Global Information Assurance Certification Paper

Steven Bourdon
SANS Security Essentials GSEC Practical Assignment, Version 1.3

A Primer on OpenVMS (VMS) Security

Introduction

There does not appear to be any papers pertaining to VMS security. Whilst this OS certainly has a long history dating back to 1978[1], it certainly is alive and its usage is rapidly growing as opposed to the general misconception that VMS installations are dwindling towards the garage VAX enthusiast. This paper will introduce the reader to the history of the VMS operating system, describe and outline the security model concept that VMS uses as well as discuss several recent vulnerabilities affecting VMS.

This is not a practical step-by-step guide to securing VMS; rather, it is an introductory primer on the security concepts and features that the operating system has. VMS system security is quite comprehensive and there exists numerous parameters and settings which would require a much more granular depth than the length of this paper would provide. The Compaq OpenVMS Guide to System Security manual listed in the List of References Section is the recommended reference for further examining VMS system security settings and how the reader may go about tailoring the security settings of their system in conjunction with their particular organizational needs or security policy. My aim therefore is to provide insight into such functionalities and to provide an overview.

A Brief History of VMS

The VMS Operating System has been around since its initial version 1.0 non-beta release in 1978. Upon the release of version 5.5 in 1992, which included Posix support, Digital Electronics Corporation (DEC) subsequently changed to the name of their OS to OpenVMS- although many users still prefer to call it VMS[2]. During the 1980's and early 1990's the VAX/Alpha minicomputer running VMS was quite popular since VMS was a stable, reliable, and well-designed OS that was well suited to high speed, real-time applications. DEC was one of the first companies to attach terminals to their minicomputers so workers didn't have to do all their data-processing jobs at a central location[3].

In 1998 Compaq acquired DEC and its ensuing proprietary VAX/Alpha architecture which ran VMS. It was during this mid 1990's timeframe that DEC's revenues declined due to the increased penetration of desktop PC's and the shift towards an open-architecture platform where UNIX could run on. Because of this decline in sales during the 90's and the immense growth of UNIX platforms, particularly SUN and HP, it was widely believed that VMS was a dying OS

[1] "VMS Release History"
[2] Ibid.
[3] Silverman

destined to disappear. This, however, is turning out to be far from the truth.
DEC had a very strong and enthusiastic user's group, DECUS, the Digital Electronic Corporation's User's Society. DECUS users firmly, and rightfully, believed that the VMS OS had numerous qualities that set it apart from the various flavours of UNIX and the open-source distributions. In addition, Compaq has maintained a very strong commitment towards the continued development and support of the VMS platform.

The current result is that VMS is staging an extremely strong comeback. In fact, the marketing director of Compaq's OpenVMS Systems Group states that there are over 400,000 systems running OpenVMS, supporting over 10 million users[4]. Sample VMS customer sites include: numerous stock exchanges, Bank Austria, Government Securities Clearing Corporation (GSCC), International Securities Exchange, Hydro Quebec, and Northern Light[5]. Intel's fabrication plants rely on the use of VMS in the fabrication of their Pentium 4 and Merced class chips![6] In summary, VMS is an OS used in high-availability areas and is well worth examining.

How Secure is VMS?

Before anything else, I would like capture your attention by starting with a subjective indicator of how secure VMS really is. To date, the U.S Department of Energy's Computer Incident Advisory Capability (DOE-CIAC) lists a total of 22 VMS security advisories; of these, only 3 have been issued since November 1994. A search of CERT advisories with keywords VMS, VAX, or OpenVMS yielded a total of less than 25 unique advisories whereas Solaris yielded 266, Linux 313, and Windows 451 respectively.

More recently, several members from the Dallas-Fort Worth Compaq User's Group (DFWCUG), one of the most active local VMS user groups nationwide, entered a bastion host Alpha box running VMS into the 2001 DEFCON hackers Mecca. The VMS box was game to 5000+ hackers whose objective was to hack the various boxes entered in the contest in order to score points. According to Opcom's report of the events "The VMS machine on the Green team was configured with Apache web server. As we are aware, VMS is an extremely secure operating system. While many other boxes in the room, mostly Unix, Linux, and forms of windows, and even a Macintosh, were compromised and subsequently attended to by their masters, the VMS system remained intact". The article closes with "During the closing session at 4 PM Sunday, the Ghetto Hackers, which are the most respected and skillful, gave the Green team 'props' because our stuff stayed up and our 'root' was the only one they did not get. We consider this a positive note and a high compliment, coming from this well-accomplished group"[7].

