Unix SVR4(OpenSolaris and illumos)Access Control

Salem Abdelhadi Ali Attia
ID:163106035

May 02, 2017

Outlines:

- What is Opensolaris
- What is illumos
- Protection Aim
- Principles of protection.
- Domain of protection
- Domain Components
- Matrix of Access Rights.
- Access Control List& Capatibility list
- Access Control Models
- Implement RBAC illumos
- Implement DAC OpenSolaris
- Command setfacl and getfacl in opensolaris
- Reference

What is Opensolaris?

- OpenSolaris is an open source computer operating system based on Solaris created by Sun Microsystems, later a part of Oracle Corporation.The first independent distribution was released on June 17, 2005.

- OpenSolaris is a descendent of the UNIX System V Release 4 (SVR4) [2]
- Open sourced subsequent to Solaris 10.
- Includes a variety of free software
- Including popular desktop and server software. [3]
- Slim Software ,automatic install
- Dtracre :Advanced debugging and tuning tool
- ZFS: huge capacity: 256 quadrillion(10^15) ZB (1 ZB = 1 billion TB)

Unfortunately

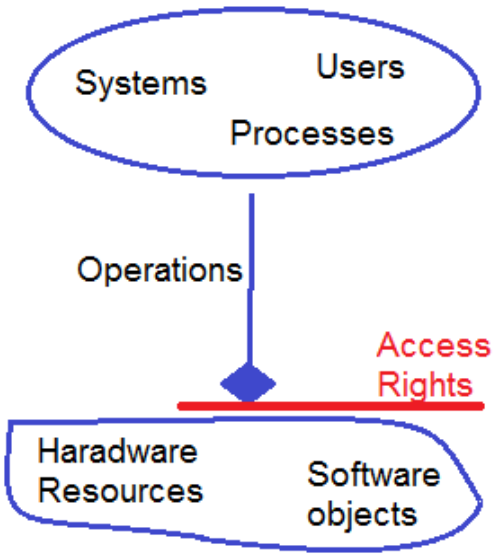
Sun was bought by Oracle in 2009, with the acquisition closing in February 2010, Orcale Close OpenSolaris
It became clear that Oracle had absolutely no interest in Open Source OS [6]

What is illumos?

- Illumos :is a free, open source operating system (OS).
- Target products (i.e. storage, virtualization, etc.)
- illumos was developed as a fork of (Unix SVR4. etc).
- Starting in the summer of 2010, Garrett D'Amore at Nexenta — with help from Rich Lowe, Jason King and others
- Network Server
- ZFS[7]
- Dtrace
- Oracle has no plans to work on illumos [5]
- Kernel-based virtual machine (KVM) supporting
- Role Based Access Control & Least Privilege [4]

Protection

- Protection refers to a mechanism for controlling the access of programs, processes, or users to the resources defined by a computer system.
- Protection ensures that each object accessed correctly and only by those processes that are allowed to do so.[11]



Protection Goals

- Operating system consists of a collection of objects, hardware or software
- Each object has a unique name and can be accessed through a well-defined set of operations
- Protection problem - ensure that each object is accessed correctly and only by those processes that are allowed to do so

Principles of Protection:

The role of protection in a computer system is to provide mechanism for the enforcement of the policies governing resource use.

- Mechanism vs Policy
- Mechanisms determine how something will be done; policies decide what will be done
- Guiding principle – principle of least privilege
- Programs, users and systems should be given just enough privileges to perform their tasks

- need-to-know principle: a process should be able to access only those resources that it currently requires to complete its task

.

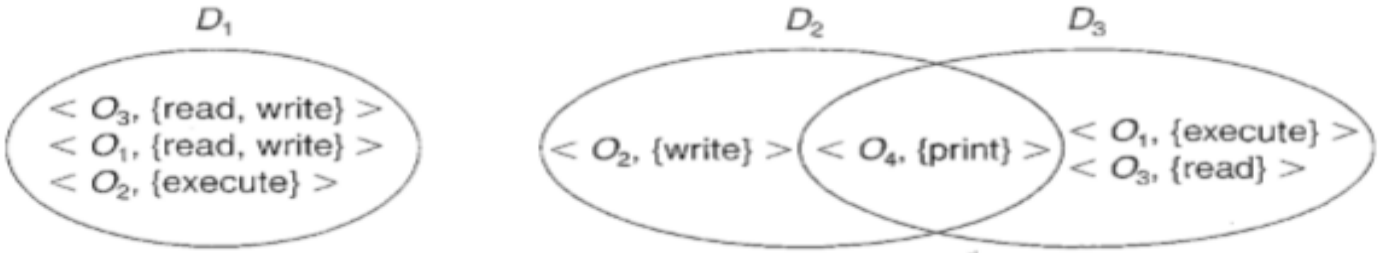
Domain Protection

- A process operates within a protection domain, which specifies the resources that the process may access.
- Each domain defines a set of objects and the types of operations that may be invoked on each object.
- The ability to execute an operation on an object is an access right.
- A domain is a collection of access rights, each of which is an ordered pair: <object-name, rights-set>

