources of the core component of any type operating system and the AES & SSH is fully dependable on P & M of the instruction level parallelism (SISD, SIMD, MISD, MIMD). The processor, memory and kernel is fully functional dependency on each other , but file and shell is the communication components between users and the OS. The processor, memory & encryption is preventing to data & services all the time.

We have to prevent, detect & correct the operating system resources all the time (Data & Services).
This PME mechanism protecting and providing high level data & services on any type of organization in around the clock (7 x 24 x 52). The above PME model maximizes the utilities of processor, memory & high utilization of encryption key at optimal cost. The processor, memory & encryption key always prevention, detection & correction at minimal cost with high availability of data & services as per business & resource requirement. Therefore, the stronger security on PME always depends on the PDC or verse versa.

Fig. 2: PME Model

This SFK pattern protecting& providing high level data & services on any type of organization in around the clock (7 x 24 x 52). The above SFK model maximizes the utilities of Shell, File & Kernel & maximizes the productivity & throughput at optimal cost. These shell, file & kernel always prevention, detection & correction at minimal cost with high availability of data & services as per business & resource requirement. Therefore, the stronger security on SFK always depends on the PME and PDC or verse versa.

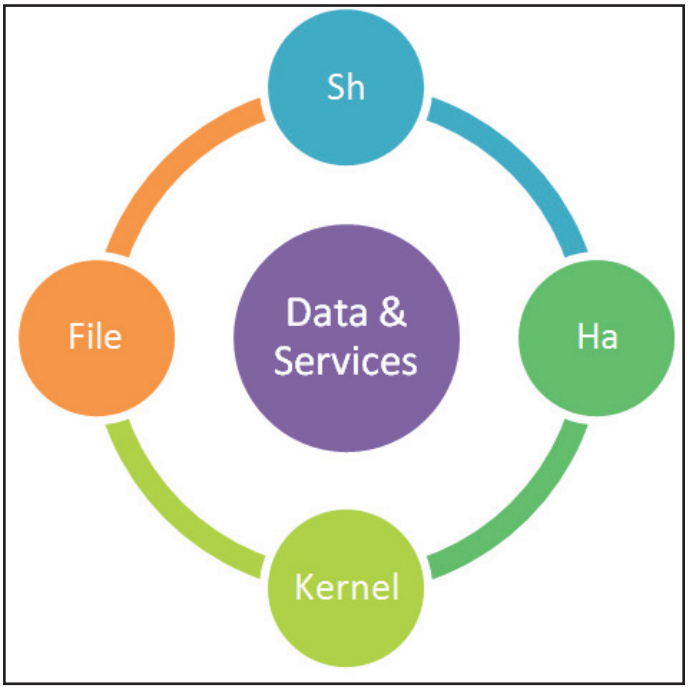


Fig. 3: SFK Model

The real time operating system control is a step by step process of securely configuring a system to protect it against unauthorized access, mean while taking steps to make the system more reliable. Generally anything that is done in the name of system. The preventive control ensures the system is secure, reliable and high available for high IT culture. The operating system control is the process to address security weaknesses in operation systems by implementing the latest OS patches, tools, hot fixes and updates by applying the 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 the external intrusion. The 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.

The preventive, detective, corrective control and high availability (Ha) preventing the data & services around the clock. The data & services are the measure components of the shell, file, memory, processor & kernel of the operating system. Therefore, the preventive control is very much essential for betterment of multi-tier IT infrastructure.

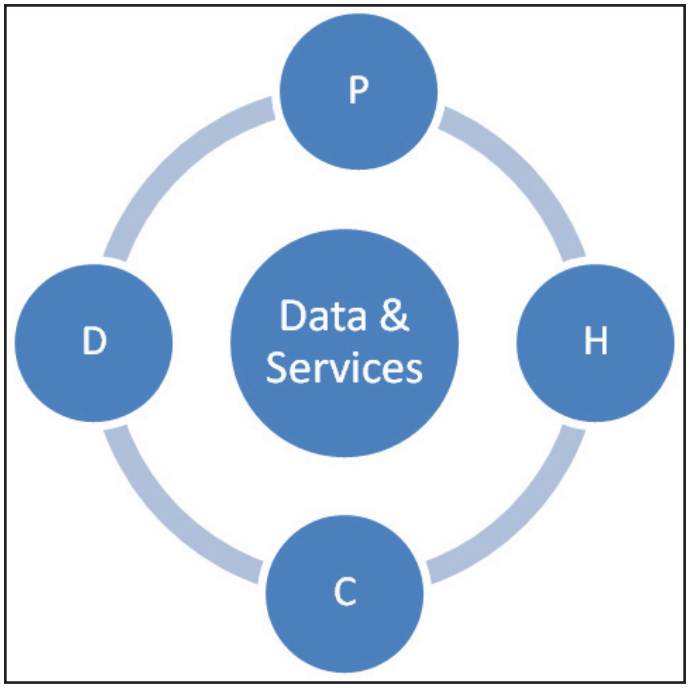


Fig. 4: PDC Model

Data Collection Based on Existing Control: (Basic Data) [13,18-19]

There are number of system control methods developed as per requirement of the secure computing to achieve the highest level of business objective. There are few methods developed based on UNIX server and system programming. Preventive control is inversely proportional to the Risk.