In summary, VMS is an extremely secure operating system. Of equal importance, however, is the care and meticulous attention most VMS System Administrators pay towards the security management of their box or cluster of boxes.

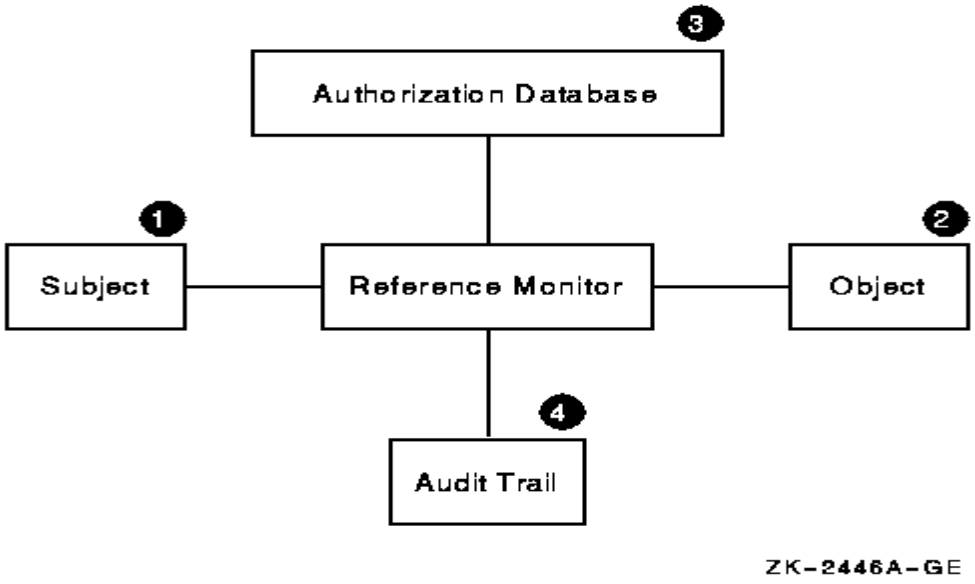
---

[4] Berlind
[5] "OpenVMS Success Stories"
[6] Magee
[7] Opcom

OpenVMS Security Model Concept

The VMS operating system implements a reference monitor concept that ties together the four essential components of system security. As you can see from the diagram below, subjects, such as users, would access a particular object and an authorization database would determine the type of access that the particular user would have. In addition, the audit trail has the capability to log the type of access requested and generate any defined security events or alarms. We will examine this model in more detail throughout the remaining part of this document. It is important to note that the reference monitor itself is not part of the kernel or is it a security-related subset. Rather, it is the components of the model that together implement this basic structure outlined below.[8]

Source: OpenVMS Guide to System Security p.2-2



1	Subjects	Active entities, such as user processes, that gain access to information on behalf of people. Users and user processes.
2	Objects	Passive repositories of information to be protected, such as directories, files, batch and printer queues, etc.
3	Authorization database	Repository for the security attributes of subjects and objects. From these attributes, the type of access (if any) is authorized. This would include the system user authorization files (SYSUAF), a rights database (RIGHTSLIST.DAT) as well as various network authorization databases (such as NET$PROXY.DAT)
4	Audit trail	Record of all security-relevant events, such as access attempts, successful or not. This would include logfiles, etc.

