

## *OpenVMS Security Update*

Helmut Ammer  
CSSC München

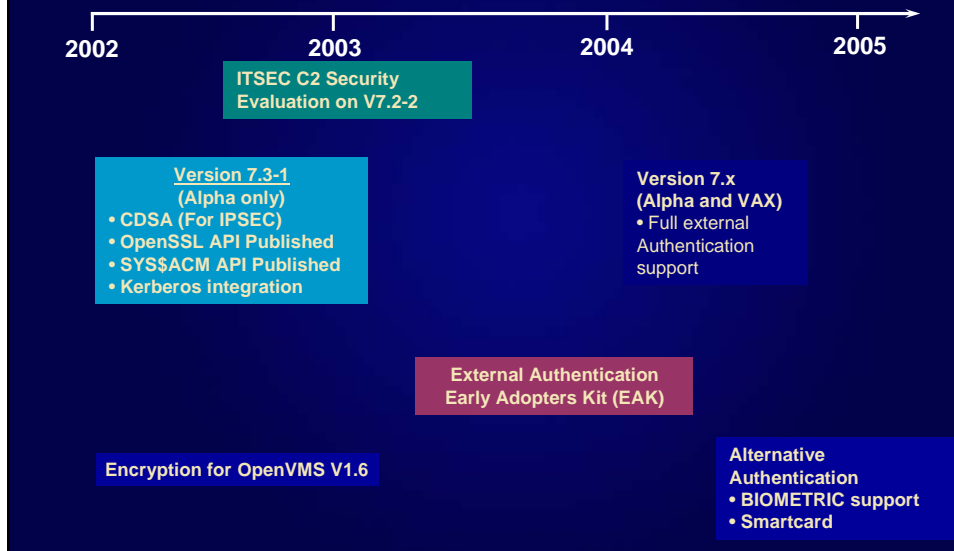
**1F06**

25. DECUS München Symposium Bonn 2002

## Überblick

- OpenVMS Security Roadmap
- Rückblick
- OpenVMS V7.3-1
- Zukunft
- ACLs

## OpenVMS Security Roadmap



## OpenVMS DECwindows MUP

DECwindows Motif Server has a potential Security vulnerability that could be exploited to allow existing users unauthorized access to data and system resources.

- CDs sind ausgeliefert. ECOs auf Webseiten
- Reboot notwendig
- Betroffen sind nur Systeme, welche DECwindows Server installiert haben
- Alle supporteten Versionen von OpenVMS Alpha, OpenVMS VAX, SEVMS VAX or SEVMS Alpha wurden darauf untersucht. Alle supporteten Versionen mit Ausnahme von OpenVMS VAX Version V5.5-2 sind betroffen

## DECwindows MUP

- Betroffene supportete Versionen:
  - OpenVMS Alpha Version 6.2 einschl. aller zugeh. Hardware Releases (z.B. Version 6.2-1H1)
  - OpenVMS Alpha Version 7.1-2
  - OpenVMS Alpha Version 7.2-1H1
  - OpenVMS Alpha Version 7.2-2
  - OpenVMS Alpha Version 7.3
  - OpenVMS VAX Version 6.2
  - OpenVMS VAX Version 7.1
  - OpenVMS VAX Version 7.2
  - OpenVMS VAX Version 7.3
  - SEVMS Alpha Version 6.2
  - SEVMS VAX Version 6.2

5

## OpenVMS V7.3

- Kerberos V1.0
  - based on MIT Kerberos Version 5 Release 1.0.5
  - Client & KDC Server
- Clusterwide Intrusion Detection
- OpenSSL integrated in CSWS (mod\_ssl)

6

## OpenVMS V7.3-1

7

## Kerberos

- Kerberos V1.0 Security Client integriert in OpenVMS V7.3-1
- Zuvor ein Layered Product

