



March 2022

Proposed L-Shape Pattern on UFS ACM For Risk Analysis

Abhishek Asthana

Shri Rawatpura Sarkar University, Raipur, India, asthanaabhishek12@gmail.com

Padma Lochan Pradhan Dr

Shri Rawatpura Sarkar University, Raipur, India, citrprcs@rediffmail.com

Follow this and additional works at: <https://commons.erau.edu/jdfsl>



Part of the [Computer Law Commons](#), and the [Information Security Commons](#)

Recommended Citation

Asthana, Abhishek and Pradhan, Padma Lochan Dr (2022) "Proposed L-Shape Pattern on UFS ACM For Risk Analysis," *Journal of Digital Forensics, Security and Law*. Vol. 17 , Article 7.

DOI: <https://doi.org/10.15394/jdfsl.2022.1728>

Available at: <https://commons.erau.edu/jdfsl/vol17/iss1/7>

This Article is brought to you for free and open access by the Journals at Scholarly Commons. It has been accepted for inclusion in Journal of Digital Forensics, Security and Law by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.



(c)ADFSL



Proposed L-Shape Pattern on UFS ACM For Risk Analysis

Abhishek Asthana¹ and Padma Lochan Pradhan²

¹Dept of Computer Science. Shri Rawatpura Sarkar University, India
asthanaabhishek12@gmail.com

²Dept of Computer Science. Shri Rawatpura Sarkar University, India
citrprcs@rediffmail.com

Abstract

At this cloud age, there is tremendous growth in business, services, resources, and cloud technology. This growth comes with risk of unsafe, unordered, and uncertainty due to unauthorized access and theft of confidential propriety data. Our objective is to model around Read, Write and Execute to resolve these unordered, unsafe, and uncertain issues. We will develop a L-Shape pattern model matching UFS ACM to minimize the accessibilities based on RIGHT & ROLE of the resources and maximize the quality of services for safety and high availability. The preventive, detective, corrective (PDC) services are the major roles for all levels of management to coordinate, control the multiple technologies and resources which are working simultaneously. It will be more ordered, accountable, and actionable on real-time access control mechanism for scalabilities, reliability, performance, and high availability of computational services. We have to make safer, certain, unified, and step-by-step normalization by applying this UFS ACM mechanism on UNIX operating system. This proposed research paper covers a wide range of areas covering optimization, normalization, Fuzzy Low, and Risk assessment.

Keyword: Access Control Mechanism (ACM), Unix File System (UFS), Read Write Execute (RWX), Real Time Operating System (RTOS), Preventive Detective Corrective Control, Risk Mitigation.

1. Introduction

The real-time operating system is a large scale has even greater responsibilities and powers for long term business like web based and mobile computing. It is just like a traffic management system, it makes sure that different programs and software packages the users and clients running at the same time do not interfere with each other. The real-time system is also responsible for risk and security angles to ensure that unauthorized users do not access the objects (Das,2017; Trent,2017; Weber, 2014).

The preventive control ACM provides accountability for individuals who are accessing sensitive information on the application, system software, server, and network. We have to develop the multi-dimensional access control mechanism for risk mitigation on a large-scale UNIX operating system based on available technology, business, and resources for better. We have to prevent our data and service from public resources and unauthorized users over a complex real-time operating system (Andrew, 2018; Weber, 2014). Now a day, increasing the complex business, applications, clients, users, and resources over a heterogeneous business domain of the multiple locations of WAN, LAN, multiple complex IT Infrastructure, computer & communications systems by IT industries has increased the uncertainty, un order, risk of theft to proprietary data & services. The operating system control & audit is a primary method of protecting, detecting & correcting the system resources (Processor, Memory, Kernel & File system) (Das, 2017; Padma,2018).

2. Access Control Mechanism

This access control mechanism is a prerequisite to preventive control. The objectives of an access control system are often described in terms of protecting system resources against inappropriate or undesired user access to the resources (FS, Database). From a business perspective, this objective could just as well be described in terms of the optimal sharing of information. After all, the main objective of IT is to make information available to users and applications (Trent, 2017; Weber, 2014). The greater degree of sharing may get in the way of resource protection; in well-managed and

effective access control system actually facilitates sharing. This is a sufficient fine-grained access control mechanism that can enable selective sharing of information wherein its absence, sharing may be considered too risky altogether over a complex LAN, WAN infrastructure (Das, 2017; Hussian, 2017; Thomas, 2017).

The access control mechanism is the process of mediating each and every request to system resources, applications, and data maintained by the real-time operating system and determining whether the request should be created, approve, granted, or denied as per top management policy. The access control mechanism, management, and decision are enforced by implementing regulations established by a security policy (Andrew, 2018); Das, 2017; Weber, 2014).

Access control is concerned with determining the allowed activities of legitimate users, mediating every attempt by a user to access a resource in the system. The complex information technology (IT) infrastructure can implement access control systems in many places and at different levels. The real-time operating systems use access control to protect files and directories. The database management systems (DBMS) apply access control to regulate access to tables and views. The most commercially available application systems implement access control, often independent of the operating systems programming and DBMSs on which they are installed on the real-time system (Trent, 2017; Yang, 2017; Weber, 2014).

2.1. Attribute Based Access Control (ABAC)

These rights and permissions are implemented differently in systems based on discretionary access control (DAC) and mandatory access control (MAC).

In any access control model, the entities that can perform actions on the system are called subjects, and the entities representing resources to which access may need to be controlled are called objects (see also Access Control Matrix: data collection). Subjects and objects should both be considered as software entities, rather than as human users: any human user can only have an effect on the system via the software entities that they control. The authorization involves the act of defining access rights for subjects. An authorization policy specifies the operations that subjects are allowed to execute within a system. The most modern operating systems implement authorization policies as formal sets of permissions that are variations or extensions of three basic types of access (RWX) (Das, 2017; Trent, 2017; Weber, 2014).

- Read (R): The subject can read, write & execute the UFS
- Read file contents ▪ List directory contents
- Write (W): The subject can change the contents of a file or directory with the following tasks:
- Add ▪ Update ▪ Delete ▪ Rename

Execute (X): If the file is a program, the subject can cause the program to be run.

2.2 Capability of UFS ACM

The access control models, used by current systems, tend to fall into one of two classes: those based on capabilities and those based on access control lists (ACLs). In a capability-based model, holding an unforgettable reference or capabilities to an object provides access to the object how possession of access is conveyed to another party by transmitting such a capability over a secure channel. In an ACL-based model, a subject's access to an object depends on whether its identity is on a list associated with the object; access is conveyed by editing the list. The different ACL systems have a variety of different conventions regarding who or what is responsible for editing the list and how it is edited (Weber, 2014). Both capability-based and ACL-based models have mechanisms to allow access rights to be granted to all members of a group of subjects (often the group is itself modelled as a subject). The access control models are sometimes categorized as either discretionary or non-discretionary. The three most widely recognized models are Discretionary Access Control (DAC), Mandatory Access Control (MAC), and Role-Based Access Control (RBAC). MAC is non-discretionary (Das, 2017; Weber, 2014).

2.3 Unix File System (UFS)

The access control model, used by current system, tend to fall into one of two classes: those based on capabilities and those based on access control lists (ACLs). In a capability-based model, holding an

unforgettable reference or capabilities to an object provides access to the object how possession of access is conveyed to another party by transmitting such a capability over a secure channel. In an ACL-based model, a subject's access to an object depends on whether its identity is on a list associated with the object; access is conveyed by editing the list. The different ACL systems have a variety of different conventions regarding who or what is responsible for editing the list and how it is edited (Weber, 2014). Both capability-based and ACL-based models have mechanisms to allow access rights to be granted to all members of a group of subjects (often the group is itself modelled as a subject). The access control models are sometimes categorized as either discretionary or non-discretionary. The three most widely recognized models are Discretionary Access Control (DAC), Mandatory Access Control (MAC), and Role-Based Access Control (RBAC). MAC is non-discretionary (Das, 2017; Weber, 2014).

2.4. Unix File System Architecture

Every directory and file on the system has an owner, and also an associated group. It also has a set of permission flags that specify separate read, writes, and execute permissions for the user (owner), group, and others (everyone else with an account on the computer). The `ls -l` command detects the permissions and group associated with files when used with the `-l` option. Some of the systems `g` options are also needed to see the group information. Every item in a Unix file system can be defined as belonging to one of three possible types (Das, 2017; Weber, 2014).

Ordinary files (Regular)

An ordinary file can contain text, data, or program information. An ordinary file cannot contain another file or directory. An ordinary file can be thought of as a one-dimensional array of bytes.

```
-rw-rw----
```

```
2 sram staff 12040 Nov 12 2014 admin
```

- Ordinary files can be created by these scripts as follow: `touch admin`, `vi admin`, `pico`, `giv`.