Problem in Existing Control

- There is no protection, detection & automatic correction on the Shell, File & Kernel. There is no balance ratio among the Kernel, Processor, Memory, File System [Encryption Key] &time slot of the high end OS. The high level decision process is required to implement resources like kernel, processor and instruction level parallelism (SISD, SIMD, MISD, MIMD) & high memory& encryption key sizes for high end business. The high end technology would be match with high volume business.

 ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)

Table 1: Data Collection on RTS

SN	SYSTEM FILES	ACTION PLAN	REMARKS
1	/etc/system	Can be update the kernel & n-bit processor	**Can be improve the system performance KERNEL**
2	/etc/hosts	Develop the scripts: allow/disallow as per policy, chimed 000= /etc/nnn-mark disallow	**preventative control [H, M, L] Can be improve the system security**
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 [H, M, L] Can be improve the system security**
4	/use/bin/rash, etc/pam.conf	Disable all remote services: chmod 000 /usr/bin/rsh, rksh, rcp, ipcs, ruser, rlogin, uptime.	**preventative control SHELL**
5	/vary/dam/message	Date & time stamp (DC: event mgmt)	**Internal audit purpose Detective control**
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 [H, M, L]**
7	/etc/init.conf	OS services, run level	**preventative control**
8	**etc/ssh/sshd_config CKM file system Automated Control**	**Cryptography enable through ssh implementation AES: 256 bits chipper chipper 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 chimed r w x (i. e. 4 2 1) – blank is nothing [H, M, L]**

Proposed Dynamic SFK PAT For Risk Optimization

This research paper contributes to the development of an optimization model that objective to determine the optimal cost, quality & time to be implemented into security mechanisms deciding on the measure component of RTS resources (i.e. SHELL, FILE & KERNEL). That's why we are calling as it SFK model for system based risk analysis. Furthermore, the model & mechanism optimize the cost, time & resources is supposed to reduce the system risks. We have to optimize the technology & resource cost and maximizes the business (throughput).

We have to implement our idea based on the isomorphism graph theory, how the operating system optimizing as per our business requirement. Our objective is that Maximize our Business (throughput) & minimizes our Technology & Resources.

The Validation & Varification of RTS Based on Proposed Prevention Matrix:

The complex RTS validation & verification for the high performance computing to manage E-Commerce, E-Payment and product like B2B, B2C, P2P & G2G. These system validation, verification & benchmarking can be define in the Prevention Matrix. We have to maintain the risk free environments on the hardware, software & application level on basis of the following data.

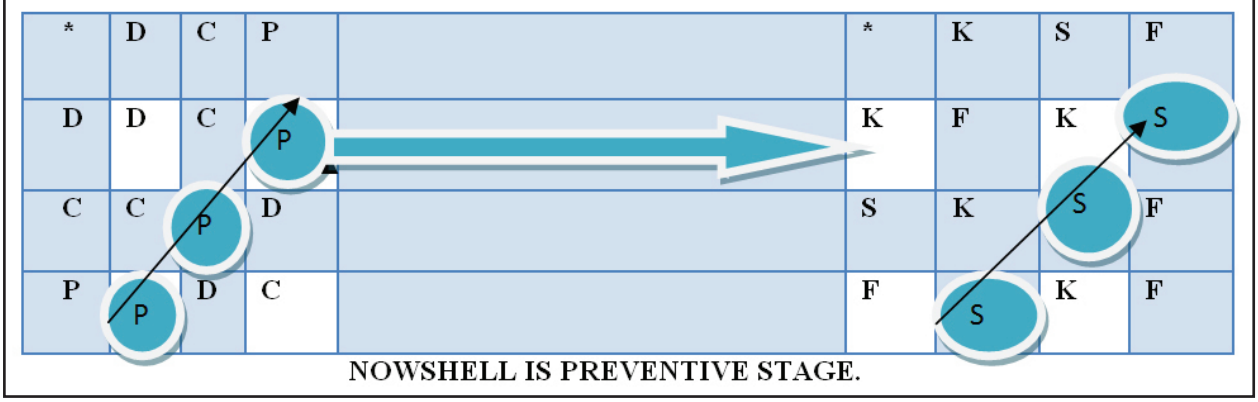
Table 2: Prevention Matrix (Resource Allocation of RTS)

(Derived Data):					Encryption Control Matrix				
E	**128**	**256**	**512**	**1024**	**2048**	**A=2^n**	AES	**MIMD**	**Ha**
S	512	1024	2048	4096	8192	S=2^n	**SSH**	MIMD	Ha
P	32	64	128	256	512	P=2^n	**Processor**	MIMD	Ha
M	16	32	64	128	256	M=2^n	**Memory(GB)**	MIMD	Ha
Sh	X	X	X	X	X	X	**Shell**	MIMD	Ha
F	X	X	X	X	X	X	**File**	MIMD	Ha
K	X	X	X	X	X	X	**Kernel**	MIMD	Ha
C	L	L	M	H	H	X	**Control**	MIMD	Ha

