OpenVMS Security

Presented by Wayne Sauer
PARSEC Group
999 18th Street, Suite 1725
Denver, CO 80202
www.parsec.com
888-4-PARSEC

HP Technology Forum & Expo 2008
Outline

- OpenVMS Security Design
- Physical Security
- Object Security
- UIC/ACL Security
- User Access
- Break-in Detection
- Network and Internet Considerations
- Encrypted Network Communication
- Kerberos
- Secure Socket Layer (SSL)
Goals

• Discuss the important points and consideration of OpenVMS Security
• Concentrate on the mechanics and mechanisms of OpenVMS features.
• Show how OpenVMS is one of the most secure operating systems on the market.
OpenVMS Security Design

- Security was designed into OpenVMS since V1.0
- Many different levels of security in OpenVMS
  - Physical Security
  - Object Security
  - User Management
  - Network Security
- Has never had a virus
Physical Security

- System
- System Console
- Storage devices and media
  - System Disk
  - Data and Database Volumes
  - Backups
- Network devices and media
Physical Security: System

- Increase system reliability through restricted access
  - Prevent intentional tampering and outage
  - Prevent outage due to accidents
- Prevent Front Panel Access
  - Halts
  - Reset/initializations
  - Power switch/source
  - Power on action settings (VAX) switch
Physical Security: Console

- Can be a big security hole for OpenVMS
  - Anyone with physical access to the console can break into OpenVMS by getting into the SYSBOOT utility.
  - Then OpenVMS can be broken into:
    - By redirecting startup
    - By changing SYSBOOT parameters
Physical Security: Getting to SYSBOOT on the Integrity Console Example

- On the Integrity shutdown to the EFI Boot Manager and select the EFI Shell and create a alias.

Please select a boot option

OpenVMS V8.2
Conversational Boot
DVD
OpenVMS Production
EFI Shell [Built-in]
Boot Option Maintenance Menu
System Configuration Menu

Use ^ and v to change option(s). Use Enter to select an option
Loading.: EFI Shell [Built-in]
EFI Shell version 1.10 [14.61]
Device mapping table
...
Shell> alias b "fs1:\efi\vms\vms_loader.efi"
Shell> b -fl 0,1

SYSBOOT>
Physical Security: Getting to SYSBOOT on the Integrity Console Example

• From the SRM prompt on the Alpha

```bash
>>> boot -flags 0,1 [device]
     (boot dkb300.3.0.13.0 -flags 0,1)
block 0 of dkb300.3.0.13.0 is a valid boot block
reading 1143 blocks from dkb300.3.0.13.0
bootstrap code read in
base = 1cc000, image_start = 0, image_bytes = 8ee00
initializing HWRPB at 2000
initializing page table at 3ffd0000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code

SYSBOOT>
```
Physical Security: Console Example

SYSBOOT> show /startup
Startup command file = SYS$SYSTEM:STARTUP.COM
SYSBOOT> set/startup opa0:
SYSBOOT> continue
...
$ set noon
$ spawn
spawn
%DCL-S-SPAWNED, process SYSTEM_132 spawned
%DCL-S-ATTACHED, terminal now attached to process SYSTEM_132
$ set noon
$ @sys$system:startup
...
$ mcr authorize
UAF> modify account_name /password...
### Physical Security: Console Example (Part 2)

```plaintext
SYSBOOT> **show maxsysgroup**
Parameter Name                  Current  Default  Min.   Max.  Unit  Dynamic
--------------------------------------- ------- ------- ------- ---- ------- -------
MAXSYSGROUP                     8        8       1      32768 UIC Group  D

SYSBOOT> **SET . %O200**

SYSBOOT> **SHOW .**

Parameter Name                  Current  Default  Min.   Max.  Unit  Dynamic
--------------------------------------- ------- ------- ------- ---- ------- -------
MAXSYSGROUP                   128        8       1      32768 UIC Group  D

SYSBOOT> **EXIT**

...$ a=128
$ show sym a
  A = 128   Hex = 00000080 Octal = 00000000200
$ mcr authorize show sauer

Username:  SAUER
Account:   STAFF
CLI:       DCL
Default:   STAFF:[SAUER]

Owner:  Sauer, Wayne
UIC:    [200,2] ([STAFF,SAUER])
Tables:  DCLTABLES
```