Directories

The directories as containers that can hold files, and other directories and subdirectories. The directory is implemented as a file that has one line for each item contained within the directory. Each line in a directory file contains only the name of the item and a numerical reference to the location of the item. The reference is called an *i*-number and is an index to a table known as the *i*-list. The *i*-list is a complete list of all the storage space available to the file system.

```
drwx----- 2 ram staff 2048 Jan
```

```
2 1997 private, Where d is directory file.
```

The directory files can be created by this command: `mkfs` and `mkdir` as private.

Special files

The special files represent input/output (i/o) devices, like a TTY (terminal), a disk drive, or a printer. Because UNIX treats such devices as files, a degree of compatibility can be achieved between device i/o, and the ordinary file i/o, allowing for the more efficient use of the software. The special files can be either character special files that deal with streams of characters or block special files, which operate on larger blocks of data. Typical block sizes are 512 bytes, 1024 bytes, and 2048 bytes. Similarly, there are symbolic links, pipe and door files are available on UFS to maintain the inter-process communication (Das, 2017; Trent, 2017).

3. Technical Literature Survey and Review

The real-time UNIX operating system, UFS ACM literature review, and survey on System Security and risk management area is a very valuable task to collect the actual data and evidence in the real life. It is one of the ongoing processes continuously. It is very time-consuming to analyze & judge the real-time data. There are many textbooks & reference books that help us to find out the real issue. There are many more references are presented in the References Section to focus and take care reference books like Ron Weber, Das, and others for helping access control mechanism analysis this basic data collection and analysis. The Researcher has to focus on the system-specific UNIX OS for security, reliability, and high availability to our business & resources all the time around the globe. Therefore, the top management has to decide& develop the ACM model on the Development, Deployment & Production level of UNIX Machine. We must concern with finding out the security

mechanism & model for risk analysis based on technology survey and data collection (Das, 2017; Trent, 2017; Weber, 2014).

3.1. Data Collection (DSS Data)

There are a number of preventive access control (PAC) defined, designed & developed as per the requirement of secure computing to achieve the highest level of business objective of the real-time Unix operating system. There are a few data models to be collected & analyzed based on UFS and operating system programming. The Unix file system has to be developed as per business requirements all the time and every time around the globe (Das, 2017; Weber, 2014).

Table 1. UFSS data collection & Analysis

SN	Inode	Subjects	L	USR	GRP	Byte	UFS Dt	Objects(UFS)	Remarks	RISK
1	173456	drw-r--r--	1	e-comm	Usr	1233	Jun 7 10:41	/etc/system	Directory file	L
2	234561	drwx-----	2	e-gov	Usr	1234	July 6 12:23	/etc/host	Directory file	M
3	890123	- rwxrwxrwx	3	e-b2b	Staff		Nov 2 00:10	Test.html	Ordinary file	H
4	876432	drwx-----	6	e-h2h	Usr	512	May 3 12:31	Public	Directory file	H
5	132456	drwxr-xr--	1	sam	Staff	1024	Nov200:10	/etc/ssh/ssh_config	Directory file	H
6	987651	drwxr-xr--	1	ram	Staff	1024	Nov200:10	Myfile	Directory file	H
7	890123	Lrwxrwxrwx	1	ram	Staff	8	May 3 12:31	Zn.dat->gold.dat	Link File	H
8	345123	crw-----	1	root	System	0	Nov200:10	/dev/rnd0a	Character Spl File	M
9	567890	brw-rw----	1	root	System	0	Nov200:10	/dev/sd0a	Block Spl File	M
10	980123	Dr--r--r--	X	root	System	0	Nov200:10	name_service_door	Door File	L
11	901234	prw-rw----	X	root	System	0	Nov200:10	/usr/lib/cron/FIFO	Pipe file	H
12	908761	Srwxrwxrwx	X	root	System	0	Nov200:10	/tmp/.X11-unix/X0	Socket file	H
13	567123	drwx--	1	ram	SA	1024	Dec 2 00:10	/etc/service	Directory file	H
14	712345	drwx--	1	USR	GRP	1024	July 6 12:23	/var/adm/message	Directory file	H

4. Problem Statement

There is a vital issue regarding the resource allocations of the multiple ROLE, RIGHT on UFS at various levels of resources management (Developer, Top, Medium & Lower mgmt.).

As per the literature survey and data collection, Preventive, Detective, and Control are not available on the recent Unix real-time operating system and the corrective action and reaction on the file system, application & resource are uncertain & unordered. The multiple Relation, Function, Operation, and Services are happening on multiple clients, businesses, applications, and resources over a complex multiple instruction data (MIMD) infrastructure. Therefore, resource conflicts are the biggest issue on a complex real-time operating system (RTOS) over multiple users & applications. Therefore, there is no balance ratio among the Business, RTOS & Resources.

4.1. Machine Level

The UFS is being used by multiple users and the single or multiple file system has many users as well as one or more processes, what will be the effect of multiple accesses with the open file systems. When are the modifications of file system made by one process observable to others or does anyone has to map the UGO to UFS & then RIGHT, ROLE of resources, and then processes through the attributes read, write and execute, due to risk mitigation over the real-time operating system to meet the user policy and procedure for the betterment of management decision? Each UFS is associated with an inode and that may be a soft link or hard link for one or more UGO. Each process links with the file system and the file system link (I-node) with role & right.

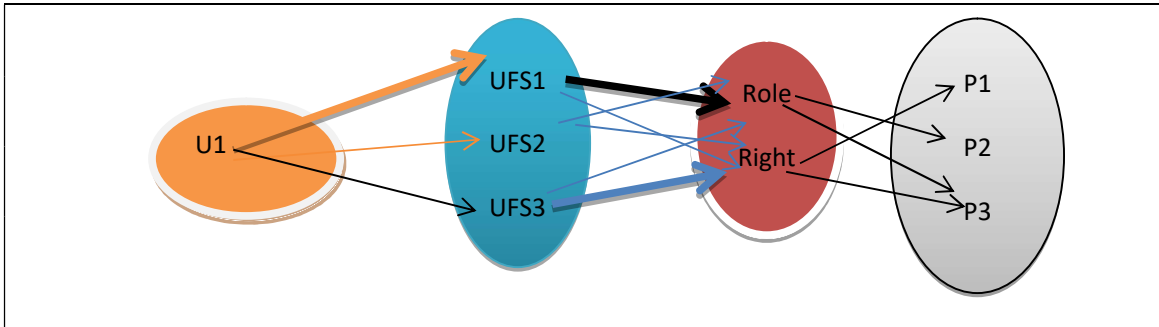


Figure 1. Pictorial Representation of USR Role & Right for UFS and Processes

The file system (UFS) being used by multiple users (1,2 3, ..., U_k) and having multiple Roles or Rights (R_1, R_2, \dots, R_m), the single or multiple file system has many users as well as one or more processes, what will be the effect of multiple accesses with the open file systems. When are the modifications of file systems made by one process observable to another process in the background of a Unix File System. The researcher has to map the UGO to the File system and then processes through the attributes read, write and execute, due to risk mitigation over a real-time operating system. Each UFS is associated with an inode and that may be a soft link or hard link. The shared lock for reading & exclusive for writing. When one process puts a write lock on the object, no other read lock or writes lock is allowed on the same object.

- One USR exists with one or many groups.
- One USR have one or many file system and One file system have one or many processes.
- One USR have one or many attributes (r, w, x), that USR link to one or many more file system (UFS). The author has to map the UGO to the File system and then Processes through the attributes
- read, write and execute, due to risk mitigation over a Unix RTOS. Each UFS is associated with an inode and that may be a soft link or hard link

5. Proposed Research Methodology

This research work contributes to the define, design, development of optimization and normalization that objective to determine the optimal cost, time and maximize the QoS to be developed and apply into the L Shape access control mechanism deciding on the measure components of UFS ACM (RWX) based on ordered & unordered combinatory method as follows.

5.1. Define

We have to define, design, develop and deploy the various method, models, mechanisms, services and fix up the major preventive access control (PAC) mechanism to maintain the desired level of risk. Meanwhile, the author can maintain the UFS ACM by applying automated Unix scripts on the real-time operating system, to optimize the risk and maximize the decision support to achieve the highest business objective.