(L- LOW RISK, M-Medium RISK, H-HIGH RISK)
(PC+DC+CC=K) [CKM=k.1/R] Fuzz's Law]
Where: Sh: Shell of the OS, K: Kernel of the RTOS, F:File System of OS, P: Processor, M: Memory , E: Encryption Key, C: Control, ***X: UNKNOWN.*** We have to assume the X values as per business requirement (BCP) and availability of resources & technology.
Let us consider R={P} & M={S, F, M, E, P, K, Ha). It is easy to verify that the following operation tables give Semi Group Structures for R & M respectively.
As per operating system the external layer (Sh) Shell have to prevent from external users. Like, wise (F) file system have to prevented as respectively P,M, E & K. as follows:
The processor is directly proportional to the Memory; Advance Encryption Standard(AES)(is directly proportional to the Memory. The system hardening is directly proportional system control, meanwhile control is proportional to the Mitigation. The set of parameter [{P, M, E, K, Ha} ∈ RM}] , Where HA is the high availability. We can optimize the risk factor by help of these five elements. All these five elements depend on eachother. The availabilities is the main concern among the all of them.As per efficient business & resource management the CKM-AES very well suited for restricted space management(Time –Space complexity) where both encryption and decryption is implemented as per table [2] data.Relation is imply as:
(R ∈ P, M, E & Ha) and (R ∈ C, I, A & Ha)
(R ∈ P, M, E & Ha) and (R ∈ P,D, C & Ha)
(R ∈ S, F, K & Ha) and (R ∈ P, D, C & Ha)
(R ∈ S, F, K & Ha) and (R ∈ C, I, A & Ha)

Let us consider R={P,D,C} & M={S, F, K). It is easy to verify that the following operation tables givesSemi Group Structures for R & M respectively.

Model:1.a PREVENTION MATRIX ON SHELL DCP Verses KSF



NOWSHELL IS PREVENTIVE STAGE.

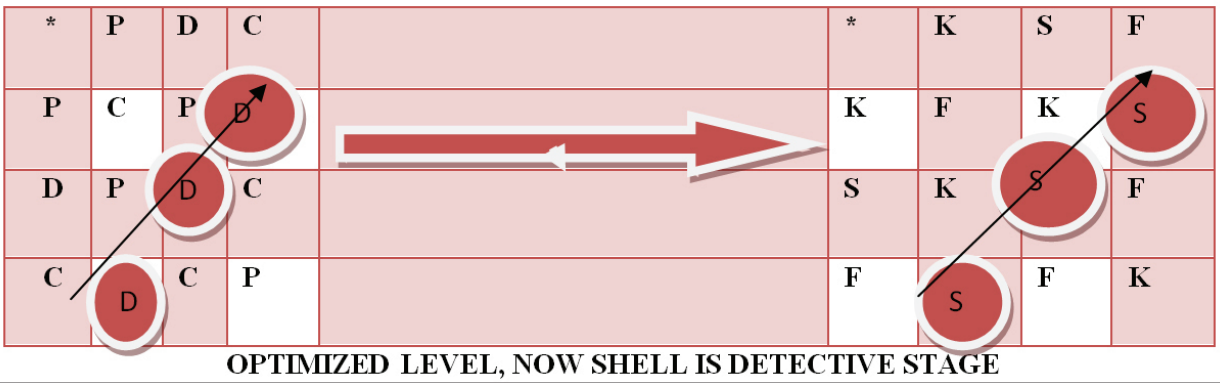
REVENTION IS BETTER THAN CURE.