8

## OpenSSL for OpenVMS Alpha

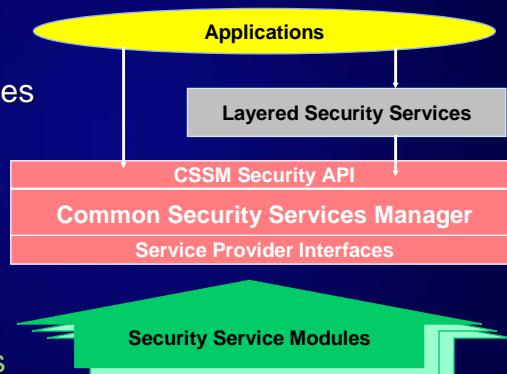
- Portierung von OpenSSL 0.9.6B
  - Layered Product (ab V7.2-2 installierbar)
  - PCSI Kit beinhaltet
    - 32-bit SSL & Crypt libraries
    - 64-bit SSL & Crypt libraries
- Eigenschaften:
  - 64-bit SSL und Crypto APIs (32 bit API's as well)
  - Dokumentation & Beispiele
    - Neues Manual – Open Source Security on OpenVMS Alpha
    - ~200 SSL APIs (60 zuvor undokumentiert)
    - ~40 Crypt APIs (10 zuvor undokumentiert)
  - Certificate Tool

## Common Data Security Architecture (CDSA)

CDSA definiert eine 4-layer Architektur für cross-platform, high-level Security Services

CSSM definiert ein common API & SPI für Security Services and Integrity Base

Service Provider implementieren selektierbare Security Services



<http://developer.intel.com/ial/security/>  
<http://sourceforge/projects/cdsa>

## CDSA for OpenVMS

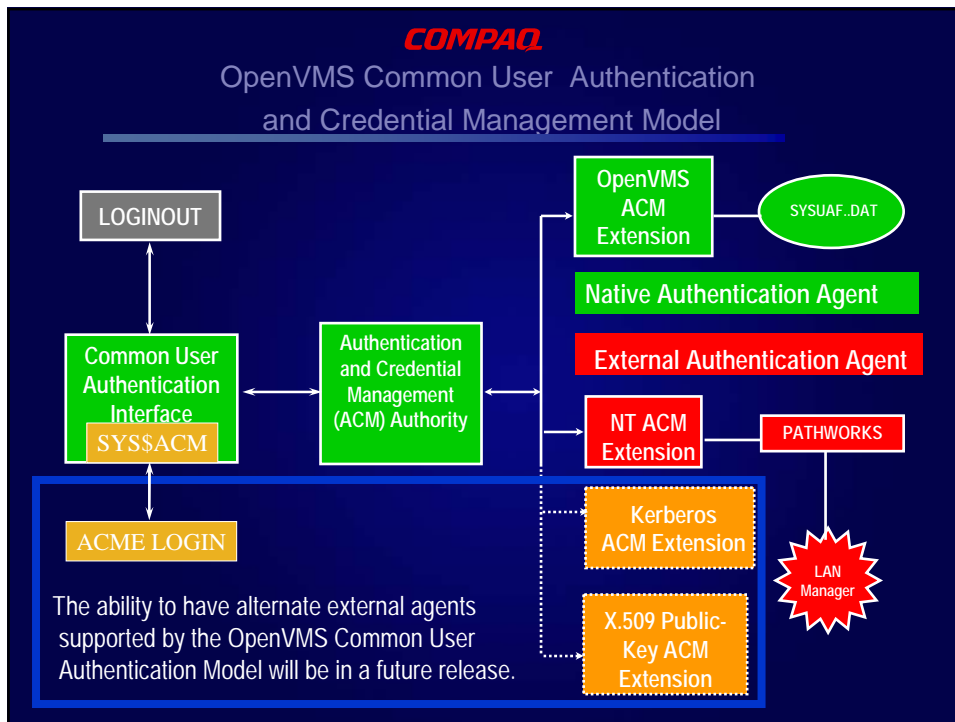
- Auslieferung als Teil von V7.3-1
- Installierbar ab OpenVMS V7.2-2
- Basiert auf Intel CDSA V2.0 Release 3
- Voraussetzung für IPSEC
- Enthält RSA & OpenSSL als Crypto Service Provider

11

## CDSA for OpenVMS

- CDSA beinhaltet:
  - CSSM Shared Library  
(Common Security Services Manager)
  - Header Files definieren CSSM APIs
  - CSPs (Cryptographic Service Provider)
  - MDS (Module Directory Services)  
ermöglicht Applikationen Service Provider zu lokalisieren