- Example: If domain D has the access right: <file F, {read, write}>, then a process executing in domain D can only read and write file F.

Domain Structure

- Access-right = <object-name, rights-set>
  where rights-set is a subset of all valid operations that can be performed on the object.
- Domain = a collection of access-rights
- A protection domain specifies the resources that the process may access [9]



Domains may share access rights.

- A process executing in either D2 or D3 can print O4
- A process must be executing in D1 to read and write O1. Also, only process in D3 may execute O1

Domain Implementation OpenSolaris :

System consists of 2 domains:

- User
- Supervisor(root)

 Domain = user-ID
 Domain switching corresponds to user ID switching
 Domain switching is accomplished through file system as follows:
 Each file has associated with it a domain bit (setuid bit) and an owner ID

.

when **setuid bit =off** user A can starts executing a file owned by user B and the the user ID of the process is set to A

But When ***setuid = on***, then user-id is set to owner of the file being executed: B. When execution completes user-id is reset. [9]

protection

How do we achieve Protection goals in OS

Access Controls one of These solutions

What is Access control

Determine whether a Subject can perform a requested operation on a Particular object

- Domain/Subject: user, process, etc.
- Operation/right: read, write , execute, etc.
- Object: file, tuple , printer, etc.

Access Matrix

- The access control matrix is a matrix with
  Each subject/domain represented by a row
  Each object represented by a column
- The entry *M[d, o]* lists the operations that domain *d* may carry out on object *o* [12]

Access Matrix

domain \ object	F_1	F_2	F_3	printer
D_1	read		read	
D_2				print
D_3		read	execute	
D_4	read write		read write	

C-List and Ac-List

Subject/Domain (User , Processes ,….) – Object (Files,)

How many Processes ,Users and files in your system?

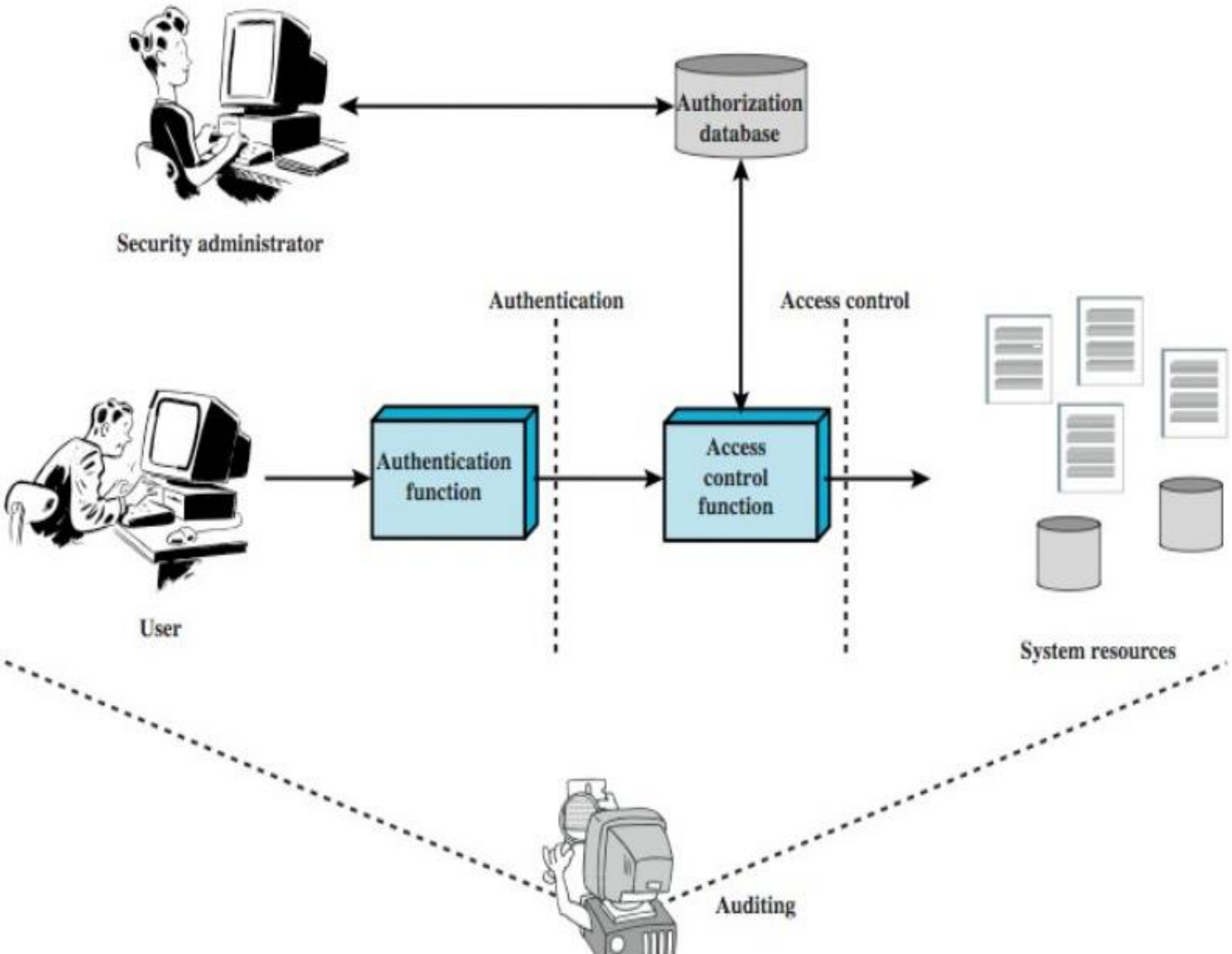
Each entity Of matrix consist set of access right