Diagonal view (P-P-P=S-S-S)=> Pattern

Let us consider that: f(D)=S, f(C)=K, f(P)=F

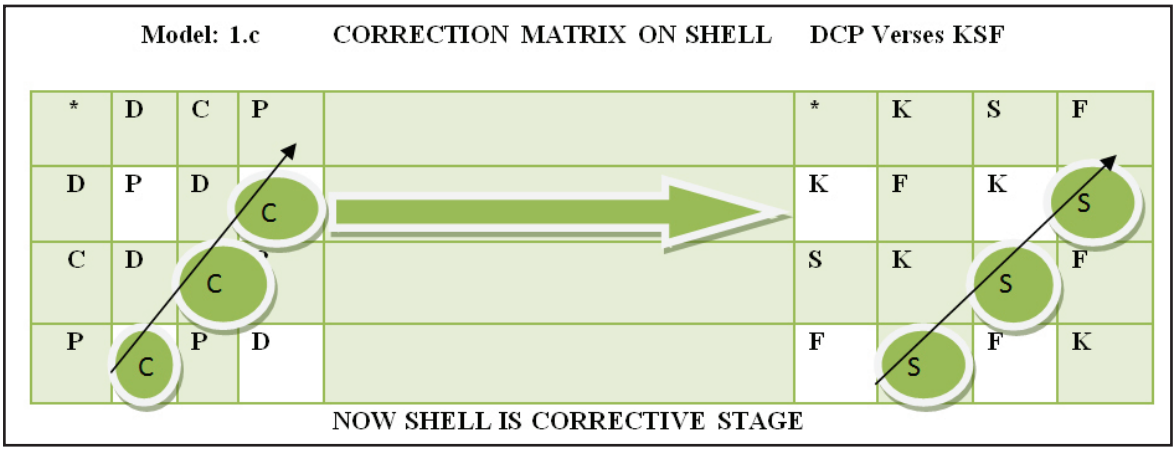
Now,replacing the function in R by their images and rearranging the tables, we obtain exactly the table for M.Thus R & M are Isomorphic. Therefore, this model is called " SEMI-GROUP as well as ISOMORPHIC MODEL ON OPERATING SYSTEM FOR RISK OPTIMIZATION". Then we can move forwards to the NEXT OPTIMIZED LEVEL as follows.