12



**COMPAQ**

## SYSSACM

- Veröffentlicht und supportet in V7.3-1
- Reduziert Authentication Calls/Schritte von 12 auf 1!
- Beispiel:  
CSWS for OpenVMS wird dies verwenden für Mod\_Auth\_vms
- Teil 1 der vollen External Authentication Lösung
  - Teil 2
    - NDA Document/EAK “ACME Developers Guide”
    - ACME Logout & Set Password

14

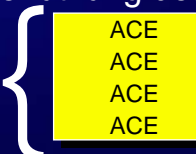
## ACLs

15

## Was sind ACLs und ACEs

- ACL = Access Control List
- Attribut eines Objekts
- ACL ist eine geordnete *Liste* von Access Control Entries, oder ACEs
- ACE Typ definiert
  - Erlaubt oder verbietet Zugriff aufs Objekt
  - Security Alarm oder Security Audit
  - Aktion beim Kreieren oder Benutzung des Objekts

ACL



16

## Objekte die ACLs unterstützen

- Files - Default
- Batch/Print Queues
- Devices
- Volumes
- System and Group Global Sections
- Logical Name Tables
- Common Event Flag Clusters
- Resource Domains
- Security Classes
- Capabilities

17

## Objekte die ACLs unterstützen

- Resource Domains
  - Namespace controlling lock manager resources
  - \$SET\_RESOURCE\_DOMAIN system service
- Security Classes
  - Parent of all classes of protected objects
  - Protects template profiles for objects
  - See OpenVMS Guide to System Security manual

18

## Beispiele

- ACL einer Logical Name Table

```

LNM$SYSTEM_TABLE object of class LOGICAL_NAME_TABLE
  Owner: [SYSTEM]
  Protection: (System: RWC, Owner: RWC, Group: R, World: R)
  Access Control List:
    ( IDENTIFIER=[PROXY,*],ACCESS=READ+WRITE)

```

- RESOURCE\_DOMAIN Security Class

```

RESOURCE_DOMAIN object of class SECURITY_CLASS
  Owner: [SYSTEM]
  Protection: (System: RW, Owner: RW, Group: R, World: R)
  Access Control List:
    ( IDENTIFIER=[TESTS],ACCESS=READ+WRITE+DELETE+CONTROL)

```

19

## Typen von ACEs

- Identifier ACE
- Default Protection ACE
- Creator ACE
- Alarm and Audit Journal ACE
- Subsystem ACE
- Application ACE

20

## Identifizier ACE

- Der gebräuchlichste ACE
- Zum Erlauben oder Verboten von bestimmten Zugriffsrechten für Personen oder Gruppen (UIC) oder Besitzer eines bestimmten Identifizierers oder *environmental* Identifizierers

21

## Identifizier ACE Format - Identifizierers

(IDENTIFIER=*identifizier*[+*identifizier*...]  
[,OPTIONS=*attributes*[+*attributes*...]],  
ACCESS=*access-type*+*[access-type*...])

- ACE Identifizier:
  - UICs
  - General identifizierers
  - Environmental identifizier
    - batch, network, interactive, local, dialup, remote

22

## Identifier ACE Format - Options

(IDENTIFIER=*identifier*[+*identifier*...]  
 [,OPTIONS=*attributes*[+*attributes*...]],  
 ACCESS=*access-type*[+*access-type*...])

- Identifier ACE Options:
  - Default
  - Hidden
  - Protected
  - Nopropagate
  - None
    - default case meaning “no attributes”

23

## Identifier ACE Format - Options

- **Default**
  - Applies to directory files only
  - Describes ACE to be placed on a file created in this directory
  - DEFAULT attribute removed from the ACE when propagated
  - Has no effect on object access
- **Hidden**
  - Indicates only application that created ACE '*should*' change it
  - Valid for all ACE types, but intended for application ACE
  - Need SECURITY privilege to display a hidden ACE

24

## Identifier ACE Format - Options

- **Protected**
  - Protects the ACE against casual deletion
  - Can only be deleted by
    - ACL Editor
    - \$ SET SECURITY /ACL=<ace> /DELETE
    - \$ SET SECURITY /ACL /DELETE=ALL
- **Nopropagate**
  - Indicates that the ACE cannot be copied by operations that usually propagate ACEs
    - \$ SET SECURITY /LIKE
    - \$ SET SECURITY /DEFAULT

25

## Identifier ACE Format - Access types

(IDENTIFIER=*identifier*[+*identifier*...]  
[,OPTIONS=*attributes*[+*attributes*...]],  
ACCESS=*access-type*[+*access-type*...])