[8] OpenVMS Guide to System Security p2-3

1. Subjects

VMS has numerous system parameters, which allow the System Administrator to tailor how a user may login in to the system, what to do if failed login attempts occur, as well as what specific rights and privileges a user may possess. Listed below are the components one would encounter when attempting to login to a VMS system and the

a) Login Banner & Welcome Message

VMS has two system logicals, which allow you to define how you wish to display the login banner as well as the messages displayed after a user successfully logs in. The SYS$ANNOUNCE logical defines the pre-login announcement while SYS$WELCOME defines the information displayed after a user logs in. Text files should be customized for your site and these logicals should then be defined in the site-specific startup command procedure SYS$MANAGER:SYSTARTUP_VMS.COM.

```
$ DEFINE/SYSTEM SYS$ANNOUNCE "@SYS$MANAGER:ANNOUNCE.TXT"
```

Sample contents of an ANNOUNCE.TXT file:

```
********************************************************************************
                                    Company XYZ
                                    -----------

Use of Company XYZ computer and network facilities requires prior authorization.  Your use
of this system may be subject to security testing and monitoring.  Unauthorized use is
prohibited and abuse is subject to criminal prosecution.

********************************************************************************

Username:
```

Telnet, DECNet, and LAT services will all use the SYS$ANNOUNCE logical. If you are offering FTP server service then a third system logical, TCPIP$FTP_SERVER_ANNOUNCE, can also be defined and placed in the SYSTARTUP_VMS.COM file. Below are a sample definition and the subsequent announcement when initiating an FTP session on a server:

```
$ DEFINE/SYSTEM TCPIP$FTP_SERVER_ANNOUNCE -
      @SYS$MANAGER:FTP_ANNOUNCE.TXT"
```

b) Login Security and the Intrusion Database

VMS utilizes a logging mechanism which will automatically begin monitoring and logging the terminal or connection after the first unsuccessful login attempt and will continue to monitor for a certain period of time. This information is logged in an Intrusion Database that contains the timestamp as well as all pertinent information regarding the attempted login, such as the source (IP address, terminal, username) and login counters (how many attempted logins, when will entry be removed from database). There are numerous login parameters that define the behavior of the login monitoring. The parameter

LGI_BRK_LIM determines the number of failed login attempts the system permits before taking evasive action and LGI_PWD_TMO is a parameter that specifies the number of seconds for a login attempt to be made before deeming the attempt unsuccessful. These two parameters define when the Intrusion Database should classify the severity of the login failure as either Suspect or as an Intruder and take evasive action which could consist of refusing further connection attempts from the source or for a specific username as well as generating a security auditing message regarding the intrusion attempt.[9]

Sample Intrusion Database Display:

```
Intrusion     Type       Count     Expiration           Source
---------     ----       -----     ----------           ------
TERMINAL      INTRUDER       6   12-MAR-2002 15:00:29.53  ROMEO/UIC_003673001001: on TITAN

NETWORK       SUSPECT        5   12-MAR-2002 15:18:08.79  ROMEO::TELNET_0A0104EC
```

c) User Security Profile

Every user has a unique security profile. The security profile consists of three elements: a User Identification Code (UIC), Rights Identifiers, and Privileges.

User Identification Code (UIC)

The UIC is composed of a member number and a group number. An example would be that user SUSAN has a member number of 147 and a group number of 750- her UIC would then be (147,750).

Rights Identifiers

A System Administrator can create identifiers that can contain an individual user or groups of users. An example would be the identifier REPORTS which was created to allow several users, all from varying UIC groups, READ access to a specific directory. An identifier allows the Sys. Admin. The ability to create groups that are not dependent upon a user's group code and can therefore assign specific rights associated with that identifier, say READ and EXECUTE, to specific files or directories.

Privileges

There are approximately 40 types of privileges a specific user can possess. These privileges range from the ability to impersonate another user (IMPERSONATE) to the ability to mount tapes (OPER), or to send and receive mail via the network (NETMBX and TMPMBX). Privileges provide the System Administrator a mechanism to allow specific users the ability to perform tasks that they otherwise would not be able to do. An example would be assigning an auditor the READALL privilege that gives the auditor the ability to read all files on the system, even if file and directory protections would normally prevent her from being able to do so.