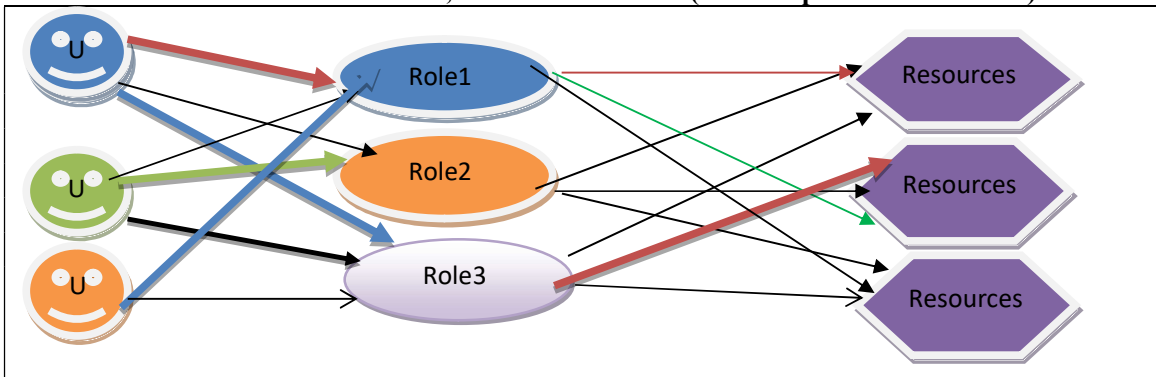
The top management needs to design & develop the policy, procedure to run a smooth business. The lower management needs to operate the services all the time and any time, but middle management has to co-ordinates and interacts in between top & lower management.

This proposed L shape pattern matching ACM can be designed and developed to protect against given types of threats, unauthorized users & uncertainty. These UFS ACM may range from simple to complex measures and usually involve system architectures, engineering disciplines, and security packages with a mixed culture of hardware, software, application, and firmware. All of these measures should work together to achieve better operation and services around the globe. These security controls can be decomposed into high, medium & low according to the primary purpose the day to day of the business.

Table 2. Allocation of octal, binary and UFS Permission to Resources

Octal	RWX	Permissions	Role&Right	Situation
0	000	None/Blank (-)	Nil	Safe
1	001	execute only(x)	Any One	Safe
2	010	write only(w)	Reserved	Un Safe
3	011	write and execute(w/x)	Reserved	Un-Safe
4	100	read only(r)	Top Mgmt	Uncertain
5	101	read and execute (r x)	Top Mgmt	Uncertain
6	110	read and write (r w)	Developer	Uncertain
7	111	read, write, and execute (full permissions) (r w x)	Developer	uncertain

Table 3. Allocation of USR, ROLE & Resource (Refer to problem Statement)



5.2 Design Methods and Mechanism of L-Shape RWX UFS ACM Model

We have to design, develop and implement the various method, models, mechanisms, services, and fix up major automated system configurations to maintain residual risk. Meanwhile, we are able to manage the PAC by applying an automated mechanism on a real-time operating system to optimize the risk and maximize the decision support to achieve the highest business objective.

Proposed Unix Machine Scripting Language

This proposed L-Shape UFS ACM model & mechanism protect and provide high-level data & services on any type of real-time system around the clock (7 x 24 x 52). This L-UFS ACM model maximum the protection on UFS, that application-optimized the cost and time. The prevention, detection, and correction at minimal cost with high availability of data and services as per business & resource requirements. Therefore, the stronger security on this UFS ACM model always depends on the allocation and distribution of Reading, Write & Executer over a UFS.

We are going to apply this scripting language, which is subsets of sub-programs, commands, command variables & build Unix commands. We can package these sets of commands into shell programming. This shell programming is used in the command mode of various shells like k Shell, Bourne Shell, Bash, etc as per availability of resources on the machine(`#ls -l; #chmod 111 menu.sh`). Our objective is that the PDC can be resolved through changing the UFS (chmod) and that should be meet the requirement of pervasive, ubiquitous, and self-autonomy.

The Set of eight attributes are { 0, r, w, x, rw, rx, wx, rwx }

We have to prove that the set {0, 1, 2, 3, 4, 5, 6, 7} = {B, R, W, X, RW, RX, WX, RWX}. is a finite combinatory order & under composition. Whereas, $S = \{0, 1, 2, 3, 4, 5, 6, 7\} = \{b, r, w, x, rw, rx, wx, rwx\}$. Our development process is that the Unix shell scripting apply to UFS ACM in a minimal cost & time.

Table 4. Allocation of UFS Attributes on RTOS

PREVENTIVE ACCESS CONTROL MECHANISM

Role Access mechanism	USR			GROUP			OTHERS			RISK	Remarks Un Safe
	Read	Write	Execu	Read	Write	Execute	Read	Write	Execute		
	r	w	x	r	w	x	r	w	x		
	4	2	1	4	2	1	4	2	1		
chmod 777 acm.sh	7	(rwxrwxrwx)		7	(rwxrwxrwx)		7	(rwxrwxrwx)		H	Developer
chmod 666 acm.sh	6	(rw-rw-rwx)		6	(rw-rw-rwx)		6	(rw-rw-rwx)		H	Developer
chmod 555 acm.sh	5	(r-xr-xr-x)		5	(r-xr-xr-x)		5	(r-xr-xr-x)		H	Top Mgmt
chmod 444 acm.sh	4	(r--r--r--)		4	(r--r--r--)		4	(r--r--r--)		M	Top Mgmt
chmod 333 acm.sh	3	(-wx-wx-wx)		3	(-wx-wx-wx)		3	(-wx-wx-wx)		H	Reserved
chmod 222 acm.sh	2	(-w--w--w-)		2	(-w--w--w-)		2	(-w--w--w-)		H	RESERVED
chmod 111 acm.sh	1	(--x--x--x)		1	(--x--x--x)		1	(--x--x--x)		L	Any
chmod 000 acm.sh	000			000			000				No access

Now we are going to apply the finite combinatory order elements.
Whereas, $S = \{0, 1, 2, 3, 4, 5, 6, 7\} = \{b, r, w, x, rw, rx, wx, rwx\}$.

Table 5. CUP-Shape Pattern Matching of UFS ACM -1NF (Unsafe-Uncertain,Semi Unification)

	U	G	O	SUM	Diff	Risk
UGO 76	7	7	7	21	-	H
UGO 54	6	6	6	18	3(1+1+1)	H
UGO 32	5	5	5	15	3(1+1+1) 0	M
UGO 10	4	4	4	12	3(1+1+1) 0	M
UGO	3	3	3	9	3(1+1+1) 0	H
UGO	2	2	2	6	3(1+1+1) 0	H
UGO	1	1	1	3	x	L

This CUP shape pattern of the ACM mechanism is only 20% resolving our purpose the rest 80% is defective and risk. This L-shape blue color 1st Column of U (7 6 5 4 3 2 1 0 in a decreasing way to down words) and last Row of UGO (1 1 1) is indicated as good one pattern for UFS ACM. The UGO Row & Column are not safe (Refers to Red Colour as 777, 666, 555, 444, 333, 222). The GO Row & Column is not a valid access control mechanism for our Business, because the risk is involved in role & right. Therefore, this pattern is 80% is defective. Then we have to move forward to the next pattern for G & O should be (1 1 1 or 0 0 0). We have to eliminate/replace the G and O value 7654321 by 1 or 0 (RED COLOUR). Grp and Others are always conflicted resources, therefore we have to bring down to 1 or 0 (UGO=711, 700) only. We have to decompose further to make ordered, fully unification and more simplification for all the time and every time around the globe.

Table 6. L Shape Pattern Matching of UFS ACM -1NF (Unsafe - Uncertain)

	U	G	O	SUM	Diff	Risk
UGO	7			21	-	H
UGO	6			18	3(1+1+1)	H
UGO	5			15	3(1+1+1) 0	M
UGO	4			12	3(1+1+1) 0	M
UGO	3			9	3(1+1+1) 0	H
UGO	2			6	3(1+1+1) 0	H
UGO	1			3	(X+X+X)	L

Note: We have to replace the G Column and O Column value 7654321 by 1 or 0 (Red Thick Arrow)

5.3. Development of L Shape UFSACM (Allocation & Distribution Mechanism)

We have to apply the UNIX FILE SYSTEM (UFS) Attributes on the following decomposition the table 5 and 6.