- Identifier ACE Access Types for Files:
  - READ
  - WRITE
  - EXECUTE
  - DELETE
  - CONTROL
  - NONE

26

## Beispiel Identifier ACE

Mr. Spacely wants only members of the board reading the Idea Box:

```
DSK:[SPROCKET]IDEAS_FILE.TXT;1 [SYSTEM] (RWED,RWED,,)
  (IDENTIFIER=[BOD,*],ACCESS=READ)
```

...but he likes to be the only one to make suggestions!!

```
DSK:[SPROCKET]IDEAS_FILE.TXT;1 [SYSTEM] (RWED,RWED,,)
  (IDENTIFIER=[BOD,SPACELY],ACCESS=READ+WRITE)
  (IDENTIFIER=[BOD,*],ACCESS=READ)
```

27

## Beispiel Identifier ACE

To keep secrets from falling into the hands of the competitors, all access to the Idea Box is protected from dialup access:

```
DSK:[SPROCKET]IDEAS_FILE.TXT;1 [SYSTEM] (RWED,RWED,,)
  (IDENTIFIER=DIALUP,ACCESS=NONE)
  (IDENTIFIER=[BOD,SPACELY],ACCESS=READ+WRITE)
  (IDENTIFIER=[BOD,*],ACCESS=READ)
```

28

## Beispiel Identifier ACE

The directory containing sprocket test data is populated by many test teams, but George has to run the data reduction tools, so he always needs to have access:

```
DSK:[ SPROCKET]TEST.DIR;1 [SYSTEM] (RWED,RWED,,)
( IDENTIFIER=[TESTS,JETSON],OPTIONS=DEFAULT,ACCESS=READ)
```

When TEST\_1.DAT is created in the test directory the ACE propagates:

```
DSK:[ SPROCKET.TEST]TEST_1.DAT;1 [SYSTEM] (RWED,RWED,,)
( IDENTIFIER=[TESTS,JETSON],ACCESS=READ)
```

29

## Beispiel Identifier ACE

One day the marketing team needs access to the test data, but care has to be taken to prevent access to subsequent runs:

```
DSK:[ SPROCKET.TEST]TEST_1.DAT;1 [SYSTEM] (RWED,RWED,,)
( IDENTIFIER=[MARKET,*],OPTIONS=NOPROPAGATE,ACCESS=READ)
( IDENTIFIER=[TESTS,JETSON],ACCESS=READ)
```

The next version of the file has the following ACL:

```
DSK:[ SPROCKET.TEST]TEST_1.DAT;2 [SYSTEM] (RWED,RWED,,)
( IDENTIFIER=[TESTS,JETSON],ACCESS=READ)
```

30

## Beispiel Identifier ACE

One day the marketing team needs access to the test data, but care has to be taken to prevent access to subsequent runs:

```
DSK:[SPROCKET.TEST]TEST_1.DAT;1 [SYSTEM] (RWED,RWED,,)
  (IDENTIFIER=[MARKET,*],OPTIONS=NOPROPAGATE,ACCESS=READ)
  (IDENTIFIER=[TESTS,JETSON],ACCESS=READ)
```

The next version of the file has the following ACL:

```
DSK:[SPROCKET.TEST]TEST_1.DAT;2 [SYSTEM] (RWED,RWED,,)
  (IDENTIFIER=[TESTS,JETSON],ACCESS=READ)
```

31

## Beispiel Identifier ACE

The printer in the administration offices is for board members and Mr. Spacely's secretary. Henry needs to be able to fix things when they print Postscript to the ANSI queue:

```
LN03$PRINT: object of class QUEUE
  Owner: [SYSTEM]
  Protection:(System: RSDM, Owner: RSDM, Group, World)
  Access Control List:
    (IDENTIFIER=[BOD,*],ACCESS=SUBMIT)
    (IDENTIFIER=[JANE],ACCESS=SUBMIT)
    (IDENTIFIER=[HENRY],ACCESS=MANAGE+CONTROL)
```