Physical Security: Satellite Console

- Preventing Conversational Booting on a Satellite
  - Prevent system modifications during boot
  - Should be disabled for unsecured workstations
  - Is not a dynamic parameter

The following example shows how to prevent conversational boot on which the node these commands are issued (this parameter value should also be set in MODPARAMS.DAT):

```
$ mcr sysgen
SYSGEN> use current
SYSGEN> set niscs_conv_boot 0
SYSGEN> write current
```
Physical Security: Storage Devices and Media

- **System Disk**
  - Normally co-located in the system
  - Sensitive security files normally located here
  - Secure all backups of the system disk
  - Ensure proper UIC/ACL security on system files

- **Non-System data**
  - Ensure all other database volumes are backed up on a regular basis.
  - All backup media should be kept secure
OpenVMS Object Security

- An Object is a component (hardware or software) of the system to which we apply permissions.
- Several different types of objects in OpenVMS.
- Objects have multiple levels of protection:
  - UIC (User Identification Code)
  - ACL (Access Control Lists)
  - Privileges
Types of OpenVMS Objects

- Capability (VAX Only)
- Common Event Flag Cluster
- Devices
- Files (including Directories)
- Global Sections
- ICC Associations
- Logical Name Tables
- Queues
- Resource Domains
- Security Class
- Volumes
OpenVMS Object Security Model

Rights to an Object

Object

- Owner UIC
- Protection Mask
- ACL (optional)

VA
HC
SC

UIC
- Privileges
- Rights List
- Identifiers

ACCESS
- Read
- Write
- Execute
- Delete
- Control
- Submit
- Manage
- Logical
- Physical
OpenVMS UIC Security

* UIC assigned to process when it is created
  - [group, member] is an octal number
  - Group numbers are any octal number between 1 and 37777
  - Member numbers are any octal number between 1 and 177777
  - Both group and member number 0 is reserved
OpenVMS UIC Security

- UIC assigned to an object to reflect the object's owner
  - Creator becomes the owner (unless the owner has a system UIC or SYSPRV, in which case the owner will be the owner of the directory)
  - Owner can change permission and ownership
- Any account that has a UIC group number equal to or less than the SYSGEN parameter MAXSYSGROUP automatically belongs to the system group
  - The System account UIC is [1,4]
OpenVMS UIC Security – Categories

**System** – determines access for any system UICs or a process with SYSPRV

**Owner** – determines the access for processes that have the same UIC as the object

**Group** – determines the access for processes that have the same group number as the object

**World** – determines access for all processes
OpenVMS UIC Security

Types of access

Read  allows a process to read the object, obtain information

Write  allows the process to modify or change the object

Execute allows the execution of the object, a command procedure or image

Delete  allows the process to remove the object

Control allows the process to change the security of the object and is implied with ownership (ACL only)

Example syntax is (S:RWED, O:RWED, G:RE W)
OpenVMS UIC Security

Types of access (Continued)

Create   In the case of volumes, allows the process to create files.

Manage   In the case of queues, allows the process to control characteristics

Submit   In the case of queues, allows the process to submit/print to the queue

Logical  Allows logical I/O to devices

Physical Allows physical I/O
Rights to an Object

Object

Owner UIC
[100,1]
Protection Mask
(S:rwed, O:rwed, G:re, W:)

UIC
[100,1]
[100,2]
[101,1]
[1,4]

Privileges
NETMBX
TMPMBX

HC

SC

VAS

IMAGE

OpenVMS UIC Security
OpenVMS UIC Security

Summary of commands

$ SET FILE /PROTECTION=(mask) /OWNER=[uic] file-spec
$ SET FILE/OWNER_UIC=(uic)
$ SET DIRECTORY/OWNER_UIC=(uic)
$ SET PROTECTION=(mask) file-spec
$ SET SECURITY /PROTECTION=(mask) /OWNER=[uic] file-spec
$ SET PROTECTION/DEFAULT
$ SET QUEUE/PROTECTION=(mask)
$ SET QUEUE/OWNER_UIC=(uic)
OpenVMS ACL Security

- Base all security on UIC and use ACL as the exception
- Uses Right Lists Identifiers
- Identifiers are added to the RIGHTSLIST.DAT file by the System Administrator
- Identifiers are then granted to users typically via the AUTHORIZE Utility
- An ACE (Access Control Entry) within the ACL contains Identifiers and the access allowed them
OpenVMS ACL Security

- When the user logs on, the identifier is included in the process rights list.