Furthermore, we have to do simplification and unification to avoid unsafe, unordered, unused as well as restricted attributes on the above composition table, that Read & Read- Execute belong to top management. The attributes are r (Read), rx (Read -Execute), rw(Read-Write), and rwx are needed for developers and x is for everyone all the time. Now, the attributes like Write (W) and Write-Execute (WX) do not assign roles & rights, due to high sensitivities, they become idle, therefore, we have to remove them from the cell or keep them reserved (Refer to Graph 2.). The Read & Read-Execute are common attributes for top management and developer. We can derive the data from the Table: 5 &6, then further we can decompose the L Shape ACM(2NF) as follows:

Table 7. L Shape Pattern Matching of UFS ACM -2NF- Safe-unification, certain & ordered

	U	G	O	SUM	Diff	Risk	RM%
UGO	7	1	1	9		M	80
UGO	6	1	1	8	1	M	80
UGO	5	1	1	7	1	M	80
UGO	4	1	1	6	1	M	80
UGO	3	1	1	5	1	M	80
UGO	2	1	1	4	1	M	80
UGO	1	1	1	3		L	80
UGO	0	0	0	0		L	80
	80	80	80	80	80	80	80



Note: We are going to achieve 80% safe & unification, because we assigned the role &right 1 to Group and others. 1st Column & Last Row is constructing L Shape Pattern. 1st Column-U & last Row UGO making safe & certain L shape for large scale business for top management, Why did researchers are focus on L Shape?, Because 1st Column & Last Row is designing, constructing, reflecting, and accounting for our purpose, Therefore, we are calling it as L Shape Pattern Matching of UFS ACM. 1st Column-U & last Row UGO making safe & certain L shape for large scale business for top management.

Table 7(a). For Grp pattern G (711, 611, 511, 411, 311, 211)

UIG	G	G	G	G	G	G	G
7-rwx							
6-rw							
5-rx							
4 r							
3 -wx							
2- w							
1 -x							
0 0	1x	1x	1x	1x	1x	1x	1x

This sub-table is derived from main table 7 for L-shape pattern matching, Now this is 1 for every row & column. 1st Column & Last Row is constructing L Shape Pattern. 1st Column-U & last Row UGO making safe and certain L shape for large scale business for top management.

Table 7(b). For Grp pattern G (700, 600, 500, 400, 300, 200)

UIG	G	G	G	G	G	G	G
7-rwx							
6-rw							
5-rx							
4-r							
3-wx							
2-w							
1-x							
0 0	0	0	0	0	0	0	0

This sub-table derived from main table 7 for L shape pattern matching, Now this Zero (0) for every row and column. 1st Column & Last Row is constructing L Shape Pattern. 1st Column-U & last Row UGO making safe & certain L-shape for large scale business for top management.

Table 8. L Shape Pattern Matching of UFS ACM -2NF - Safer, Ordered & unification

	U	G	O	SUM	Diff	Risk	RM%
UGO	7	0	0	7		M	90
UGO	6	0	0	6	1	M	90
UGO	5	0	0	5	1	M	90
UGO	4	0	0	4	1	M	90
UGO	3	0	0	3	1	M	90
UGO	2	0	0	2	1	M	90
UGO	1	0	0	1		L	90
UGO	0	0	0	0			90
RM%	90	90	90	90	90	90	90



Note 1: We are going to achieve more than 90% safer (ordered) & unification because the author assigned the Role and Right zero (0) to Group and others. The 1st Column & Last Row is constructing L Shape Pattern. The 1st. Column-U & Last Row UGO making safer & certain L shape for large scale protection to our model and mechanism for better risk assessment and decision system. Note 2: 300, 200 Reserved because Write is the High Risk and Write cannot be performed without Reading.

Table 8(a). For Others pattern O (711, 611, 511, 411, 311, 211)

U/O	0	0	0	0	0	0	0
7-rwx							
r-rw							
5-rx							
4-r							
3-wx							
2-w							
1-x							
0 0	1	1	1	1	1	1	1

This sub-table is derived from main table 7 for L shape pattern matching, Now this one (1) for every row & column. 1st Column & Last Row is constructing L Shape Pattern. The 1st Column-U & Last Row UGO making safer. certain, and unification L shape for large-scale protection system.

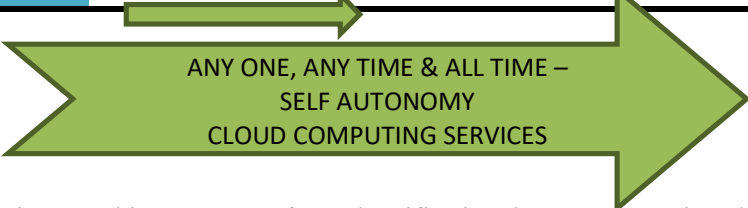
Table 8(b). For Others pattern O (700, 600, 500, 400, 300, 200)

U/O	0	0	0	0	0	0	0
7-rwx							
r-rw							
5-rx							
4-r							
3-wx							
2-w							
1-x							
0 0	0	0	0	0	0	0	0

This sub table derived from main table 7 for L shape pattern matching, Now this 0 for every row & column. 1st Column & Last Row is constructing L Shape Pattern. 1st Column-U & last Row UGO making safe & certain L shape for large scale business for top management.

Table 9. Unification of UFS Attributes 3NF (Safe-certain, ordered and Full-Unification for Everyone)


	U	G	O	SUM	Diff	Risk	RM%
UGO	1	1	1	3		L	100
UGO	1	1	1	3	1	L	100
UGO	1	1	1	3	1	L	100
UGO	1	1	1	3	1	L	100
UGO	1	1	1	3	1	L	100
UGO	1	1	1	3	1	L	100
UGO	1	1	1	1		L	100
	100	100	100	100	100	100	100


 ANY ONE, ANY TIME & ALL TIME –
 SELF AUTONOMY
 CLOUD COMPUTING SERVICES

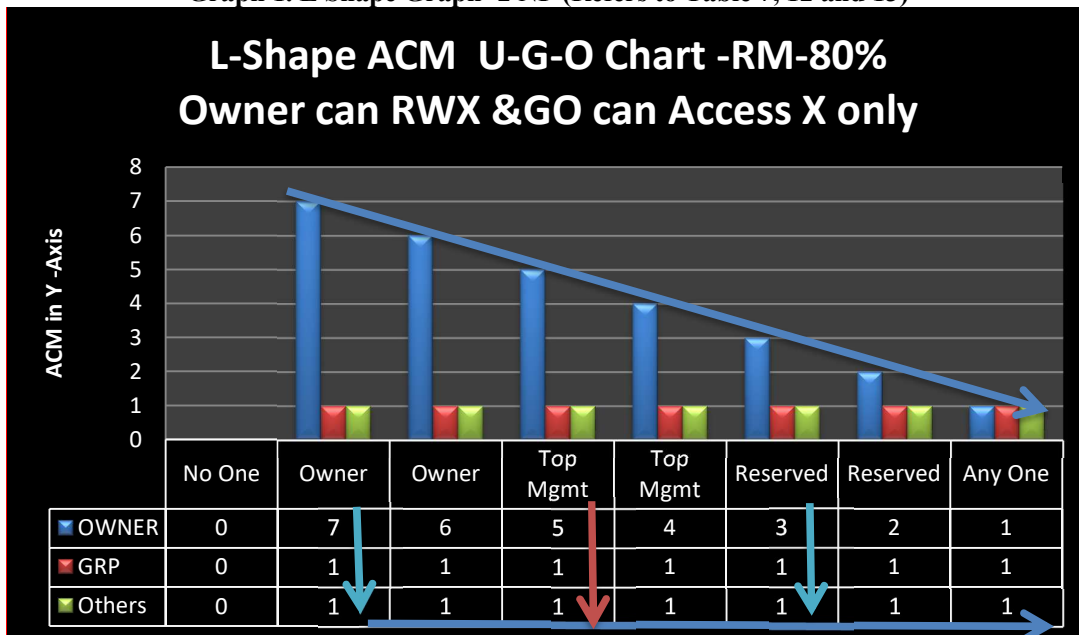
Note: Now, we are going to achieve 100% Safer and Unification, because we assigned the Role and Right 1 to everyone (User, Group, and others) all the time & every time, which will be supported to our IoT and Cloud computing. Now execute (x) access to all the Columns and all Rows are constructing L-Shape Pattern. The 1st Column-U & Last Row UGO making safer and certain the L-shape for large scale business for top management.