32

## Default Protection ACE

- Applies to directory files only
- Used to describe what SOGW ("UIC-based") protection to apply to files that are created in this directory
- Default protection ACEs are propagated (unless marked NOPROPAGATE) to newly created subdirectories
  - Files in subdirectories will have the same default protection
  - The actual SOGW protection code is NOT applied to the subdirectory

33

## Default Protection ACE Format - Options

```
(DEFAULT_PROTECTION  
    [,OPTIONS=attribute[+attribute...]],access)
```

- Default Protection Options:
  - Hidden
  - Protected
  - Nopropagate
  - None

34

## Default Protection ACE Format - Access

(DEFAULT\_PROTECTION  
[,OPTIONS=attribute[+attribute...]],access)

- Default Protection Access
  - Example SOGW mask:  
(DEFAULT\_PROTECTION,S:RWED,O:RWED,G:RE,W)
  - Omitted user categories imply no access for that category

35

## Beispiel Default Protection ACE

The company has a public directory. All files in the directory should always be world readable:

```
DSK:[SPROCKET]PUBLIC.DIR;1 [SYSTEM] (RWED,RWED,RE,E)
      (DEFAULT_PROTECTION,S:RWED,O:RWED,G:RE,W:RE)
```

A File created here will get this protection mask:

```
DSK:[SPROCKET.PUBLIC]PUB.DOC [SYSTEM] (RWED,RWED,RE,RE)
```

A subdirectory created in the public directory will inherit this ACE:

```
DSK:[SPROCKET.PUBLIC]SUB.DIR;1 [SYSTEM] (RWE,RWE,RE,E)
      (DEFAULT_PROTECTION,S:RWED,O:RWED,G:RE,W:RE)
```

36

## Creator ACE

- Applies to directory files only
- Places an ACE on a newly created file describing access for the file's creator
- Only applied when the following conditions exist:
  - File is not owned by the UIC of the process creating the file
  - The process creating the file doesn't have system privileges

37

## Creator ACE Format - Options

```
(CREATOR [,OPTIONS=attribute[+attribute...]]  
      ,ACCESS=access-type[+access-type...])
```

- Creator ACE Options
  - Protected
  - Nopropagate
  - None

38

## Creator ACE Format - Access Types

```
(CREATOR [,OPTIONS=attribute[+attribute...]]
,ACCESS=access-type[+access-type...])
```

- Creator ACE Access Types:
  - READ
  - WRITE
  - EXECUTE
  - DELETE
  - CONTROL
  - NONE

39

## Beispiel Creator ACE

It was decided that the original submitter of a file to the public directory should retain full access to the file:

```
DSK:[SPROCKET]PUBLIC.DIR;1 [SYSTEM] (RWED,RWED,,)
(IDENTIFIER=PUB_ACCESS,ACCESS=READ+WRITE)
(DEFAULT_PROTECTION,S:RWED,O:RWED,G:RE,W:RE)
(CREATOR,ACCESS=READ+WRITE+EXECUTE+DELETE)
```

A file created here by George will inherit the protection mask and an ACE allowing George full access:

```
DSK:[SPROCKET.PUBLIC]PUB.DOC [SYSTEM] (RWED,RWED,RE,RE)
(IDENTIFIER=[JETSON],ACCESS=READ+WRITE+EXECUTE+DELETE)
```

40

## Alarm/Audit Journal ACE

- **First ACEs in the ACL, always enforced.**
- **Specifies the access that causes a security alert**
  - An ALARM ACE sends an alarm to all security terminals
  - An AUDIT ACE sends an audit message to the audit journal
- **Enabled only if ACL events are audited or alarmed**
  - \$ SET AUDIT /ALARM /ENABLE=ACL
  - \$ SET AUDIT /AUDIT /ENABLE=ACL
- **Disabled by turning off ACL audits or alarms**
  - \$ SET AUDIT /ALARM /DISABLE=ACL
  - \$ SET AUDIT /AUDIT /DISABLE=ACL
- **No effect on access**