- Process rights list may be modified on the fly if it is added to the RIGHTSLIST.DAT with a dynamic attribute or the process has CMKRN1 privilege.
OpenVMS UIC Security

Rights to an Object

Object
UIC=[17,4]
(RWED,RWED,RE,)
ACL
Identifier=Parsec1, Access=read
Identifier=Blue, Access=noaccess
Identifier=Testacl, Access=all

Process
rightslist
Blue
Parsec1
Testacl

UIC
[100,765]

Privileges
NETMBX
TMPMBX
Example, adding and granting an identifier:

UAF> **add/id testacl**

%UAF-I-RDBADDMSG, identifier TESTACL value %X80010261 added to rights database

UAF> **grant/id testacl parsecl**

%UAF-I-GRANTMSG, identifier TESTACL granted to PARSEC1

UAF> **show/id testacl**

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TESTACL</td>
<td>%X80010261</td>
<td></td>
</tr>
</tbody>
</table>

UAF>

UAF> **show/id testacl/full**

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TESTACL</td>
<td>%X80010261</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Holder</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARSEC1</td>
<td></td>
</tr>
</tbody>
</table>

UAF> **show/rights parsecl**

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Value</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TESTACL</td>
<td>%X80010261</td>
<td></td>
</tr>
</tbody>
</table>

UAF> **exit**
OpenVMS ACL Security

Example, Using a UIC Identifier:

Username: parsecl
Password:

Welcome to OpenVMS (TM) Alpha Operating System, Version V8.3 on node CLASS3
Last interactive login on Friday, 14-MAR-2008 15:00:17.58

$ show proc/priv
Node: CLASS3           Process name: "PARSEcl"

Authorized privileges:
NETMBX       TMPMBX

Process privileges:
NETMBX           may create network device
TMPMBX           may create temporary mailbox

Process rights:
PARSEC1                           resource
INTERACTIVE
REMOTE
TESTACL
...
$
OpenVMS ACL Security

Example, Using a UIC Identifier:

$ type [mehlhop.webinar]a.a
%TYPE-W-OPENIN, error opening $22$DKA300:[MEHLHOP.WEBINAR]A.A;1 as input
-RMS-E-PRV, insufficient privilege or file protection violation
$ lo

From a privileged account or an account that has write access to the file

CLASS3$ set security/acl=(id=parsec1,access=read) a.a
CLASS3$ dir/security a.a

Directory $22$DKA300:[MEHLHOP.WEBINAR]

A.A;1 [STAFF,MEHLHOP] (RWED,RWED,RE,)

(IDENTIFIER=[PARSEC1],ACCESS=READ)

Total of 1 file.
CLASS3$
Example: Using a UIC Identifier

CLASS3$ set host 0

Welcome to OpenVMS (TM) Alpha Operating System, Version V8.3

Username: parsecl
Password:

Welcome to OpenVMS (TM) Alpha Operating System, Version V8.3 on node CLASS3

Last interactive login on Monday, 17-MAR-2008 17:25:16.01

$ type [mehlhop.webinar]a.a
This is a test file

$
Example: Using a General Identifier

$ type [mehlhop.webinar]b.b
%TYPE-W-OPENIN, error opening $22$DKA300:[MEHLHOP.WEBINAR]B.B;1 as input
-RMS-E-PRV, insufficient privilege or file protection violation
$ lo

From a privileged account or an account that has write access to the file

CLASS3$ set security/acl=(id=testacl,access=read) b.b
CLASS3$ set security/acl=(id=testacl,access=read) b.b
CLASS3$ dir/sec b.b

Directory $22$DKA300:[MEHLHOP.WEBINAR]

B.B;1 [STAFF,MEHLHOP] (RWED,RWED,RE,)
(IDENTIFIER=TESTACL,ACCESS=READ)