Table 10. RESERVED FOR FUTURE BCP (Full-Unification)

	U	G	O	SUM	Diff	Risk	RM%
UGO	0	0	0	0		L	100
UGO	0	0	0	0	0	L	100
UGO	0	0	0	0	0	L	100
UGO	0	0	0	0	0	L	100
UGO	0	0	0	0	0	L	100
UGO	0	0	0	0	0	L	100
UGO	0	0	0	0		L	100
UGO	100	100	100	100	100	100	100

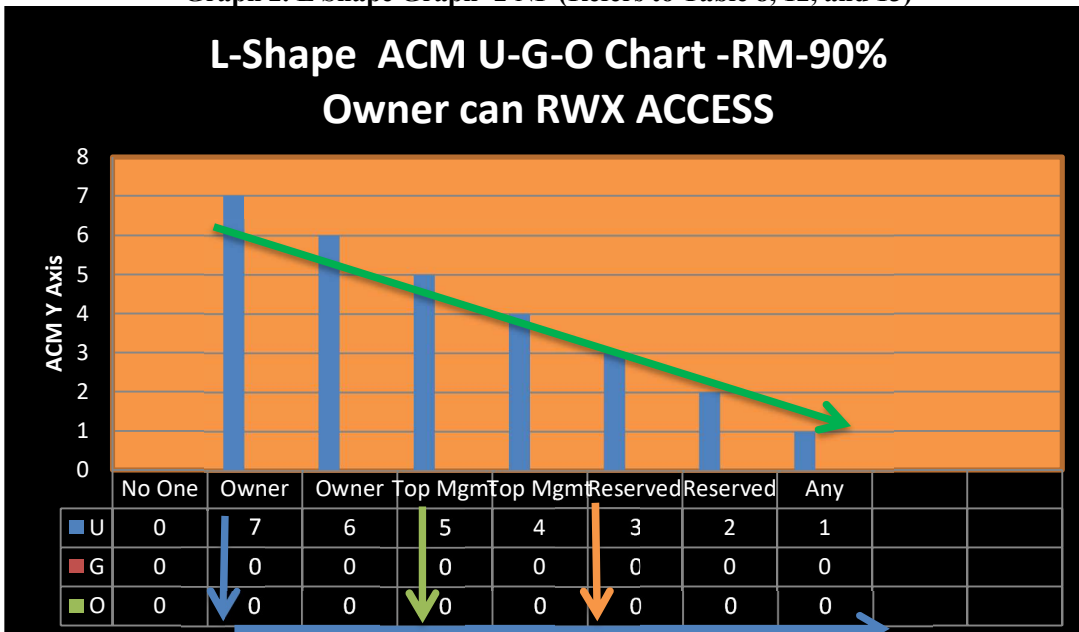

 NO BODY ACCESS- RESERVED FOR FUTURE BCP

Graph 1. L Shape Graph- 2 NF (Refers to Table 7, 12 and 13)



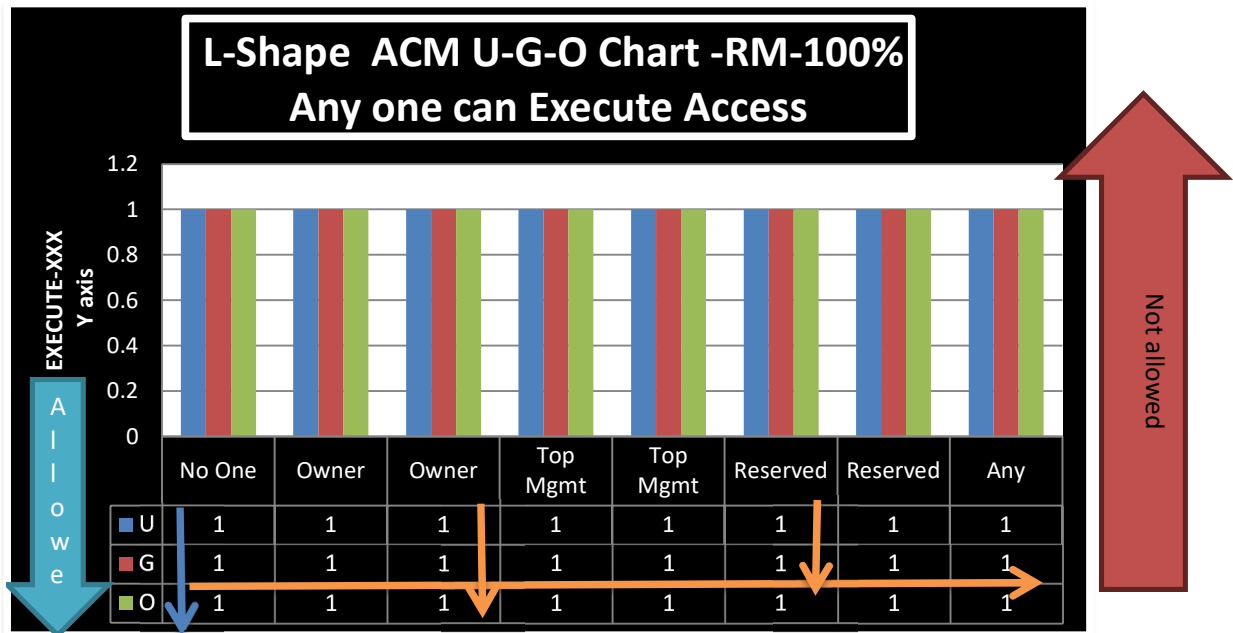
Note: Now G& O are achieving equal capability, responsibilities to perform their gobs (Access-Execute). Now all Column& Last Row is constructing L Shape Pattern except the last column. Now all Column& Last Row is constructing L-Shape Pattern except for the last column. In any situations the Last Column and Last Row itself is better for Cloud computing.

Graph 2. L Shape Graph- 2 NF (Refers to Table 8, 12, and 13)



Note: Now the Group and Other are nowhere to do their capability, responsibilities to perform their job (Access Deny). Now all Column& Last Row is constructing L-Shape Pattern except for the last column. Any is the situated at last Column and Last Row itself for IoT & EDGE computing.

Graph 3. L Shape Graph- 3 NF (Refers to Table 9, 12, and 13)



Now USR, GROUP & OTHERS are achieving equal capability, responsibilities to perform their jobs (Execute). Master (Owner) can seat anywhere, but G and O cannot seat in the place of Owner. Now all Column & Last Row is constructing L-Shape Pattern except the Last Column. Any is situated at last Column & Last Row itself for managing the cloud computing.

THUMB RULE FOR ACM:

The Master and Owner can seat anywhere and do their Role and Responsibilities, but User & Other cannot be allowed to do those capabilities.

The top-down flow of ACM is allowed, but down top (Lower to Higher) is not allowed. The Master can do the other's jobs, but others are not allowed all types of activities.

5.4 Action and Reaction of the TOP Management (RX-DSS)

The top management has Role and Right (100%) capabilities to define, deploy, test, verify, and validation of any ACM for customers & vendors at the right time and the right place for the betterment of the IT organization. The top management can able to decide on Software development and product overview at any time anywhere. The top management is also responsible for an overview of the operation and services of the real-time performance, benchmarking, throughput, fault tolerance, and ethical hacking on any type of hardware, software, networks, and application. How the application and business are performing and behaving on the right way when several users accessing the desired data & services all the time and every time in around the globe. The verities of the test can be done through this automated testing phase (Regression, Integration, Boundary, System, Alpha, and Beta). The Read & Execute the Role & Right job can be performed by top management, and then they can able to formulate the right policy and procedure for all the time and any time of anti-fragile technology (Refers Table 7, 8, 13 & 14 and Graph 1-2).

5.5. Experimental Mechanism and Analysis on UNIX Machine

We have to apply real time experiment on UFS ACM as follow

```
pl@pl-HP-15-Notebook-PC:~/log$ ls -iltra
```

```
      U  G  O
134208 -rwxrwxrwx 1plpl  727 2014-11-08 16:02 menu1.sh
141049 -rwxrwxrwx 1plpl  461 2014-11-08 16:17 menu4.sh Original UFS
141050 -rwxr-xr-x 1plpl  547 2014-11-08 16:37 menu5.sh
140886 -rwxrwxrwx 1plpl  505 2014-11-09 16:52 menu.sh
141063 -rwxr-xr-x 1plpl  839 2014-11-10 19:48 perftest.sh
141051 -rwxrwxrwx 1plpl  830 2014-11-10 19:56 performance.sh
270235 -rw-r--r-- 1plpl  235 2014-11-10 20:18 vmtxt
269614 drwxr-xr-x 3plpl 4096 2014-11-11 15:42 .
141036 drwxr-xr-x 35 plpl 4096 2014-11-24 15:01 ..
pl@pl-HP-15-Notebook-PC:~/log$
```

We can further apply the preventive access control(PAC) mechanism to optimize risk (Higher to Lower)