Model:1.b DETECTION MATRIX ON SHELL PDC Verses KSF

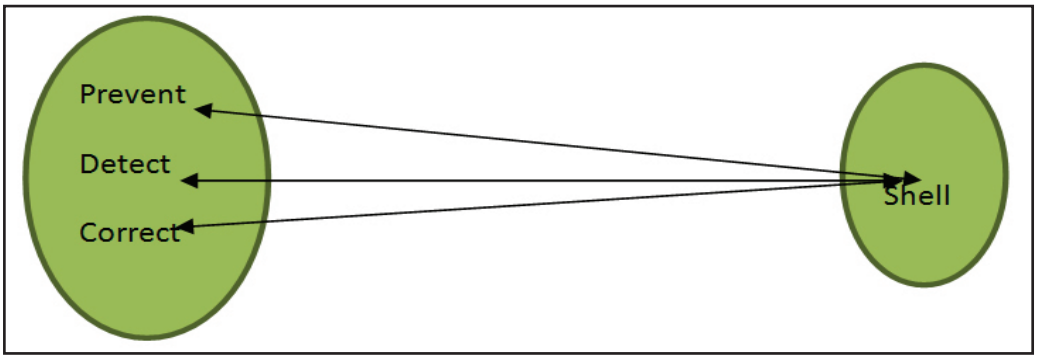


OPTIMIZED LEVEL, NOW SHELL IS DETECTIVE STAGE

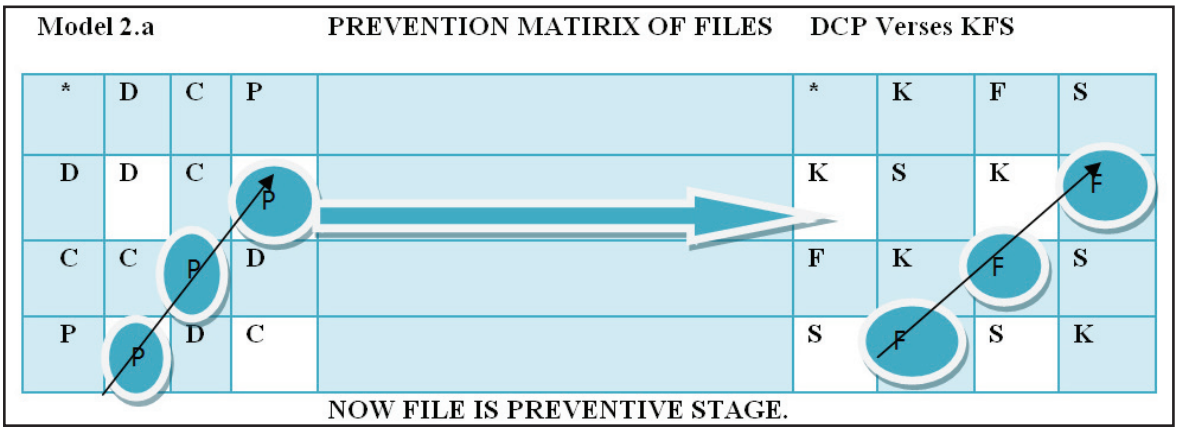
Diagonal view (D-D-D=S-S-S) => Pattern



Diagonal view (C-C-C=S-S-S) => Pattern

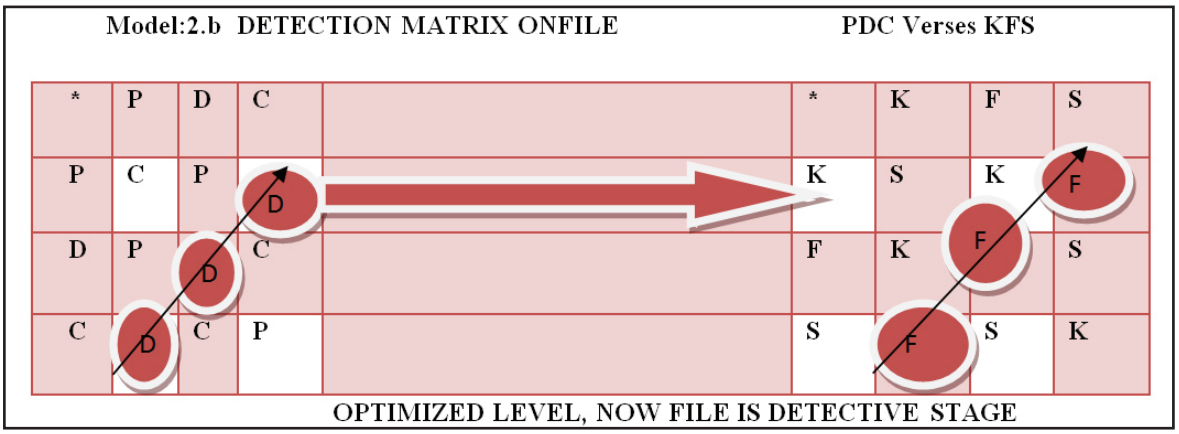


This graphical representation show that, shell is prevented, detected and corrected.

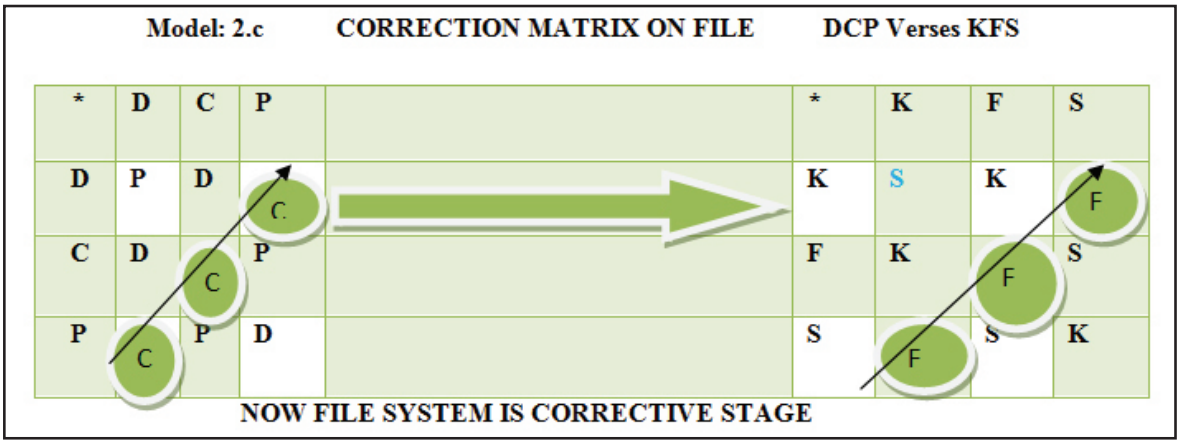


PREVENTION IS BETTER THAN CURE.