ACL F1 [(D1,r),(D4,rw)]

Most Of Operating System use Access control List Including "OpenSolaris"

Capability List D1[(F1,r),(F3,r)]

domain \ object	F_1	F_2	F_3	printer
D_1	read		read	
D_2				print
D_3		read	execute	
D_4	read write		read write	


Authorization database
Security administrator
Authentication
Access control
Authentication function
Access control function
User
System resources
Auditing

Access control Models

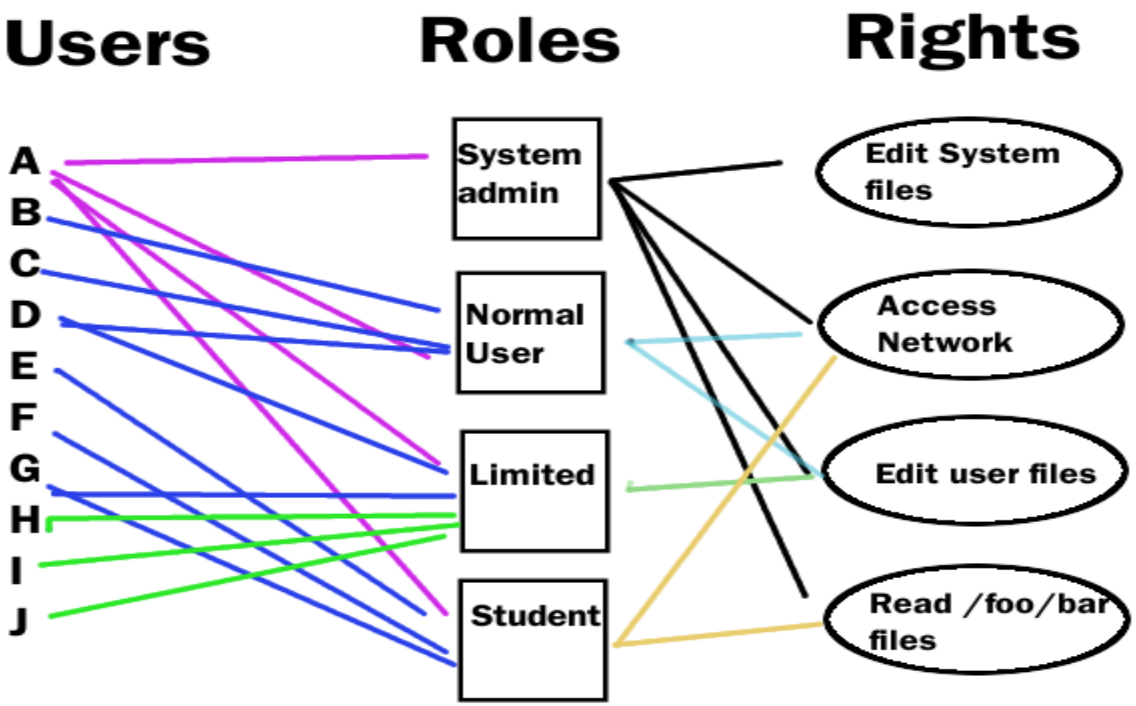
- Discretionary Access Control
  - Users (typically object owner) can decide permission assignments
- Mandatory Access Control
  - System administrator decides on permission assignments
- Role Based Access control
  - Roles have privilege to access objects And Subject /User Assiaction by That roles