[9] "Controlling Breakins"

Sample User Authorization Record

```
Username: MHARTWELL                         Owner:  Mark Hartwell
Account:  DEVTEAM                           UIC:  [3447,101]([DEVTEAM,MHARTWELL])
CLI:      DCL                               Tables: DCLTABLES
Default:  DISK$STAFF:[MHARTWELL]
LGICMD:   SYS$COMMON:[SYSMGR]LOGIN.COM
Flags: RESTRICTED
Primary days:   Mon Tue Wed Thu Fri
Secondary days:                     Sat Sun
Network:  ##### Full access ######            ##### Full access ######
Batch:    ##### Full access ######            ##### Full access ######
Local:    -----  No access  ------            -----  No access  ------
Dialup:   -----  No access  ------            -----  No access  ------
Remote:   -----  No access  ------            -----  No access  ------
No access restrictions
Expiration: 31-MAY-2002 00:00    Pwdminimum:  6   Login Fails:     0
Pwdlifetime:         60 00:00    Pwdchange:  27-NOV-2001 08:24
Last Login: 27-DEC-2001 16:35 (interactive), 5-NOV-2001 20:35 (non-interactive)
Maxjobs:         0  Fillm:       100  Bytlm:        64000
Maxacctjobs:     0  Shrfillm:      0  Pbytlm:           0
Maxdetach:       0  BIOlm:       150  JTquota:       4096
Prclm:           8  DIOlm:       150  WSdef:         4000
Prio:            4  ASTlm:       250  WSquo:         4000
Queprio:         4  TQElm:        10  WSextent:     16384
CPU:        (none)  Enqlm:      2000  Pgflquo:      65535
Authorized Privileges:
  ACNT         ALLSPOOL     ALTPRI       GROUP        GRPNAM       GRPPRV
  NETMBX       OPER         READALL      SHARE        TMPMBX       UPGRADE
  WORLD
Default Privileges:
  ACNT         ALLSPOOL     ALTPRI       AUDIT        BUGCHK       BYPASS
  CMEXEC       CMKRNL       DIAGNOSE     DOWNGRADE    EXQUOTA      GROUP
  GRPNAM       GRPPRV       IMPERSONATE  IMPORT       LOG_IO       MOUNT
  NETMBX       OPER         PFNMAP       PHY_IO       PRMCEB       PRMGBL
  PRMMBX       PSWAPM       READALL      SECURITY     SETPRV       SHARE
  SHMEM        SYSGBL       SYSLCK       SYSNAM       SYSPRV       TMPMBX
  UPGRADE      VOLPRO       WORLD
```

2. Objects

There are 11 classes of VMS Objects. The most common ones are directories, files, batch or printer queues, and volumes (such as disks or tapes). Each class of object contains its own categories of access types. For example, a file would contain the READ, WRITE, EXECUTE, DELTE, and CONTROL access types whereas a printer queue would contain the READ, SUBMIT, DELETE, MANAGE, and CONTROL access types.

Every VMS object contains a security profile which determines who has what type of access to an object. An object's security profile consists of the owner of the object, the object protection code, and an access control list (ACL). Below is the order in which the system determines if, and what type of, access a user is granted.[10]

a) Access Control List (ACL)

If the object has any ACLs, the system will first examine the access control list and see if

[10] LeClerc

any specific identifiers are present which the specified user holds. If the user holds the specified identifier then they will be granted the type of access defined by that identifier.

Example: SUSAN from Accounting holds the identifier REPORTS. She attempts to modify the payroll report file that has an ACL that states that the identifier REPORTS can have READ access to the file. The system will deny her from modifying the payroll report and examine the Object Protection Code.

b) Object Protection Code

If the system could not find any ACL's for the user it will proceed to examine the object protection code. The protection code consists of 4 categories of users: SYSTEM, OWNER, GROUP, and WORLD. Each category of users can then, in turn, have a define type of access (READ, WRITE, EXECUTE, etc.). The GROUP category is based upon the group field of the owner of the object.