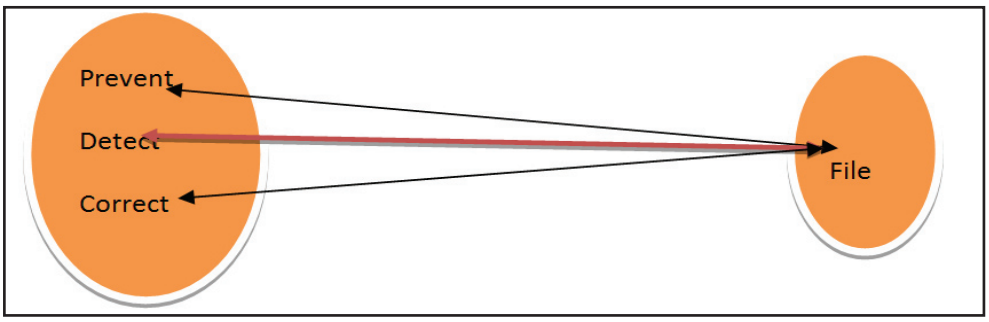
Diagonal view (P-P-P=F-F-F) => Pattern



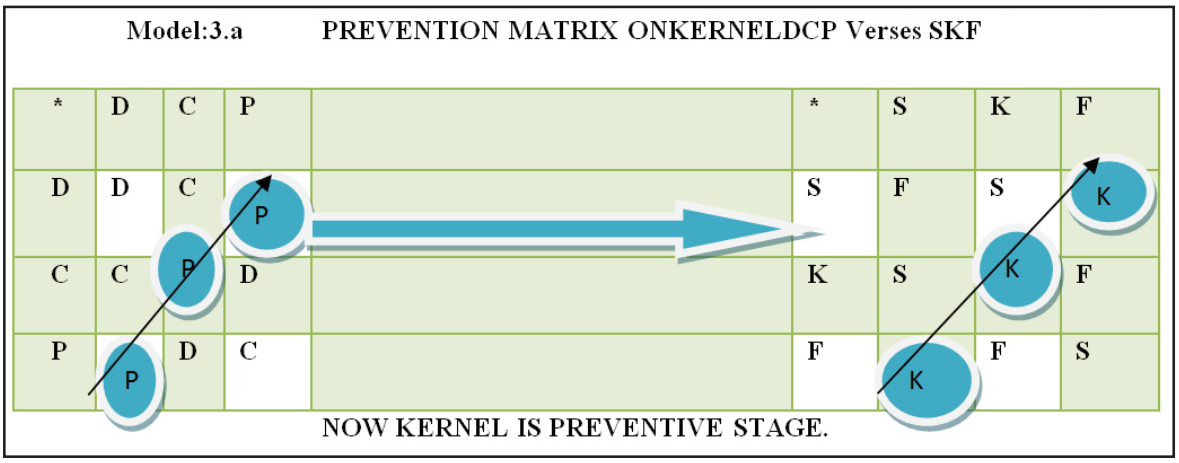
Diagonal view (D-D-D=F-F-F) => Pattern



Diagonal view (C-C-C=F-F-F)=> Pattern

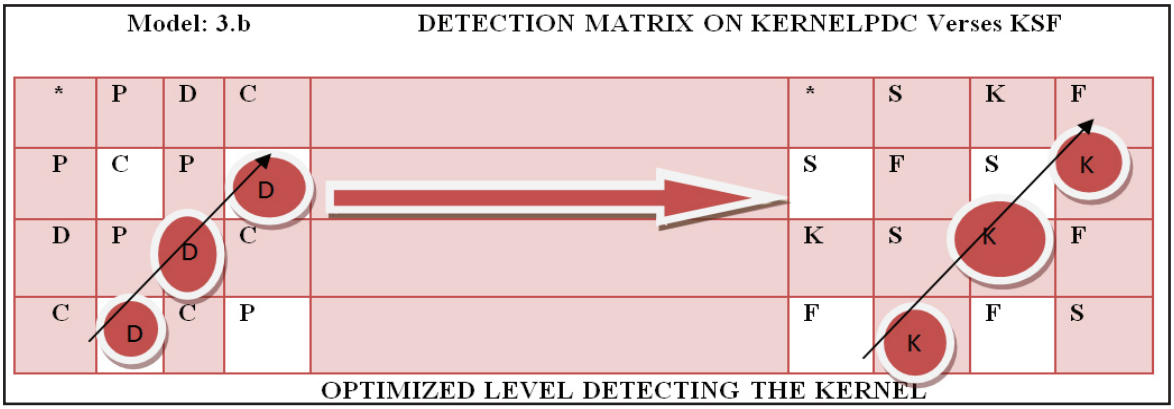


This graphical representation show that, File system is prevented, detected and corrected.

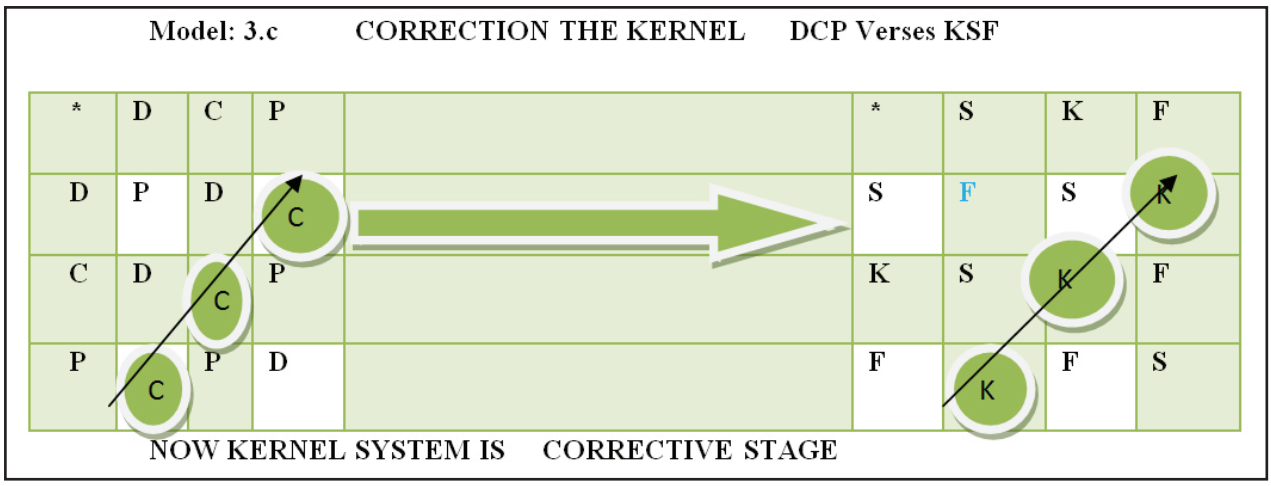


PREVENTION IS BETTER THAN CURE.