Dynamic Shell Scripting Programm.

```
pl@pl-HP-15-Notebook-PC:~/log$ cat acm.sh
```

```
#ACTION I. PREVENTIVE ACCESS CONTROL MECHANISM
#!/bin/sh
#menu.sh: Uses case to offer 8-item menu
#
echo " MENU- RISK ANALYSIS ON UNIX RTOS: UFS ACM Verification&
Validation\n
1. List out Current UFS(Attributes) \n2. List out Open File system\n3.
Long listing of UFS status)
4. Modification of UFS-ACM( UGO) \n5. Process Status of USER \n6.USER
status)\n7.IDof user on the current system\n8.list of attributes in
UFS(ACM)\n9. Quit to UNIX\nEnter your option: \c"
read choice
case "$choice" in
1)  ls -a>fl ;;
2)  lsof>lsof ;;
3)  ls -l> long ;;
4)  chmod711 menu*.* ;; (711-000)
5)  ps -aef | greppl>ps ;;
6)  users>ulist ;;
7)  who -Hart >wlist ;;
8)  ls -iltra>ufslist ;;
9)  exit ;; *) echo "Invalid option "      # ;; not really required for
the last option
esac
```

```
pl@pl-HP-15-Notebook-PC:~/log$ sh acm.sh ( Run the script on unix
command mode)
```

```
pl@pl-HP-15-Notebook-PC:~/log$ ls -l
-r-x--x--x 1 plpl  727 2014-11-08 16:02 menu1.sh
-r-x--x--x 1 plpl  461 2014-11-08 16:17 menu4.sh
-r-x--x--x 1 plpl  547 2014-11-08 16:37 menu5.sh
-r-x--x--x 1 plpl  505 2014-11-09 16:52 menu
-Remarks : USR or Business Owner could able to do READ, WRITE & EXECUTE,
But Other Group & Other can not do any things. That's why blank ----- is
there on subject)
```

Action on UFS ACM

```
pl@pl-HP-15-Notebook-PC:~/log$ chmod 777 menu*.sh (ACTION)
```

```
pl@pl-HP-15-Notebook-PC:~/log$ ls -l(Review the Reaction )
```

Table 11.ls -l (Long Listing of UFS ACM Table)

PC	Detection& Correct of UFS Detail							
Action	#chmod 711 menu*.sh (Correct)						RISK	1NF
Reaction #ls -l (DC)	-rwx--x--x 1 plpl	727	2014-11-08	16:02	menu1.sh	H	Role & Right Developer Top Mgmt	
	-rwx--x--x 1 plpl	461	2014-11-08	16:17	menu4.sh			
	-rwx--x--x 1 plpl	547	2014-11-08	16:37	menu5.sh			
	-rwx--x--x 1 plpl	505	2014-11-09	16:52	menu.sh			
Action	#chmod 700 menu*.sh							
Reaction #ls -l (DC)	-rwx----- 1 plpl	727	2014-11-08	16:02	menu1.sh	H	Role & Right Developer	
	-rwx----- 1 plpl	461	2014-11-08	16:17	menu4.sh			
	-rwx----- 1 plpl	547	2014-11-08	16:37	menu5.sh			
	-rwx----- 1 plpl	505	2014-11-09	16:52	menu.sh			
Action	#chmod 611 menu*.sh							
Reaction #ls-l (DC)	-rw---x--x 1 plpl	727	2014-11-08	16:02	menu1.sh	H	Role & Right Developer Top Mgmt	
	-rw---x--x 1 plpl	461	2014-11-08	16:17	menu4.sh			
	-rw---x--x 1 plpl	547	2014-11-08	16:37	menu5.sh			
	-rw---x--x 1 plpl	505	2014-11-09	16:52	menu.sh			
Action	#chmod 511 menu*.sh							
Reaction #ls- l (DC)	-r-x--x--x 1 plpl	727	2014-11-08	16:02	menu1.sh	M	Top Mgmt DSS	
	-r-x--x--x 1 plpl	461	2014-11-08	16:17	menu4.sh			
	-r-x--x--x 1 plpl	547	2014-11-08	16:37	menu5.sh			
	-r-x--x--x 1 plpl	505	2014-11-09	16:52	menu.sh			
Action	#chmod411 menu*.sh							
Reaction #ls-l (DC)	-r----x--x 1 plpl	727	2014-11-08	16:02	menu1.sh	M	Top Mgmt DSS	
	-r----x--x 1 plpl	461	2014-11-08	16:17	menu4.sh			
	-r----x--x 1 plpl	547	2014-11-08	16:37	menu5.sh			
	-r----x--x 1 plpl	505	2014-11-09	16:52	menu.sh			
Action	#chmod311 menu*.sh							
Reaction #ls-l (DC)	-r--wx--x 1 plpl	727	2014-11-08	16:02	menu1.sh	H	This is a invalid option, Write Cannot do, without Read	
	-r--wx--x 1 plpl	461	2014-11-08	16:17	menu4.sh			
	-r--wx--x 1 plpl	547	2014-11-08	16:37	menu5.sh			
	-r--wx--x 1 plpl	505	2014-11-09	16:52	menu.sh			
Action	#chmod211 menu*.sh							
Reaction #ls-l (DC)	-w--wx--x 1 plpl	727	2014-11-08	16:02	menu1.sh	H	This is a invalid option, Write Cannot do, without Read	
	-w--wx--x 1 plpl	461	2014-11-08	16:17	menu4.sh			
	-w--wx--x 1 plpl	547	2014-11-08	16:37	menu5.sh			
	-w--wx--x 1 plpl	505	2014-11-09	16:52	menu.sh			
Action	#chmod111 menu*.sh							
Reaction #ls-l (DC)	---x--x--x 1 plpl	727	2014-11-08	16:02	menu1.sh	L	Any Time, Any Body, Every whre Evey time	
	---x--x--x 1 plpl	461	2014-11-08	16:17	menu4.sh			
	---x--x--x 1 plpl	547	2014-11-08	16:37	menu5.sh			
	---x--x--x 1 plpl	505	2014-11-09	16:52	menu.sh			
Action	#chmod000 menu*.sh							
Reaction #ls-l (DC)	----- 1 plpl	727	2014-11-08	16:02	menu1.sh	L	No Access Highly Secure	
	----- 1 plpl	461	2014-11-08	16:17	menu4.sh			
	----- 1 plpl	547	2014-11-08	16:37	menu5.sh			
	----- 1 plpl	505	2014-11-09	16:52	menu.sh			

Now we are going to decomposed further to get our achievement better for anywhere and anytime.

Table 12. 2NF Brief summary of 1NF

PC	Detection of UFS Detail	RISK	2NF
Action	#chmod711 menu*.sh (Correct Control)		7+1+1=9
Reaction #ls -l (DC)	-rwx--x---x 1 plpl 727 2014-11-08 16:02 menu1.sh -rwx--x---x 1 plpl 461 2014-11-08 16:17 menu4.sh -rwx--x---x 1 plpl 547 2014-11-08 16:37 menu5.sh -rwx--x---x 1 plpl 505 2014-11-09 16:52 menu.sh	H Good	Role & Right Developer
Action	#chmod511 menu*.sh (Correct Control)		2 NF 4+1+1=6
Reaction #ls -l (DC)	-r-x--x---x 1 plpl 727 2014-11-08 16:02 menu1.sh -r-x--x---x 1 plpl 461 2014-11-08 16:17 menu4.sh -r-x--x---x 1 plpl 547 2014-11-08 16:37 menu5.sh -r-x--x---x 1 plpl 505 2014-11-09 16:52 menu.sh	M Better	Role & Right Top Mgmt
Action	#chmod111 menu*.sh		3NF 1+1+1=3
Reaction #ls-l (DC)	---x--x---x 1 plpl 727 2014-11-08 16:02 menu1.sh ---x--x---x 1 plpl 461 2014-11-08 16:17 menu4.sh ---x--x---x 1 plpl 547 2014-11-08 16:37 menu5.sh ---x--x---x 1 plpl 505 2014-11-09 16:52 menu.sh	I Best	Any Time, Any Body, Every whre Evey time