Example: SUSAN from Accounting is a member of the accounting group. FRED is the owner of the file and he is also a member of the accounting group. SUSAN attempts to modify the payroll report file using an editor. The system, after examining the access control list, examines the protection code and finds that members of the accounting group are permitted READ and WRITE access. She is therefore granted access to modify the file.

c) User Privileges

Finally, if the system could not find any ACLs for the user and the user is not permitted any access to the specified file after examining the object protection code, the privileges a user may possess are examined. If the user holds special privileges, such as the auditor with the READALL privilege, then they will be granted access to the object. Bypass (BYPASS), group (GRPPRV), and system (SYSPRV) are other privileges a user may possess which can raise a given users' ability to access a particular object. These privileges should therefore be tightly controlled and only granted with management approval or in accordance with your organization's security policy.

3. Authorization Database

The Authorization database is actually composed of numerous files. It is important to note that the owner of these files is the SYSTEM account and that discretion must be used in modifying the default protections for these files. Below is a list of the authorization files and their purpose:

SYSUAF.DAT

This is the system user authorization file. It contains the user information such as usernames, passwords, UICs, and user privileges. Because of the sensitive information contained in this file, VMS uses a function known as a Purdy polynomial where the username, password, and some salt are added together to create a one-way hash.[11]

[11] OpenVMS Wizard

RIGHTSLIST.DAT
This file contains the rights identifier database

NET$PROXY.DAT
This file contains the DECNet proxy account database

TCPIP$PROXY.DAT
This file contains the proxy account database for remote ("r") services as well and for client NFS accounts.

QMAN$MASTER.DAT
This file contains the master queue manager database that holds the security information for all batch and printer queues.

VMS$OBJECTS.DAT
This file contains the object database and their associated security profiles.

It is recommended in cluster environments that a common cluster-wide disk be used and that system logicals point to a single copy of these Authorization Database files in order to maintain a single security domain.

4. Audit Trail

Audit Server
VMS provides a robust set of auditing capabilities that permits the System Administrator to tailor the setup of auditable events as well as define the method in notifying of occurrences. The configuration of the VMS audit server is stored in the VMS$AUDIT_SERVER.DAT database file. Audit information can be generated as either events or alarms where auditing activity is either logged as an event to the security logfile, SECURITY.AUDIT$JOURNAL, as an alarm to an operator terminal or print device for immediate notification. In addition, the audit server can log security events to a remote node for archival and/or analysis. The security audit server can be configured to report on security-related activities in three ways[12].

a) User Authorization File
The audit flag can be set to allow auditing of events that are related to specific users. For example, an audit flag set for the user SUSAN would enable auditing of all activities related to the user process SUSAN.

b) Event Classes
There are 19 event classes that the audit server can be configured to audit. The classes range from object access, successful and unsuccessful login attempts, to

[12] OpenVMS Guide to System Security, Chapter 9.

the specific use of a privilege and changes of system parameters. By default, VMS will audit login failures, intrusion attempts (from the Intrusion Database), as well as any changes to the authorization database files (SYSUAF.DAT, NET$PROXY.DAT, etc.) as well as attempts to change the audit server configuration via the SET AUDIT command.

c) Access Control Entry (ACE)

A more selective and granular method of auditing activities is through the use of either the Audit ACE or Alarm ACE. An access control entry (ACE) is either an audit or alarm entry that is entered as part of an objects access control list. This is commonly done on sensitive files as well as important batch queues. For example, the ACL for a batch queue called BACKUP$BATCH may contain an audit or alarm ACE so that there is security notification whenever an attempt is made to modify the BACKUP$BATCH queue, such as hold, delete or submit backup jobs.

The reporting capabilities of the audit server give the System Administrator the ability to produce reports of varying breadth and depth. Listed below are several example reports from the audit server.