Total of 1 file.
CLASS3$
Example: Using a General Identifier

CLASS3$ set host 0

Welcome to OpenVMS (TM) Alpha Operating System, Version V8.3

Username: parsecl
Password:
Welcome to OpenVMS (TM) Alpha Operating System, Version V8.3 on node CLASS3
Last interactive login on Monday, 17-MAR-2008 17:35:14.15

$ type [MEHLHOP.WEBINAR]b.b
Test file to be examined by using a general identifier
$
OpenVMS Security: Privileges

- **BYPASS** – Bypass all protections
- **READALL** – Bypass protections for read access only
- **SYSPRV** – Access an object using the SYSTEM category protection mask
- **GRPPRV** – Access an object using the SYSTEM category protection mask if the user has the same group number as the object
- **VOLPRO** – Overrides volume protection
- **IMPERSONATE** – Allows a process to create or assume a persona
All user account information for the system or cluster is in the User Authorization File (SYSUAF.DAT).

The SYSUAF.DAT file is not an ASCII file and can be modified by using the AUTHORIZE utility.

Profile in the SYSUAF.DAT file is identified by the USERNAME and contains:

- Identification information
- Quota and limit settings
- Privileges
- Encrypted password
OpenVMS User Access to the System

- A Username and Password must be entered (minimum 1 maximum 2 passwords/username and possibly one system password)
- Username identifies the record in the SYSUAF.DAT file
- The entire password is verified by OpenVMS by encrypting the password that was entered and comparing it with the encrypted password field in the SYSUAF.DAT record
- You can set a password minimum length and lifetime with the commands in the AUTHORIZE utility:
  
  UAF> MOD username/PWDMINIMUM=
  UAF> MOD username/PWDLIFETIME=
OpenVMS User Logon

- All passwords are pre-expired by default when reset or the account is created by the system manager.
- The following UAF FLAGS affect the security of the Username and Password:
  - DISFORCE_PWD_CHANGE
  - DISPWDDIC
  - DISPWDHIS
  - DISUSER
  - GENPWD
  - LOCKPWD
  - PWD_EXPIRED
  - PWD2_EXPIRED
  - PWDMIX
Additional UAF Flag Keywords

- Additional UAF flag keywords that affect security:
  - AUDIT
  - AUTOLOGIN
  - CAPTIVE
  - DEFCLI
  - DISCTLY
  - DISIMAGE
  - RESTRICTED
OpenVMS Break-in Detection

- OpenVMS employs automatic break-in detection and evasion.
- Once a login failure occurs, the user becomes a suspect and is monitored by the system.
- Suspects become intruders by exceeding their allowed login failures during the monitoring period.
OpenVMS Break-in Detection

• Login failures are logged into the intrusion database, and is maintained by the Security Server process.

• You can display content of the database by issuing the DCL command $ SHOW INTRUSION.

• You can delete content of the database by issuing the DCL command $ DELETE/ INTRUSION.
OpenVMS Break-in Detection

- Login behavior is controlled through the following set of dynamic SYSGEN parameters

```bash
$ mcr sysgen
SYSGEN> show /lgi
Parameters in use: Active

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Current</th>
<th>Default</th>
<th>Min.</th>
<th>Max.</th>
<th>Unit</th>
<th>Dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td>LGI_CALLOUTS</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>255</td>
<td>Count</td>
<td>D</td>
</tr>
<tr>
<td>LGI_BRK_TERM</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Boolean</td>
<td>D</td>
</tr>
<tr>
<td><strong>LGI_BRK_DISUSER</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Boolean</td>
<td>D</td>
</tr>
<tr>
<td>LGI_PWD_TMO</td>
<td>30</td>
<td>30</td>
<td>0</td>
<td>255</td>
<td>Seconds</td>
<td>D</td>
</tr>
<tr>
<td>LGI_RETRY_LIM</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>255</td>
<td>Tries</td>
<td>D</td>
</tr>
<tr>
<td>LGI_RETRY_TMO</td>
<td>20</td>
<td>20</td>
<td>2</td>
<td>255</td>
<td>Seconds</td>
<td>D</td>
</tr>
<tr>
<td><strong>LGI_BRK_LIM</strong></td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>255</td>
<td>Failures</td>
<td>D</td>
</tr>
<tr>
<td>LGI_BRK_TMO</td>
<td>300</td>
<td>300</td>
<td>0</td>
<td>5184000</td>
<td>Seconds</td>
<td>D</td>
</tr>
<tr>
<td>LGI_HID_TIM</td>
<td>300</td>
<td>300</td>
<td>0</td>
<td>1261440000</td>
<td>Seconds</td>
<td>D</td>
</tr>
</tbody>
</table>
```
OpenVMS Break-in Detection Example