Table 13. 3NF UNIVERSAL UNIFICATION FOR EVERY ONE

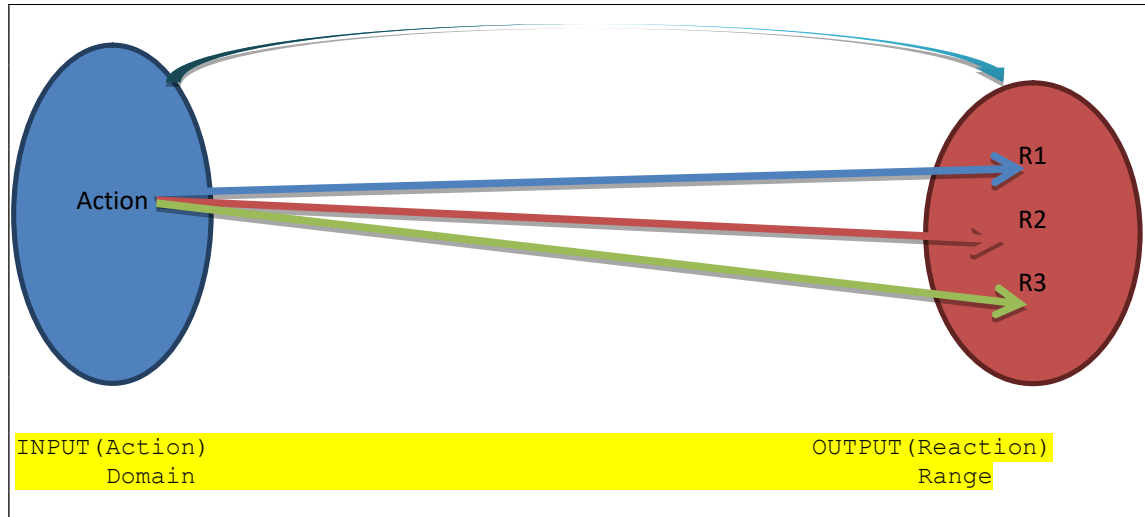
Action	#chmod111 menu*.sh		3NF
Reaction #ls-l (DC)	---x--x---x 1 plpl 727 2014-11-08 16:02 menu1.sh ---x--x---x 1 plpl 461 2014-11-08 16:17 menu4.sh ---x--x---x 1 plpl 547 2014-11-08 16:37 menu5.sh ---x--x---x 1 plpl 505 2014-11-09 16:52 menu.sh	I Best	Any Time, Any Body, Every whre Evey time

5. 6. Testing of L-Shape UFS ACM

The top management has to audit these above L Shape UFS ACM by real-time Unix shell scripting program. This ACM will be more scalable, high available, accountable for performance, fault tolerance, throughput, benchmarking, and risk optimization on any computational services all the time& every time. We have to make more simplification, unification, and step-by-step normalization by applying this ordered UFS ACM mechanism on the distributed object-oriented system on multi-dimensional work culture.

The author has to test and implement the following scripts on RTOS Unix-based platform for our secure, reliable, accountable, and high availability for multiple functions on multiple clients, applications, businesses, and resources anytime and anywhere around the globe. We have to audit the reliability, scalability, performance, benchmarking of this Real-Time System, when we are running this ACM Scripts by the following method, when millions of users access the Web Portal around the globe for all the time and any time (O' Reilly, 2002; Sun-Microsystems, 2003).

Table 14. ACTION Verses REACTION



The Unix shells scripts have the capabilities that, one action is creating many more reactions (results) that one going to prove on an experimental basis on the machine.

5.6.1 UNIX SHELL SCRIPTING FOR EXPERIMENTAL TEST

Why does the researcher need these Unix scripts?

When several Owners, groups & Others are working simultaneously on development, deployment, testing, and much more work is going on UFS ACM over a complex platform. The top management should be an overview of the performance, fine-tuning, throughput & benchmarking for fault tolerance. We have to apply this script on Unix Command Mode as follow:

```
#!/bin/sh
#menu.sh: Uses case to offer 8-item menu
#
echo " MENU- RISK ASSESSMENT ON REAL TIME UNIX OPERATING SYSTEMS:
PERFORMANCE, BENCHMARKING & FAULT TOLERANCE VERIFICATION-ANALYSIS\n
1. Listout CPU Status \n2. listout Memory status\n3. Free Memory(free)
4. Uptimes of the RTOS \n5. Load Factor of UNIX RTS \n6. Process Status of RTS\n7.Process
Tree\n8. Inter Process Comm\n9. Quit to UNIX\nEnter your option: \c"
read choice
case "$choice" in
1) top>toptxt ;;
2) vmstat -a >vmtxt ;;
3) free -mt>freetxt ;;
4) uptime>uptimetxt ;;
5) w>wtxt;;
6) ps -aufe>pstxt ;;
7) pstree | more >pstreetxt ;;
8) ipcs>ipcstxt ;;
9) exit ;;
*) echo "Invalid option " # ;; not really required for the last option
esac
# Save the program program-name.sh (source code file)
# Run the script # ./program-name.sh or # sh program-name.sh
# Output will be display interactively one by one in batch mode asper desired manner:
```

5. 7. Result of L-Shape UFS ACM

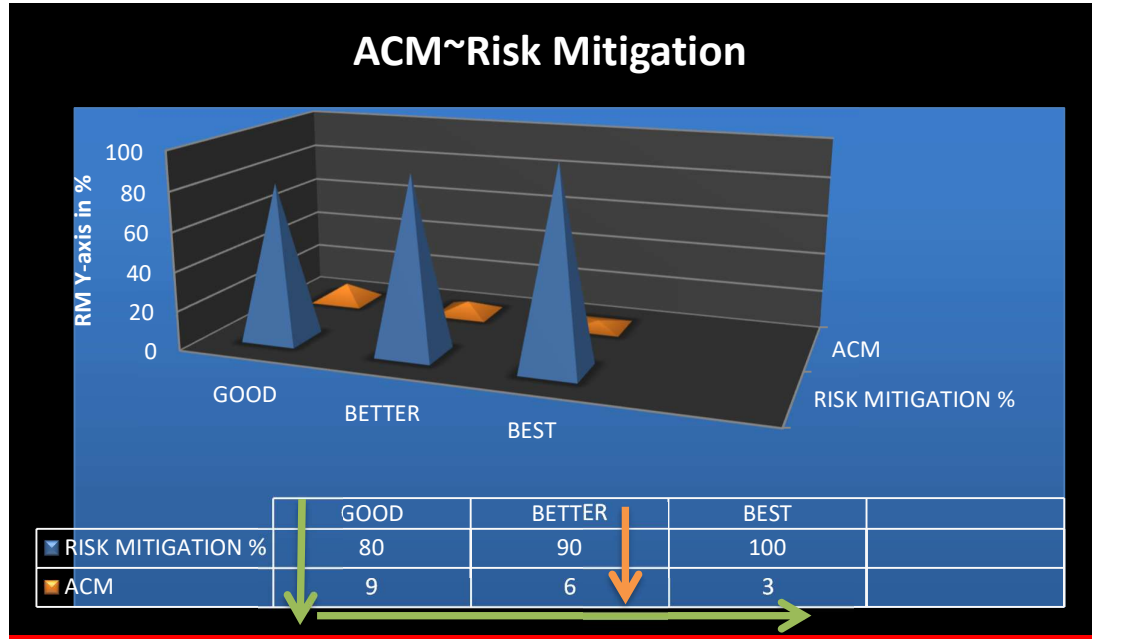
The author has concluded that the 0 & 1 values are available in table 7, 8,12 and 13 on 2NF & 3NF regarding unification, safe, accountability, measurability, action abilities, reliability & high availability of top management decisions for day-to-day operation and maintenances services. The Preventive, detective & corrective control can be performed by applying this L shape ACM on any UNIX Real-time operating system. The value 0 & 1 for Crips & Good Better Best for Fuzzy is fully applicable & utilising here to performed our research work satisfactory. Both CRIPS & FUZZY is satisfying our project work on table 7 - 8, Graph 1 – 2.

As per FUZZ'S LAW=> ACM= PC, PC=k. Cost : Where C is a Cost.

If ACM is High the risk is low.

GRAPHICAL REPRESENTATION

Graph 4. GRAPH L-Shape Pattern Matching UFS ACM for TQM, ROI, ROA



Note: Refers to Table 7, 8, 9, 13 & 14

Now, GOOD Column and ACM Last Row are constructing L- Shape Pattern except the last column. The Last Column is a better one as well is highly secure for cloud computing.

7. Discussion

- This proposed research work is fully satisfied with the following points:
- The top management can apply the dynamic decision support system (ROI, ROA, TCO & TQM).
- The design & developmental mechanism supports the safer, ordered certain, action, reaction, prevention, detection, correction, authentication, integrity, integration, performance, throughput, fault tolerance simultaneously on Unix Operating system for all the time and every time.
- This L Shape pattern matches UFS ACM minimizing the risk and maximizing the dynamic decision support system.
- This ACM dynamically supports the pervasive, ubiquitous, self-autonomy & anti-fragile system.
- Our proposed L Shape ACM mechanism is followed the soft computing principle of Crips & Fuzzy Rules simultaneously.
- This L- Shape pattern matching ACM is locally & globally integrated with Table2Table, NF2NF & Graph2Graph.

- This proposed research work is proving theoretical and & experimental results are available on Tabular, Graphical, and Normal Form, which is always safer, certain, ordered, unification & simplification for anytime and anywhere.