Sample Summary Report

It is important to note that a summary report is based upon the life of the audit journal file, security.audit$journal. If the System Administrator does not create a new version daily, which I highly recommend as standard practice, then a new version of the audit file is created each time the system is rebooted, which is not very often for VMS!

```
YIPPIE|MREARDON>analyze/audit/summary security.audit$journal

Total records read:       11757          Records selected:         11757
Record buffer size:         512
Successful logins:           23          Object creates:              33
Successful logouts:          17          Object accesses:           6455
Login failures:               1          Object deaccesses:         4836
Breakin attempts:           189          Object deletes:             109
System UAF changes:           0          Volume (dis)mounts:           2
Rights db changes:            0          System time changes:          0
Netproxy changes:             0          Server messages:              0
Audit changes:                1          Connections:                  0
Installed db changes:         0          Process control audits:      20
Sysgen changes:               0          Privilege audits:            71
NCP command lines:            0          Persona audits:               0
```

Sample Brief Report

```
YIPPIE|MREARDON> analyze/audit/brief security.audit$journal

      Date / Time         Type        Subtype       Node    Username      ID    Term
-------------------------------------------------------------------------------
28-FEB-2002 19:15:07.25 AUDIT      AUDIT_LOG_FIRST  ROMEO                00000000
28-FEB-2002 22:48:16.64 DELETE     OBJ_DELETE       ROMEO  STAFF_OPER    20206429 LTA5130
28-FEB-2002 22:48:16.64 DEACCESS   OBJ_DEACCESS     ROMEO                20206429
 1-MAR-2002 07:56:56.39 DELETE     OBJ_DELETE       ROMEO  STAFF_OPER    202056C9 TNA2463
 1-MAR-2002 07:56:56.39 CREATE     OBJ_CREATE       ROMEO  STAFF_OPER    202056C9 TNA2463
 1-MAR-2002 07:56:56.39 PRIVILEGE  PRVAUD_SUCCESS   ROMEO  STAFF_OPER    202056C9 TNA2463
 1-MAR-2002 07:56:56.39 DELETE     OBJ_DELETE       ROMEO  STAFF_OPER    202056C9 TNA2463
 1-MAR-2002 07:56:56.41 ACCESS     OBJ_ACCESS       ROMEO  STAFF_OPER    202056C9 TNA2463
 1-MAR-2002 07:56:56.41 ACCESS     OBJ_ACCESS       ROMEO  STAFF_OPER    202056C9 TNA2463
 1-MAR-2002 07:56:56.41 ACCESS     OBJ_ACCESS       ROMEO  STAFF_OPER    202056C9 TNA2463
 1-MAR-2002 07:56:56.41 DEACCESS   OBJ_DEACCESS     ROMEO                202056C9
 1-MAR-2002 07:56:56.43 LOGIN      REMOTE           ROMEO  STAFF_OPER    202056C9 Host:
10.1.16.166 Port: 1065
```

A Single Record from a Sample Full Report

```
YIPPIE|MREARDON> analyze/audit/full security.audit$journal -
                      /since=21-feb/output=audit_full_21FEB02.txt

Security audit (SECURITY) on YIPPIE, system id: 3073
Auditable event:          Object deletion
Event time:               21-FEB-2002 16:40:56.11
PID:                      2020EEBC
Process name:             _LTA5288:
Username:                 STAFF_OPER
Process owner:            [STAFF,STAFF_OPER]
Terminal name:            LTA5288
Image name:               DSA2:[SYS0.SYSCOMMON.][SYSEXE]LOGINOUT.EXE
Object class name:        LOGICAL_NAME_TABLE
Access requested:         DELETE
Logical name table name:  LNM$JOB_811D08C0
Parent name table name:   LNM$SYSTEM_DIRECTORY
Access mode:              KERNEL
Status:                   %SYSTEM-S-NORMAL, normal successful completion
```

Other Important Logfiles

In order to get a bigger picture it is often best to examine and compare the security journal along with other logfiles for suspicious activity. The names of several other important VMS logfiles as well as a description are listed below.

Accounting file

The accounting file ACCOUNTING.DAT contains resource utilization information. A key indicator of suspicious activity would be an observed change in the utilization of a resource, such as repetitive activation of an image (executable) file which is has never been reported before and unknown. VMS contains several accounting parameters that allow for the System Administrator to select the breadth and depth of accounting information reported.

Operator Log

The operator log OPERATOR.LOG is a file that records messages that are broadcast by the Operation Communication Manager (OPCOM). OPCOM can be configured to send messages to terminals as well as to the operator logfile. Like the accounting process, the OPCOM process can be configured to select which classes of information reported to an operator terminal and operator logfile. The classes of information reported include messages pertaining to devices, printers, network, cluster, as well as security events. The initial usage of the OPCOM tool was for users to contact an operator and request the mounting of a tape or other operator function. Today, OPCOM usage has grown to include the reporting of numerous system events and an operator terminal or logfile can be sources of valuable information when piecing together information in conjunction with the security audit journal.