Comparison of the features for the most common access control models.[8]

	DAC	MAC	RBAC
Authorization paradigm	Ownership	Administration	Role
Type of administration	Hard	Medium	Easy
Flexibility	High	Low	High
Good for distributed environments	Yes	No	Yes
Widely used	Yes	No	Yes
Handles dynamic changes	Yes	No	Yes
Handles task-based control	No	No	Yes
Level of security	Low	High	High
Level of assurance	Low	High	High
Incorporates easily into technologies (e.g., web services)	No	No	Yes
Able to express other models	No	No	Yes

illumos Used Role Based Access Control(RBAC)

illumos: RBAC - Role Based Access Control, for granting least-privilege access to processes and users.



Role Based Access control

Subjects are assigned Roles which have predefined associated permissions to perform certain operation on the objects.

The main features of RBAC are

- Centralized & Decentralized at once
- Permissions are enforced through Access Control List (ACL) attached to objects

Discretionary Access Control in OpenSolaris

- In this show you will get familiar with the implementation of DAC AC modol in OpenSolaris with the ZFS file system.
- traditional read=write=execute right for ACL entries, and similar to Windows, in OpenSolaris we have many more different permissions (e.g. append, read acl, etc.). For a some list refer to Table 3.1. In Linux ACLs are congured and viewed with the commands setfacl and getfacl - for example the [11]

Attribute	Description
execute	Execute a file
delete_child	Delete a file within a directory
read_attributes	Read basic attributes (non-ACL) of a file
write_attributes	Write basic attributes to a file or directory
delete	Delete a file
read_acl	Read the ACL
write_acl	Modify the ACL (needed to use chmod or setfacl)
write_owner	Use chown to change ownership of a file
synchronize	Access file locally via synchronous reads and writes

Some DAC In OpenSolaris

Setting an ACL on a File

```
$  setfacl -s user::rw-,group::r--,other:---,mask:rw-,user:george:rw-   ch1.doc
$  ls -l
total 124
-rw-r-----+  1 nathan  sysadmin   34816 Nov 11 14:16 ch1.doc
-rw-r--r--   1 nathan  sysadmin   20167 Nov 11 14:16 ch2.doc
-rw-r--r--   1 nathan  sysadmin    8192 Nov 11 14:16 notes
$  getfacl ch1.doc
# file: ch1.doc
# owner: nathan
# group: sysadmin
user::rw-
user:george:rw-    #effective:rw-
group::r--         #effective:r--
mask:rw-
other:---
```

Checking If a File Has an ACL

```
$  ls -l ch1.doc
-rwxr-----+  1 nathan   sysadmin      167 Nov 11 11:13 ch1.doc
```

Some DAC In OpenSolaris

The following example sets the file owner permissions to read/write/execute, file group permissions to read only, other permissions to none, and the ACL mask permissions to read on the ch2.doc file. In addition, the user george is given read/write permissions; however, due to the ACL mask, the effective permissions for george are read only.

```
$  setfacl -s u::7,g::4,o:0,m:4,u:george:7 ch2.doc
$  getfacl ch2.doc
# file: ch2.doc
# owner: nathan
# group: sysadmin
user::rwx
user:george:rwx          #effective:r--
group::r--               #effective:r--
mask:r--
other:---
```

Some DAC In OpenSolaris

- **Example--Copying an ACL**
- The following example copies the ACL on ch2.doc to ch3.doc.

```
getfacl ch2.doc | setfacl -f - ch3.doc
```

- **Example--Deleting ACL Entries on a File**

The following example deletes the user george from the ch4.doc file

```
$  setfacl -d user:george ch4.doc
```

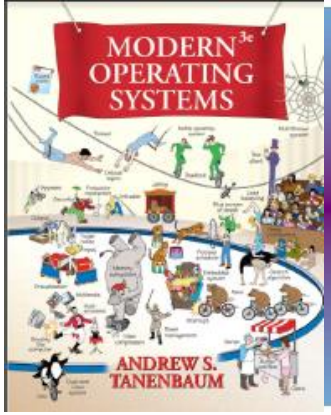
.