8. Conclusion

This L-Shape UFSACM will be great help in decision support system (DSS) for management, when the real-time system is under uncertainty, unordered, and unsafe. This PAC ACM Method, Model, and Mechanism and Control provide accountability for individuals who are accessing sensitive data information on the application, system software, server, and network. This is accomplished through RTOS that requires identification, authentication, authorization, accountability, non-repudiation, availability, reliability & integrity are available in the system security. This research paper is a practical idea and have been studied with both analytical & graphical methods. Key points of the UFS attributes and characteristics derived from the NF-NF, table to table & graph to graph applied to the pattern matching UFS ACM. The PAC mechanism detects the various level of the risk like High, Medium, and Low and manage the risk as per requirement for our normalization method. The risk analysis will be a great support to avoid conflict among the resources. In this way, we can achieve the operational and service goal and finally maintain better services that are satisfied with the Fuzzy Rule's If control is high, then Risk will be below.

References

- Andrew, Bill; Richard (2018) "UNIX Network Programming" New Delhi India, PHI
- Aloul, F., Al-Ali, A.R., Al-Dalky, R., Al-Mardini, M., El-Hajj. (2012). Smart grid security: Threats, vulnerabilities and solutions. *International Journal of Smart Grid and Clean Energy* 1(1), 1–6.
- Antonio F.S., Ramos Jose L.H., Moreno M.V.(March, 2014) A decentralized approach for Security and Privacy challenges in the Internet of Things. *Proceedings of the IEEE World Forum on Internet of Things*; Seoul, Korea, pp. 67–72.
- Bernard, K. (2007). *Discrete mathematical structures*. New Delhi, India: Person Education India (PHI).
- Busch, M., Koch, N., Suppan, S.(2014). Modeling Security Features of Web Applications. In: Heisel, M., Joosen, W., Lopez, J., Martinelli, F. (eds.) *Engineering Secure Future Internet Services*. LNCS, vol. 8431, pp. 119–139. Springer, Heidelberg.
- Beckers, K., Faßbender, S., Heisel, M., Küster, J.-C., Schmidt. (2012). Supporting the development and documentation of ISO 27001 Information Security Management Systems through security requirements engineering approaches. In: Barthe, G., Livshits, B., Scandariato, R. (eds.) *ESSoS 2012*. LNCS, vol. 7159, pp. 14–21. Springer, Heidelberg.
- Beckers, K., Hatebur, D., Heisel, M. (2013). A problem-based threat analysis in compliance with Common Criteria. In: *Proceedings of the International Conference on Availability, Reliability and Security (ARES 2013)*, pp. 111–120.
- Beckers, K., Heisel, M., Solhaug, B., Stølen.(2013).: ISMS-CORAS – A structured method for establishing an ISO 27001 compliant information security management system. Tech. Rep. A25626, SINTEF ICT (2013).
- Bertolino, A., Daoudagh, S., El Kateb, D., Henard, C., Le Traon, Y., Lonetti, F., Marchetti, E., Mouelhi, T., Papadakis. (2013). Similarity testing for access-control. Submitted to *Information and Software Technology*.
- Andrew Mallet. (2014). *CentOS System Administration Essential*, Packt Publishing <https://www.packtpub.com/product/centos-system-administration-essentials/9781783985920>
- CISA Review Manual 26th Edition. (2013). ISAC, USA.
- CISA Certification Guide. (2017). ISACA, USA

- Coriolis. (2017). CISSP exam cram. Coriolis Group Books. New Delhi, India: Dreamatech.
- D. Hussein, E. Bertin, and V. Frey. (2017). "A community-driven access control approach in Distributed environments," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 146–153.
- Fitzgerald, W.M., Turkmen, F., Foley, S.N., O’Sullivan, B.(2012). Anomaly analysis for physical access control security configuration. In: *Proceedings of the 7th International Conference on Risks and Security of Internet and Systems*.
- Fotiou N, Machas A, Polyzos GC, Xylomenos G (2014) Access control delegation for the cloud In: *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference On*, 13–18. IEEE, Canada.
- Hwang, Kai. (2008). *Advance computer architecture*. New Delhi, India: Tata McGraw Hill.
- H. Ren, Y. Song, S. Yang, and F. Situ.(2016). "Secure smart home: A voiceprint and internet-based authentication system for remote accessing," in *Proceedings of the 11th International Conference on Computer Science and Education, ICCSE 2016*, pp. 247–251.
- I.Agadakos, P. Hallgren, D. Damopoulos, A. Sabelfeld, and G. Portokalidis.(December 2016). "Location-enhanced authentication using the IoT because you cannot be in two places at once," in *Proceedings of the 32nd Annual Computer Security Applications Conference, ACSAC 2016*, pp. 251–264, USA.
- J. L. H. Ramos, M. P. Pawlowski, A. J. Jara, A. F. Skarmeta, and L. Ladid.(2015). "Toward a lightweight authentication and authorization framework for smart objects," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 4, pp. 690–702.
- Montesino, R., Fenz, S.: *Information security automation: How far can we go?* In: *Sixth International Conference on Availability, Reliability and Security (ARES 2011)*, pp. 280–285. IEEE Computer Society (2011).
- M. Cagnazzo, M. Hertlein, and N. Pohlmann.(2016). *An Usable Application for Authentication, Communication and Access Management in the Internet of Things*, Springer International Publishing, Cham, Switzerland.
- M. Trnka and T. Cerny.(2017). Authentication and authorization rules sharing for internet of things," *Software Networking*, no. 1, pp. 35–52.
- N. Shone, C. Dobbins, W. Hurst, and Q. Shi.(2015). Digital memories based mobile user authentication for IoT," in *Proceedings of the IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pp. 1796–1802.
- Padma Pradhan.(April-June, 2017). Proposed Heuristics Model Optimizing the Risk on RTS", *IJSDA6.2*. WOS 2160-9772.
- Padma Pradhan.(2017). Proposed Round Robin CIA Pattern on RTS for Risk Assessment: IS Security & Risk Assessment", *IJDCF9.1 WOS*, Scopus 1941-6210.
- Padma.(2018).Dynamic Scripting Language Optimizing theRisk on RTOS, *IJCNIS vol.10.9(47-59)*.
- R. Roman, J. Zhou, and J. Lopez.(2013). "On the features and challenges of security and privacy in distributed internet of things," *Computer Networks*, vol. 57, no. 10, pp. 2266–2279.dministrator
- Loger & Lom. (2017).*System Administrator Ethics*, Apress (<https://www.oreilly.com/library/view/system-administration-ethics/9781484249888>)
- S. Bandara, T. Yashiro, N. Koshizuka, and K. Sakamura.(2016). "Access control framework for API-enabled devices in smart buildings," in *Proceedings of the 22nd Asia-Pacific Conference on Communications, APCC 2016*, pp. 210–217.

- S. Lee, J. Choi, J. Kim.(2017). FACT: Functionality-centric access control system for IoT programming frameworks,” in Proceedings of the 22nd ACM Symposium on Access Control Models and Technologies, SACMAT 2017, pp. 43–54, USA.
- Shon, H. (2012). Security mgmt practices. New Delhi, India: Wiley Publishing Inc.
- Sumitabh, Das. (2017). UNIX system V UNIX concept & application. Delhi, India: Tata McGraw Hill.
- Sun-Microsystems.(2003). UNIX Sun Solaris system administration. USA
- S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Portisini.(2015). Security, privacy and trust in Internet of Things: the road ahead,” Computer Networks, vol. 76, pp. 146–164.
- Tran, L.M.S., Solhaug, B., Stølen, K.(2013) An approach to select cost-effective risk countermeasures. In: Wang, L., Shafiq, B. (eds.) DBSec 2013. LNCS, vol. 7964, pp. 266–273. Springer, Heidelberg.
- Tanenbaum. (2010). Operating System Design And Implementation. New Delhi, India: Person Education India (PHI).
- Tanenbaum.(2009). Computer Network. New Delhi, India: Person Education India (PHI).
- O’ Reilly. (2002). Essential of system administration. O’ Reilly Media. USA
- Weber, Ron. (2014). Information system control & audit. New Delhi, India: Person Education India (PHI).
- Trent R Hein.(2017). Unix and Linux System Administration Hand Book, 5th Edition, Addison Wesley, New Delhi, India.
- Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao.(2017). A Survey on Security and Privacy Issues in Internet-of-Things,” IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1250–1258.
- Xu, D., Thomas, L., Kent, M., Mouelhi, T., Le Traon.(2017). A model-based approach to automated testing of access control policies. In: Proc. of the 17th ACM Symposium on Access Control Models and Technologies (SACMAT), pp. 209–218.