Service specific logfiles

Various service logfiles are also of importance. Depending on the services that a VMS system may be running, logfiles could include:

- FTP– especially important if providing anonymous FTP service.

- SMTP – for monitoring incoming and outgoing mail traffic
- DHCP server
- NTP (Network Time Protocol) – for system time synchronization
- CSWS Compaq Secure Web Server logs

Vulnerabilities Worth Discussing

I remarked earlier that the number of vulnerabilities for VMS compared to other operating systems is remarkably small. I would, however, like to point out several known and/or recent vulnerabilities that are severe enough to justify examination and explanation.

a) Telnet/FTP traffic

A well-known and common vulnerability of VMS has been the usage of unencrypted Telnet and FTP traffic. The risk of packet sniffing can be quite real if Telnet and FTP services are offered over the Internet or if you are operating in an environment, such as academic, where your intranet traffic must be carefully managed. This is a very important risk! According to Compaq Support, TCP/IP version 5.2, which is available to government clients, has secure shell (SSH) support. Non-government clients will have to await the mid-2002 release of TCP/IP version 5.3 for potential support of SSH-- I have emailed the Compaq TCP/IP product manager and have yet to receive an official response regarding general public SSH support. I do know, however, that Compaq has noted the customer demand for secure Telnet and FTP. In the meantime, there exists an open-source SSH server for VMS that can be located at http://www.er6.eng.ohio-state.edu/~JONESD/ssh/DOC/. This SSH server is the only port available for VMS and it currently does not support SSH version 2 protocol. I would advise caution in implementing this in a production VMS environment without first extensively testing the stability and performance of it for your environment.

b) SNMP Vulnerability

In mid February 2002, news quickly became widespread regarding vulnerabilities that existed in the SNMP protocol. According to the CERT Advisory CA-2002-03 "Vulnerabilities in the decoding and subsequent processing of SNMP messages by both managers and agents may result in denial-of-service conditions, format string vulnerabilities, and buffer overflows. Some vulnerabilities do not require the SNMP message to use the correct SNMP community string."

According to Compaq's statement in response to this multi-vendor advisory, the SNMP agent for VMS TCP/IP is impacted. The SNMP image executes from a non-privileged account and is therefore not a risk for security compromise; however, Compaq further indicates that the SNMP vulnerabilities can cause the SNMP agent to terminate due to an access violation. Any further incoming SNMP requests will cause an automatic restart of the agent. Image patches for TCP/IP services are in the final stages of testing and Compaq states that they will release an ECO (Engineering Change Order) update to resolve the problem[13].

[13] "(SSRT0799) Potential Security Vulnerabilities in SNMP"

Compaq's Insight Manager software utilizes SNMP traps from management agents. There does exist a management agent for VMS. The agent can send traps into the Insight Manager framework or it can operate independently via a browser interface. This software should be used with caution for two reasons. While the first reason would certainly be the known vulnerabilities of VMS SNMP, the second reason arises from the non-secure method of browser-based management. The web agent for VMS does not encrypt communications and should at not be used to manage servers over the Internet. Compaq Engineering has also indicated that they are working on updating the Compaq OpenVMS Web-Enabled Management Agents with SSL support. In summary, I would highly advise against the use of SNMP based management over the Internet and restrict the usage inside the corporate intranet until an SNMP patch is released as well as until Compaq provides SSL support for the web-based VMS management agent.

c) DECWindows Motif

In October 2001, Compaq released a mandatory patch update for all VMS systems running the DECWindows Motif server. This mandatory update applies to both the Alpha and VAX platforms running any VMS version above 5.5-2[14]. A vulnerability was found where local users were able to gain access to unauthorized resources. Further information can be found at http://www.openvms.compaq.com/decw-mup-faq.html

Conclusion

This article will have hopefully given you a general overview of VMS system security. It is my sincere hope that individuals and organizations really take a look and consider the VMS platform for their environment. Numerous organizations are happily, and securely, using VMS as their main servers for important and vital services such as DNS, web, and data warehousing, to name a few.