41

## Alarm/Audit Journal ACE Format - Options

```
(AUDIT=SECURITY,[OPTIONS=attribute[+attribute...]]
,ACCESS=access-type[+access-type...])
```

```
(ALARM=SECURITY,[OPTIONS=attribute[+attribute...]]
,ACCESS=access-type[+access-type...])
```

- Alarm/Audit ACE options
  - Default
  - Hidden
  - Protected
  - Nopropagate
  - None

42

## Alarm/Audit Journal ACE Format - Access Type

```
(ALARM=SECURITY,[OPTIONS=attribute[+attribute...]]
,ACCESS=access-type[+access-type...])
```

- Alarm/Audit ACE Access Types:
  - Read
  - Write
  - Delete
  - Execute
  - Control
  - *and* SUCCESS or FAILURE or both

43

## Beispiel Alarm/Audit Journal ACE

Mr. Spacely wants to adequately monitor their accounting data file:

```
ACCOUNTNG.DAT;1          [SYSTEM]          (RWED,RWED,RE, )
  (ALARM=SECURITY,ACCESS=DELETE+CONTROL+SUCCESS)
  (AUDIT=SECURITY,ACCESS=DELETE+CONTROL+SUCCESS)
```

They'd also like to see anybody that plays with the payroll file:

```
PAYROLL.DAT;1          [SYSTEM]          (RWED,RWED,RE, )
  (ALARM=SECURITY,ACCESS=WRITE+DELETE+CONTROL+SUCCESS+FAILURE)
  (AUDIT=SECURITY,ACCESS=WRITE+DELETE+CONTROL+SUCCESS+FAILURE)
```

44

## Subsystem ACE

- Grants additional identifiers to a process while it is running the image to which the Subsystem ACE applies
- Applies to executable images only
  - Not applicable to sharable images
- Similar in function to installing an image with privs
- Must enable volume support of subsystem ACEs
  - \$ SET VOLUME /SUBSYSTEM

45

## Subsystem ACE Format - Options

```
(SUBSYSTEM,[OPTIONS=attribute[+attribute...],]
  IDENTIFIER=identifier
  [,ATTRIBUTES=attribute[+attribute...]]
  [,IDENTIFIER=identifier
  [,ATTRIBUTES=attribute[+attribute...]],...])
```

- Subsystem ACE Options
  - Protected
  - Nopropagate
  - None

46

## Subsystem ACE Format - Identifiers

```
(SUBSYSTEM,[OPTIONS=attribute[+attribute...],]
  IDENTIFIER=identifier
  [,ATTRIBUTES=attribute[+attribute...]]
  [,IDENTIFIER=identifier
  [,ATTRIBUTES=attribute[+attribute...]],...])
```

- Subsystem ACE Identifiers
  - UICs
  - General identifiers
  - Environmental identifiers
    - batch, network, interactive, local, dialup, remote

47

## Subsystem ACE Format - Attributes

```
(SUBSYSTEM,[OPTIONS=attribute[+attribute...],]
  IDENTIFIER=identifier
  [,ATTRIBUTES=attribute[+attribute...]]
  [,IDENTIFIER=identifier
  [,ATTRIBUTES=attribute[+attribute...]],...])
```

- Subsystem ACE Attributes
  - Resource
    - For file objects only
    - Identifier holders can charge disk space to the identifier

48

## Beispiel Subsystem ACE

The only application allowed to write to the master database is the Corporate Console. Only the Corporate Viewers can read it.

The SUBSYSTEM ACEs grant identifiers to the processes:

```
CONSOLE.EXE;1      [ SYSTEM]      ( RE,RE,E,E )
  ( SUBSYSTEM, IDENTIFIER=CONSOLE )
VIEWER.EXE;1      [ SYSTEM]      ( RE,RE,E,E )
  ( SUBSYSTEM, IDENTIFIER=VIEWER )
```

The database can then be protected using those identifiers:

```
DATABASE.DAT;1    [ SYSTEM]      ( RE,RE,, )
  ( IDENTIFIER=CONSOLE, ACCESS=READ+WRITE )
  ( IDENTIFIER=VIEWER, ACCESS=READ )
```