CLASS3$ SET HOST 0

Welcome to OpenVMS (TM) Alpha Operating System, Version V8.3

Username: PARSEC1
Password:
User authorization failure
Username: PARSEC1
Password:
User authorization failure
Username: PARSEC1
Password:
User authorization failure
%REM-S-END, control returned to node CLASS3::
OpenVMS Break-in Detection Example

CLASS3$ SHOW INTRUSION

<table>
<thead>
<tr>
<th>Intrusion</th>
<th>Type</th>
<th>Count</th>
<th>Expiration</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>NETWORK</td>
<td>SUSPECT</td>
<td>3</td>
<td>18-MAR-2008 18:11:51.17</td>
<td>CLASS3::MEHLHOP</td>
</tr>
</tbody>
</table>

CLASS3$ SET HOST 0

Welcome to OpenVMS (TM) Alpha Operating System, Version V8.3

Username: PARSEC1
Password:
User authorization failure
Username: PARSEC1
Password:
User authorization failure
Username: PARSEC1
Password:
User authorization failure
%REM-S-END, control returned to node CLASS3::
OpenVMS Break-in Detection Example

CLASS3$ SHOW INTRUSION

<table>
<thead>
<tr>
<th>Intrusion</th>
<th>Type</th>
<th>Count</th>
<th>Expiration</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>NETWORK</td>
<td>INTRUDER</td>
<td>6</td>
<td>18-MAR-2008 18:02:54.57</td>
<td>CLASS3::MEHLHOP</td>
</tr>
</tbody>
</table>

CLASS3$ SET HOST 0

Welcome to OpenVMS (TM) Alpha Operating System, Version V8.3

Username: PARSEC1
Password:
User authorization failure
Username: PARSEC1
Password:
User authorization failure
Username: PARSEC1
Password:
User authorization failure
%REM-S-END, control returned to node CLASS3::
# OpenVMS Break-in Detection Example

CLASS3$ **SHOW INTRUSION**

<table>
<thead>
<tr>
<th>Intrusion</th>
<th>Type</th>
<th>Count</th>
<th>Expiration</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>NETWORK</td>
<td>INTRUDER</td>
<td>9</td>
<td>18-MAR-2008 18:02:54.57</td>
<td>CLASS3::MEHLHOP</td>
</tr>
</tbody>
</table>

CLASS3$ **DEL/INTRUSION CLASS3::MEHLHOP**

CLASS3$ **SHOW INTRUSION**

%SHOW-F-NOINTRUDERS, no intrusion records match specification

CLASS3$
OpenVMS Break-in Detection Example

CLASS3$ SET HOST 0

Welcome to OpenVMS (TM) Alpha Operating System, Version V8.3

Username: PARSEC1
Password:

Welcome to OpenVMS (TM) Alpha Operating System, Version V8.3 on node CLASS3

Last interactive login on Tuesday, 18-MAR-2008 17:37:08.78

13 failures since last successful login

$
Security Auditing

• OpenVMS has the ability to audit nearly everything that happens on the system.
• The following are items that you can audit:
  
  - ACL Mount INSTALL Authorization
  - Time SYSGEN Identifier Connection
  - NCP Audit Persona Process
  - Breakin Login Logfailure Logout
  - Privilege Use FILE access All
• There are two types of auditing
  
  - Alarms - go to any terminal that has been enabled as the operator terminal; by default the console terminal
  - Audits - go to the audit server log file
Security Auditing

- **To view security auditing:**
  
  $ show audit

- **To enable security auditing:**
  
  $ set audit/audit/enable=item

- **To enable security alarms:**
  
  $ set audit/alarm/enable=item

- **To disable security auditing:**
  
  $ set audit/audit/disable=item

- **To disable security alarms:**
  
  $ set audit/alarm/disable=item
Security Auditing

$ show audit

System security alarms currently enabled for:
   ACL
   Authorization
   Audit: illformed
   Breakin: dialup, local, remote, network, detached
   Logfailure: batch, dialup, local, remote, network, subprocess, detached