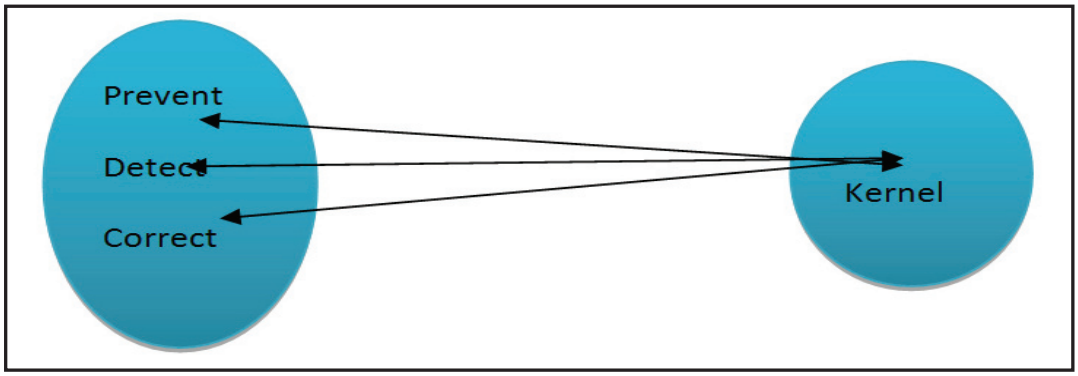
Diagonal view (P-P-P=K-K-K)) => Pattern



Diagonal view (D-D-D=K-K-K)=> Pattern



Diagonal view (C-C-C=K-K-K)) => Pattern



This graphical representation shows that, kernel is prevented, detected and corrected.This is the dynamic life cycle of PDC&SFK based on the semi-group, isomorphic & directed graph theory. When SFK optimization technique applied on OS, the space & time complexity of Processor, Memory and users details can be detected by OS system parameters is already defined in existing risk assessment method on file system (/var/adm/messages), then we can fix up thefile system of encryption (AES) Key as per availability of technology,resources and business requirement. In this way we can dynamically optimize our technology & business risk.

We can conclude that from the above optimization model the, shell, file, processor,memory &encryptionkey have to take highest priorities of the Preventive, Detective and Corrective, action plan, which is shown in the model 1, 2& 3 respectively. In this way, we can improve the performance & security of the high end Processor, Memory on instruction pipe lines like :SISD, SIMD, MISD, MIMD. But, the encryption control will be facilitate and resolve the various issue when it spans several jobs and applications are running simultaneously under heterogeneous complex based web infrastructure around the web world. These above optimization models will be very helpful for Instruction level parallelism for high end computing. We hope this theoretical and experimental idea will be very much help to the parallel computing environment to optimize the operating system software risk. We can improve our risk optimization model that,which will be help to the risk management on operating system.

Table 3: Life Cycle & Brief Summary of PDC & SFK Pattern

MODEL	STAGE	DESCRIPTION	ACTION PLAN
Model-1.a	1st Round: Shell	Pattern view (P-P-P-S-S-S)	Preventive Control on Shell
Model-1.b		Patternview (D-D-D-S-S-S)	Detective Control on Shell
Model-1.c		Patternview (C-C-C-S-S-S)	Corrective control on Shell
Model-2.a	2nd Round: File	Pattern view (P-P-P-F-F-F)	Preventive Control on File
Model-2.b		Pattern view (D-D-D-F-F-F)	Detective Control on File
Model-2.c		Pattern view (C-C-C-F-F-F)	Corrective control on File
Model-3.a	3rd Round: Kernel	Pattern view (P-P-P-K-K-K)	Preventive Control on Kernel
Model-3.b		Patternview(D-D-D-K-K-K)	Detective Control on Kernel
Model-3.c		Patternview(C-C-C-K-K-K)	CorrectiveControl on Kernel
Model-1.c	Automated	Pattern view (C-C-C-S-S-S)	Corrective control on Shell
Model-2.c	Corrective Control	Patternview (C-C-C-F-F-F)	Corrective control on File
Model-3.c		Pattern view(C-C-C-K-K-K)	Corrective control on File

6.1 Practical Implact Analysis on SFK Model:

We have to verify & validated the RTS integrity, high availability, reliability, scalability, Authentication, Authorization & Confidentiality of the main components of real time operating system like: Shell, File, Kernel, Processor, Memory & Encryption as per business requirement & availability of technology. We can apply some review method on internal UNIX operating system of super user mode [5-6, 9-10].