49

## Application ACE

- Application defined
- Application managed
  - Access via system services
- You'll know these when you see them!

```
DSK: [ SPROCKET] FILE.TXT      [ JETSON]      ( RWED,RWED,, )
  ( UNKNOWN=%X80, SIZE=%D163, FLAGS=%X0C00, ACCESS=%X06900000,
  DATA=%X00000008,%X00000001,%X1D1C07F7,%X0000FFFF,%X43020434,
  %X00030020,%X10654FDD,%X00000000,%XFD232200,%X1F1F1EFF)
```

50

## Creating and Managing ACLs

- Utilities that operate on ACLs
- System services that operate on ACLs
- Who can set up ACLs
- Order of access processing

51

## What Utilities Operate on ACLs and ACEs

- SET SECURITY /ACL
- SET ACL (obsolete)
- SET FILE /ACL (obsolete)
- EDIT /ACL
- ACE propagation
  - COPY
  - CREATE
  - RENAME
  - BACKUP
- SHOW SECURITY /ACL
- SHOW ACL (obsolete)
- SHOW FILE /ACL (obsolete)
- DIR /FULL
- DIR /SECURITY
- DIR /ACL (obsolete)

52

## Utilities - Set Security

- The V6+ way to access security attributes
  - SET SECURITY /ACL
  - SET SECURITY /DEFAULT
    - Applies template security settings to object as if newly created
- ```
$ SET SECURITY /DEFAULT FILE.DAT
```
- SET SECURITY /LIKE
    - Applies security settings from the named object
- ```
$ SET SECURITY /LIKE=NAME=FOO.DAT  
FILE.DAT
```

53

## Utilities - Set Security and Templates

- Besides security attributes on objects, all types of objects have security templates that also have security attributes
- ACLs, owner field, and protection codes set on the object template are inherited by the object when created
- \$ SHOW SECURITY /CLASS=SECURITY\_CLASS \*
  - Shows what security classes exist, and what the security settings are for the templates

54

## Who's Allowed to Put an ACL on an Object?

- The owner
- Users who have control access
- Users with privileges that grant control access
  - BYPASS, GRPPRV, SYSPRV
  - BYPASS, GRPPRV, SYSPRV, OPER for queues
  - BYPASS, GRPPRV, SYSPRV, SYSNAM for logical name tables

55

## What is the Order of Access Checking?

- Order of security checking
  - ACLs
  - then SOGW (also known as "UIC-based" protection)
    - **if object owner UIC is zero, protection code access is disabled!**
  - then privileges
    - BYPASS, GRPPRV, READALL, SYSPRV
    - OPER for queues
    - SYSNAM for logical name tables

56

## What Is The Order Of ACE Processing?

- ACEs processed from top to bottom
- The first matching identifier stops ACL processing
- If explicitly denied, only System and Owner protection codes checked, and then privileges
  - (IDENTIFIER=[FOO],ACCESS=NONE)
- If implicitly denied, all protection codes are checked, and then privileges.
  - (IDENTIFIER=[BAR],ACCESS=READ)

57

## Beispiel ACE Processing

It turns out that Mr. Spacely wants to be able to contribute ideas from home, so we need to rearrange the following ACL:

```
DSK:[SPROCKET]IDEAS_FILE.TXT;1  [SYSTEM] (RWED,RWED,,)
      (IDENTIFIER=DIALUP,ACCESS=NONE)
      (IDENTIFIER=[BOD,SPACELY],ACCESS=READ+WRITE)
      (IDENTIFIER=[BOD,*],ACCESS=READ)
```

We'll put the dialup access restriction at the end:

```
DSK:[SPROCKET]IDEAS_FILE.TXT;1  [SYSTEM] (RWED,RWED,,)
      (IDENTIFIER=[BOD,SPACELY],ACCESS=READ+WRITE)
      (IDENTIFIER=[BOD,*],ACCESS=READ)
      (IDENTIFIER=DIALUP,ACCESS=NONE)
```

58

## The Condensed Version

- ACLs allow finer granularity control over objects
- All OpenVMS security objects support ACLs
- Security messages for all object accesses
- Manage ACLs from DCL or system services
- ACLs on directories can describe ACLs automatically applied to files created there

59

## Fragen?



60

**COMPAQ**