I would also like to point out that Compaq offers a hobbyist program where anyone can obtain a copy of the VMS operating system as well as numerous layered products for $30, the cost of media and shipping. The hobbyist program is operated by Montagar Software Concepts and further information can be found at http://www.montagar.com/hobbyist/index.html. Interested hobbyists should take note that VMS requires an Alpha or VAX machine to run on, which can easily be found at auction sites such as Ebay- at least until 2004 when Compaq is scheduled to release the Intel Itanium porting of VMS ☺.

[14] "SSRT0738: OpenVMS Security Mandatory Update, OVMSUP03"

List of References

Berlind, David. "Compaq: VMS is alive, well – and kicking" Enterprise. 19 Oct. 2001. URL: http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2819065,00.html (21 Jan. 2002)

"CERT® Advisory CA-2002-03 Multiple Vulnerabilities in Many Implementations of Simple Network Management Protocol (SNMP)" CERT® Coordination Center. 12 Feb. 2002. URL: http://www.cert.org/advisories/CA-2002-03.html (10 Mar. 2002)

"CERT® Advisory CA-2002-09 Statistical Weaknesses in TCP/IP Initial Sequence Numbers" CERT® Coordination Center. 01 May 2001. URL: http://www.cert.org/advisories/CA-2001-09.html (10 Mar. 2002)

"Controlling Breakins" Computer Services, New Mexico Military Institute. 31 Aug 1994. URL: http://www.nmmi.cc.nm.us/~sidersb/SECURITY/breakins.html (27 Jan. 2002)

LeClerc, Rey. " DEC VAX/VMS Operating System Security Review" URL: http://www.auditnet.org/docs/decvaxvm.txt (12 Feb. 2002)

Magee, Mike. "Intel's dependent on the Alpha chip: Uses it to make Itanics" The Inquirer. 11 Jan. 2001. URL: http://www.theinquirer.net/01110105.htm (05 Mar. 2002)

McMillan, R. "A Practical Exercise in Securing an OpenVMS System", Proceedings from DECUS Symposium, 1993. URL: http://nsi.org/Library/Compsec/openvms.txt (12 Feb. 2002)

Opcom. "DefCon 9 – What I did on my summer vacation" Quadwords Volume XV, Number 7. Jul. 2001. URL: http://www.dfwcug.org/DFWCUG_newsletters/200107.pdf (06 Mar. 2002)

OpenVMS Guide to System Security. Houston: Compaq Computer Corporation, 1999. Part Number: AA-Q2HLD-TE. Also freely available online at URL: http://www.openvms.compaq.com:8000/73final/6346/6346pro.html (13 Mar. 2002)

"**OpenVMS Success Stories"** URL: http://www.openvms.compaq.com/success-stories.html (11 Mar. 2002)

OpenVMS Wizard, question #3039 "Password Hashing Algorithym? (Purdy)" 17 Sep.1999. URL: http://www.openvms.compaq.com/wizard/ (05 Mar. 2002)

Silverman, Dwight. "Compaq buys Digital, rises to industry's top 3." Houston Chronicle. 26 Jan. 1998. URL: http://www.chron.com/content/chronicle/page1/98/01/27/com.html (21 Jan. 2002)

"(SSRT0799) Potential Security Vulnerabilities in SNMP" Compaq Computer Corporation-Software Security Response Team. 18 Feb. 2002. URL: http://www.kb.cert.org/vuls/id/IAFY-

55KQYQ (05 Mar. 2002)

"SSRT0738: OpenVMS Security Mandatory Update, OVMSUP03" Compaq Computer Corporation-Software Security Response Team. 30 Oct. 2001. URL: http://online.securityfocus.com/advisories/3630 (11 Mar. 2002)

"VMS Release History" URL: http://www.openvms.compaq.com/openvms/os/openvms-release-history.html (12 Feb. 2002)