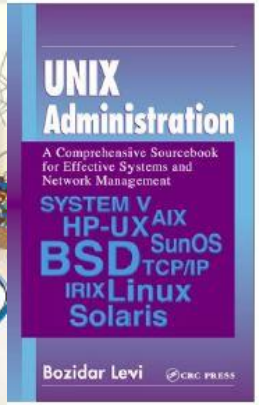
Some DAC In OpenSolaris

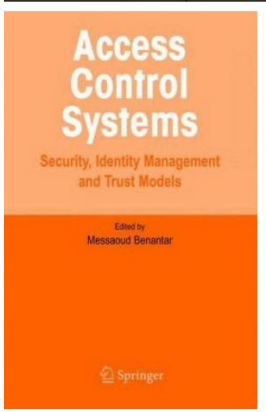
- **Examples--Displaying ACL Entries for a File**
- The following example shows all the ACL entries for the ch1.doc file. The #effective: note beside the user and group entries indicates what the permissions are after being modified by the ACL mask.

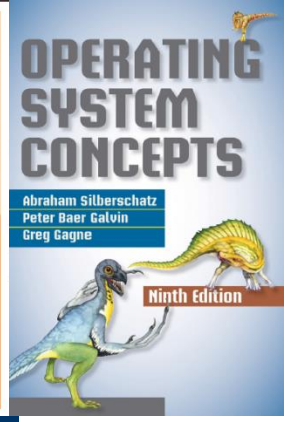
```
$  getfacl ch1.doc
# file: ch1.doc
# owner: nathan
# group: sysadmin
user::rw-
user:george:r--          #effective:r--
group::rw-               #effective:rw-
mask:rw-
other:---
```

References









- (Universit¨at Bochum) B. Cubaleska, A. Filyanov., http://www.ei.rub.de/media/trust/lehrmaterialien/265/e0b3e4694ed7fb83262b0a8ab5b61146a7d9c309/main-exercises.pdf, June 2011, Access 04-2017
- (UNIVERSITY OF NEBRASKA–LINCOLN) http://cse.unl.edu/~ylu/csce855/notes/access-control.ppt, , Access 04-2017
- [2] https://doc.lagout.org/operating%20system%20/Solaris/Zones%20Solaris/opensolaris-overview.ppt ., Access 04-2017
- [3] https://doc.lagout.org/operating%20system%20/Solaris/Zones%20Solaris/opensolarisfinal.ppt ., Access 04-2017
- [4] https://pt.slideshare.net/BrianBennett3/illumos-lopsa-sd?qid=11936b59-9895-4a4a-80c9-01368f2eb4c7_Brian Bennett_"Illumos — LOPSA SD"_Published at May-2015
- [5] http://webcache.googleusercontent.com/search?q=cache:9_uiAXYXFmAJ:searchstorage.techtarget.com/definition/illumos+&cd=1&hl=ar&ct=clnk&gl=ly
- [6] https://www.slideshare.net/BrianBennett3/illumos-lopsa-sd
- [7] https://www.openindiana.org/overview/illumos/, Access 04-2017
- [8] Ana Ferreira,"Modelling Access Control for Healthcare Information Systems",OCT 2010,p 5,https://www.cs.kent.ac.uk/pubs/2010/3078/content.pdf, ., Access 04-2017
- [9] ABRAHAM SILBERSCHATZ,PETER BAER GALVIN,GREG GAGNE OPERATING SYSTEM CONCEPTS 9ED ,WILEY,2013.
- [10] (Forum phoronix) ttps://www.phoronix.com/forums/forum/so ftware /oracle-solaris/45799-why-use-illumos., Access 04-2017
- [11] Ahmad-Reza Sadeghi, “System Security II”http://www.ei.rub.de/media/trust/lehrmaterialien/265/e0b3e4694ed7fb83262b0a8ab5b61146a7d9c309/main-exercises.pdf, ftware /oracle-solaris/45799-why-use-illumos., Access 04-2017
- [12] “**Access Control**” , http://cse.unl.edu/~ylu/csce855/notes/access-control.ppt,Access in 03-2017

The End

Thank You For All