Table 4: (Verification of RTOS)

SN	SCRIPT S & COMMANDS (INPUT)	DESCRIPTIONS ACTION PLAN	Risk Analysis (OUTPUT)
01	iostat	Input /output statistics	CPU & Device Utilization, HA availability, Reliability & integrity of Processor. 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. HA availability, Reliability & integrity of Memory. PRIMARY RISK ASSESSMENT
04	sar	system activities	Activities report on: paging & swapping of OS detail. PRIMARY RISK ASSESSMENT
05	ps –ef | grep	ACTIVITIES OF PROCESSOR	The suspious processor or orphan/dead one. [space & time complexity issue] SECONDARY RISK ASSESSMENT
06	lsof | 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, Accountability & Authentication
10	/etc/.profile	USER PROFILE INCLUDING SHELL	Profile file SECONDARY RISK ASSESSMENT
11	/var/adm/message	System mesg (event mgmt) DC	Date & time stamp SECONDARY RISK ASSESSMENT
12	/use/bin/rash, etc/pam.conf	Disable all remote services: chmod 000 /usr/bin/rsh, rksh,rcp, ruser,rlogin, uptime.	Preventative control
13	/var/adm/syslog	syslog system logs	Detective control, Accountability & Authentication
14	/var/adm/sulog	super user log	Detective control, Accountability & Authentication
15	/var/adm/loginlog	user login log	Detective control, Accountability & Authentication
16	etc/ssh/sshd_config	AES, CKM Key mgmt	Run the scripts: Preventive control

Acheivement & Results:

- Maximize the protection, detection &correction(PDC) at optimal cost and time.
- Maximize the performance, reliability, high availability at optimal cost(TQM).
- Maximum utilization of resources (PME) at minimal cost at right time in right way.

Table 5: Risk Optimization Matrix on RTS

SN	OUTPUT / INPUT	PRODUCTIVITY	THROUGHPUT	SERVICES	COST	TIME	(ROI +TCO+TQM)
1	PDC	MAXIMUM	MAXIMUM	MAXIMUM	MINIMAL	OPTIMAL	MAXIMUM
2	PME	MAXIMUM	MAXIMUM	MAXIMUM	MINIMAL	OPTIMAL	MAXIMUM
3	SFK	MAXIMUM	MAXIMUM	MAXIMUM	MINIMAL	OPTIMAL	MAXIMUM
4	Business	MAXIMUM	MAXIMUM	MAXIMUM	MINIMAL	OPTIMAL	MAXIMUM
5	Resources	MAXIMUM	MAXIMUM	MAXIMUM	MINIMAL	OPTIMAL	MAXIMUM
6	∑(RM)	MAXIMUM	MAXIMUM	MAXIMUM	MINIMAL	OPTIMAL	GT=∑(RM)

Conclusion

This dynamic semi-group SFK patternproviding maximum audit for individuals who are accessing sensitive information on shell, file system & kernel. The audit functions is accomplished through PDC control mechanisms that require identification, authentication, authorization, accountability, non-repudiation, availability, reliability & integrity through the preventive control on various file system.[/var/adm/message, /var/adm/loginlog, /var/adm/sulog]. (Detective Control). This SFK pattern for preventive, detective and corrective action plan for the safest ways to save the resources for all the time and every time. The security programmer & administrator now has a base knowledge of security, server hardening, intrusion detection, auditing and security tools. This knowledge can be directly applied to their servers and many vulnerable holes will now be filled. Therefore, it is critical that every Unix security-minded programmer maintains their knowledge of security by researching and referring to the internet resources that have been mentioned in survey data collection.

References

[1] Bernard, Kolman,"Discrete Mathematical Structures", New Delhi, India: Person Education India. (PHI), 2007.

[2] BruceSchneier,"Applied Cryptography", New Delhi, India: Wiley 6 Chap, 1996.

[3] Edgar, G.,"Discrete Mathematics with Graph Theory", New Delhi, India: Person India. (PHI), 2007.

[4] Joe. L Matt.,"Discrete Mathematics for Scientist and Mathematician", New Delhi, India: Person Education India. (PHI), 2008.

[5] Hwang, Kai,"Advance Computer Architecture. New Delhi, India: Tata McGraw Hill, 2008.

[6] O' Reilly,"Essential of System Administration", O' Reilly Media: USA, 1995.

[7] Shon, Harrish,"CISSP Exam study guide", New Delhi, India: Dreamtech, 2002.

[8] Shon, Harrish,"Security Mgmt Practices", New Delhi, India : Wiley Publishing Inc., 2002.

[9] Sumitabh, Das,"UNIX System V UNIX Concept & Application", Delhi, India: Tata McGraw Hill, 2009.

[10] Sun-Microsystem,"UNIX Sun Solaris system administration", 2002.

[11] William Stalling,"Cryptography and Network Security, 2006.

[12] Weber, Ron,"Information System Control & Audit", New Delhi, India: Person Education India. (PHI), 2002.

Padma Lochan Pradhan received M.Sc. (Physics with Electronics) from Sambalpur University in 1983 and M Tech in Computer Science in 2012 from Berhampuer University, India. He is interested in system security, cryptography, Real time operating system, system programming & risk mgmt. In around 20 years in IT industries and 10 years in academic & research in various capacity in IBM, Sun Micro system, Thomson scientific (ISI) in USA as well as Indian Telephone Industries etc. Now, working as an Associate Professor (Computer Science Dept) in Central Institute Raipur, CG, India. Apart from this, he is a PG DBA and certified UNIX SUN Solaris expert.