System security audits currently enabled for:
   ACL
   Authorization
   Audit: illformed
   Breakin: dialup, local, remote, network, detached
   Login: batch, dialup, local, remote, network, subprocess, detached, server
   Logfailure: batch, dialup, local, remote, network, subprocess, detached, server
   Logout: batch, dialup, local, remote, network, subprocess, detached, server
Security Auditing

$ set audit/audit/enable=sysgen
$ set audit/alarm/enable=time
$ show audit
System security alarms currently enabled for:
  ACL
  Authorization
  Time
    Audit: illformed
    Breakin: dialup, local, remote, network, detached
    Logfailure: batch, dialup, local, remote, network, subprocess, detached
System security audits currently enabled for:
  ACL
  Authorization
  SYSGEN
    Audit: illformed
    Breakin: dialup, local, remote, network, detached
    Login: batch, dialup, local, remote, network, subprocess, detached, server
    Logfailure: batch, dialup, local, remote, network, subprocess, detached, server
    Logout: batch, dialup, local, remote, network, subprocess, detached, server
$
Security Auditing

- To generate Audit reports, issue:
  $ analyze/audit/qualifiers [file-spec]
- The default file-spec is the audit server log file
  SYS$MANAGER:SECURITY.AUDIT$JOURNAL
- The following are the qualifiers that can be specified:
  /BEFORE /BINARY /BRIEF /EVENT_TYPE
  /FULL /IGNORE /OUTPUT /INTERACTIVE
  /PAUSE /SELECT /SINCE /SUMMARY
Security Auditing – Example

```
$ ana/audit/since=1-jan-2008/summary sys$manager:security.audit$journal

Total records read: 2152248  Records selected: 52823
Record buffer size: 512
Successful logins: 3113  Object creates: 549
Successful logouts: 4975  Object accesses: 25152
Login failures: 102  Object deaccesses: 14209
Breakin attempts: 26  Object deletes: 659
System UAF changes: 12  Volume (dis)mounts: 1
Rights db changes: 2  System time changes: 9
Netproxy changes: 0  Server messages: 0
Audit changes: 47  Connections: 9
Installed db changes: 3  Process control audits: 787
Sysgen changes: 0  Privilege audits: 3113
NCP command lines: 30  Persona audits: 25
$```
Security Auditing - Example

$ ana/audit/since=1-mar-2008/event=authorization sys$manager:security.audit$journal

<table>
<thead>
<tr>
<th>Date / Time</th>
<th>Type</th>
<th>Subtype</th>
<th>Node</th>
<th>Username</th>
<th>ID</th>
<th>Term</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-MAR-2008 13:31:48.12</td>
<td>SYSUAF</td>
<td>SYSUAF_MODIFY</td>
<td>CLASS8 SYSTEM</td>
<td>24800427</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-MAR-2008 13:36:42.04</td>
<td>SYSUAF</td>
<td>SYSUAF_MODIFY</td>
<td>CLASS8 &lt;login&gt;</td>
<td>24800428 _TNA3:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-MAR-2008 13:37:12.19</td>
<td>SYSUAF</td>
<td>SYSUAF_MODIFY</td>
<td>CLASS8 SYSTEM</td>
<td>24800428 TNA3:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-MAR-2008 16:09:44.33</td>
<td>SYSUAF</td>
<td>SYSUAF_ADD</td>
<td>CLASS8 SYSTEM</td>
<td>25000446 TNA4:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-MAR-2008 16:09:44.37</td>
<td>RIGHTSDB</td>
<td>RDB_ADD_ID</td>
<td>CLASS8 SYSTEM</td>
<td>25000446 TNA4:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-MAR-2008 16:10:08.05</td>
<td>SYSUAF</td>
<td>SYSUAF_MODIFY</td>
<td>CLASS8 SYSTEM</td>
<td>25000446 TNA4:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-MAR-2008 16:46:55.99</td>
<td>SYSUAF</td>
<td>SYSUAF_ADD</td>
<td>CLASS8 SYSTEM</td>
<td>25000446 TNA4:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-MAR-2008 16:46:56.01</td>
<td>RIGHTSDB</td>
<td>RDB_ADD_ID</td>
<td>CLASS8 SYSTEM</td>
<td>25000446 TNA4:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-MAR-2008 16:46:56.34</td>
<td>SYSUAF</td>
<td>SYSUAF_MODIFY</td>
<td>CLASS8 SYSTEM</td>
<td>25000446 TNA10:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8-MAR-2008 14:33:04.81</td>
<td>SYSUAF</td>
<td>SYSUAF_MODIFY</td>
<td>CLASS8 SAUER</td>
<td>25E0046C RTA1:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11-MAR-2008 08:58:33.21</td>
<td>SYSUAF</td>
<td>SYSUAF_MODIFY</td>
<td>CLASS8 STUDENT207</td>
<td>25E004F8 RTA2:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21-MAR-2008 11:54:55.81</td>
<td>SYSUAF</td>
<td>SYSUAF_MODIFY</td>
<td>CLASS8 &lt;login&gt;</td>
<td>298006AF _TNA5:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Command >

End Of File for input reached.
Network and Internet consideration

• Minimize the use of username and passwords over network

For example, consider the access control string below:

```
$ copy/log xyz.dat alpha2"spencer foobar"::dka200:[foobar]
```

• In the above example the username and password would be sent in a packet over the network in plain text

• Someone looking over the shoulder of someone else typing from the command line can see the username, password and nodename
DECNET Proxy Example

CLASS2> mc authorize
UAF> add/proxy class8::sauer sauer/default
%UAF-I-NAFADDMMSG, proxy from CLASS8::SAUER to SAUER added
UAF> show/proxy class8::sauer

Default proxies are flagged with (D)

CLASS8::SAUER
  SAUER (D)
UAF> remove/proxy class8::sauer
%UAF-I-NAFREMMSGS, proxy from CLASS8::SAUER to * removed
UAF>
TCP/IP Proxy Example

$ set process/privilege=(sysprv,syslck)

$ tcpip
TCPPIP> add proxy williams/remote_user=williams /host=yahoo.parsec.com
TCPPIP> show proxy williams

<table>
<thead>
<tr>
<th>VMS User_name</th>
<th>Type</th>
<th>User_ID</th>
<th>Group_ID</th>
<th>Host_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>williams</td>
<td>CD</td>
<td>WILLIAMS</td>
<td></td>
<td>YAHOO.PARSEC.COM</td>
</tr>
</tbody>
</table>

TCPPIP> Exit

$ tcpip

TCPPIP> remove proxy williams

<table>
<thead>
<tr>
<th>VMS User_name</th>
<th>Type</th>
<th>User_ID</th>
<th>Group_ID</th>
<th>Host_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>williams</td>
<td>CD</td>
<td>WILLIAMS</td>
<td></td>
<td>YAHOO.PARSEC.COM</td>
</tr>
</tbody>
</table>

Remove? [N]: y

TCPPIP> Exit
Network and Internet Consideration

Hubs vs. Switches

• A hub essentially connects all the wires together

• Switches and routers are store and forward boxes

• Throw in network monitoring analyzers and

  ➢ When connected to hub all data is viewable
  ➢ When connected to a switch only the data on that system can be monitored
Encrypted Network Communication

- **Secure Shell (SSH)**
  - Protects the user’s data on network by encrypting it
  - Supported authentications include password, public key and host based

  - OpenVMS implementation of SSH server does not use the secondary password for user accounts
  - Keys are normally generated when SSH is initially configured
  - Enabled via an option in TCPIP$CONFIG.COM
Encrypted Network Communication

$ ssh system@class3.parsec.com !use system as a username not the current one
Host key not found from database.

Key fingerprint:
xizif-vobyc-sucep-myvac-kyhil-devas-kyzev-cumus-hysec-lyhen-fexyx

You can get a public key's fingerprint by running

$ ssh_keygen "-F" publickey.pub

on the keyfile.

Host key saved to ssh2/hostkeys/key_22_class3_parsec_com.pub

Host key for class3.parsec.com, accepted by williams Fri May 16 2008 19:33:55

system's password:
Authentication successful.

Welcome to OpenVMS (TM) Alpha Operating System, Version V8.3
 Last interactive login on Thursday, 10-APR-2008 15:07:34.16
 Last non-interactive login on Friday, 16-MAY-2008 11:52:22.56

$
Encrypted Network Communication

- Secure Shell (SSH)
  - Supports stunneling or secure tunneling
    - Provides encrypted communication for applications not designed for it
    - Tunnel set up when SSH connection is set up
    - Application communications to remote host through tunnel
    - Support included for passive mode FTP and X11 tunneling
Encrypted Network Communication

CLASS1> ssh system@class3.parsec.com -"R" ftp/2001:localhost:21
system's password:
Authentication successful.
Welcome to OpenVMS (TM) Alpha Operating System, Version V8.3
    Last interactive login on Friday, 16-MAY-2008 13:39:31.12
    Last non-interactive login on Friday, 16-MAY-2008 11:52:22.56
CLASS3> ftp localhost 2001
220 paul.parsec.com FTP Server (Version 5.7) Ready.
Connected to LOCALHOST.
Name (LOCALHOST:system): williams
331 Username williams requires a Password
Password:
230 User logged in.
FTP> passive on
Passive is ON.
FTP> ls x.*
227 Entering Passive Mode (127,0,0,1,192,26)
150 Opening data connection for x.* (127.0.0.1,49179)
x.bck;2
x.x;32
226 NLST Directory transfer complete
17 bytes received in 00:00:00.00 seconds (162.12 Mbytes/s)
FTP> quit
221 Goodbye.
CLASS3>
Encrypted Network Communication

Secure File Transfer (SFTP)

- Same communications protocol as SSH
- Is not as advanced as OpenVMS FTP
- Uses the same public and private keys used by SSH providing host authentications
- Enabled via the same option as SSH in TCPIP$CONFIG.COM
Kerberos

- Three headed dog that guarded the gate to Hades
- Created by MIT to provide strong authentication for client/server applications
- Configuration not covered in this session
- Overview of the three parts of Kerberos
Kerberos Support

- Kerberos Version 2.1 is based on MIT Kerberos V5
- Release 1.2.6, with CERT patches through 1.2.8

Operating System Support

- OpenVMS Industry Standard 64 V 8.2 or higher
- OpenVMS Alpha V 7.2-2 or higher
- OpenVMS VAX V 7.3
Kerberos

**TCP/IP Transport**

- hp TCP/IP Services for OpenVMS V 5.5 or higher (for Kerberos on I64 and Alpha V 8.2)
- hp TCP/IP Services for OpenVMS V 5.4 or higher (for Kerberos on Alpha V 7.3-2)
- hp TCP/IP Services for OpenVMS V 5.3 or higher (for Kerberos on VAX)
- If using third-party TCP/IP product such as Multinet or TCPware from Process Software Corporation, please contact them for support versions
Kerberos

First head of three headed dog represents the Kerberos server

- Key Distribution Center (KDC)
- Authentication Service (AS)
- Ticket Granting Service (TGS)

- The server contains all passwords associated with each principal and should be highly secured
Kerberos

- Second head of three headed dog represents the client

- Any entity that gets a service ticket for a Kerberos service

- Server must be configured as a client
  - Allows client utilities to be used to manage the server
Kerberos

- Third head of three headed dog represents the application server
  - Also known as Kerberized programs that clients communicate with using Kerberos tickets

  - OpenVMS currently provides a Kerberized version of Telnet
    - Once authentication has completed, all other communication is normal for the application

  - Tickets are time stamped to limit reuse
    - Because of the time limited value of the tickets, time must be synchronized on all systems involved
Kerberos

Kerberos Server

Authentication Service

Key Distribution Center

Ticket-Granting Service

Kerberos Client

Login Ticket

Username

TGT

Application

Server/App

Ticket

Application Server

Ticket
Secure Socket Layer

- Secure web browser (https://) uses SSL
- Based on OpenSSL 0.9.7d and includes latest security updates from OpenSSL.org
- Easily integrated into any application that wants secure implementation (at the programming level)

- Operating System
  - OpenVMS Industry Standard 64 V 8.2 or higher
  - OpenVMS Alpha V 7.3-2 or higher
  - OpenVMS VAX V 7.3
Question & Answer

Presented by Wayne Sauer

www.parsec.com

888-4-PARSEC
sauer@parsec.com