# What About VM Security?

VM Software, Inc. acknowledges the valuable assistance of Coopers & Lybrand in the development of this booklet. Coopers & Lybrand is one of the world's largest auditing, tax and management consulting firms with 14,000 people serving clients from 96 offices in the United States.



June 1988

User control. Accessibility. Interactivity. These are hallmarks of IBM®'s VM operating system. VM is unparalleled as a foundation for the multitudes of users and diverse applications that characterize the dynamic organization.

But can you have all of this and still keep your critical information resources protected?

Users are human beings. They make mistakes. They change positions and departments. They leave the organization. Along the way, some of them even sneak into each other's data. The downside of an active computing environment can be loss of control and serious security breaches, regardless of the operating system. The result is sleepless nights for the system staff responsible for maintaining and protecting critical corporate information resources.

Native VM has provisions for security, and in some cases these are enough. But in most organizations, added protection is needed to not only tighten security, but to streamline the often burdensome task of providing system administration for users.

VM is a different "animal" from other operating systems when it comes to security, so that adequate protection of the VM environment requires an approach based on an understanding of VM's unique design. Security measures do not need to interfere with the capabilities that make VM more flexible and user-oriented than the other IBM mainframe operating systems. The information on the following pages is an overview of the unique security features of the VM operating system, including the directory, logging on to the system, and sharing data. Special EDP auditing concerns are outlined. Conclusions center around options for protecting the VM-based system, with a focus on security system software.

Getting the most from VM begins with security measures that enhance the capabilities of VM and complement the needs of the organization.

## VM's Provision for Security

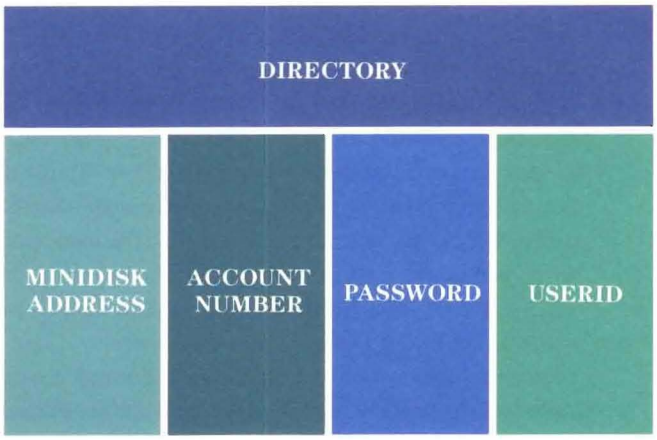
Data security is generally thought of in terms of preventing users from accessing, or modifying, the data of other users. Without data integrity, the information system, whether it is a personal computer or a sophisticated distributed environment, is of minimal benefit to users. Threats to data integrity, generally thought of as being maliciously oriented, may also be the result of simple error. It is the job of the security system to protect against threat to data integrity.

There are three major entities in the information processing environment: users, their data, and the resources that users rely on for storing and maintaining this data. In VM, there are also two types of relationships which govern the access to the data, ownership and access. Physical security, through pass keys and door locks, offers some protection over the hardware resources and the software contained on the machines. Through VM's system of passwords, users maintain ownership to the data associated with their assigned minidisk, and users may in turn grant access to other users as they see fit. VM, in its "vanilla" form, has made provisions for a level of protection of the major entities. But are they enough for every organization?

The spirit of VM is user control. It is inherently open and accessible to end users, providing a flexible environment for interactive productivity. Some VM installations find the security provisions adequate for their needs, particularly those with few if any users. Once there is user access, the security issues become much more complex. To ignore the security issues is to risk one of the most valuable assets of the organization.

## The Directory: Who's Who in the User Community

Virtual machine access, even if it is access to a user's own virtual machine, begins in the directory. It is here that userids, passwords, minidisk addresses, and other vital information about the users in the VM environment are stored. The directory is the place where security begins and, too often, ends. VM's Control Program (CP) controls the directory because it is through CP that the user is allowed access to the computer.




The CP directory, also more broadly termed the VM directory, serves as a repository for information including the definitions of users, their CP privilege classes (which determine the CP commands that they are allowed to use), and the location of their minidisks (where data is stored by users). A user is identified by a userid which may be up to 8 characters in length. Each userid in the directory also has a password associated with it.

A user may grant access to the data stored on his minidisk to another user. This is accomplished through the use of a minidisk password which is also stored in the directory. There are two types of data sharing in VM. The first is the directory link, which is used mainly for common applications like CMS. This is automatically executed when a user logs on, and is maintained until the user explicitly detaches this link. Directory links facilitate the process of gaining access to frequently used applications. For example, through a directory link an accounts payable clerk, when logging on, has automatic access to the accounting software package needed to perform this job function.

Users may also share data through user links. A user may explicitly link to another user's minidisk if the directory contains a link password for that user's minidisk. For example, a user may devise a link password, request that the security manager enter it in the directory, and then give this link password to other users so that they can access his data. This is a separate password from the user's logon password, but may ultimately accomplish the same purpose. Also, a password of ALL provides data sharing to any other user, by default. This is a reserved password, only to be used when granting access to all other users.

Without granting access to a minidisk through a link password, the "default" in VM is not to grant access at all. Passwords are defined in the same directory statement that defines the minidisk. As a result, at the same moment that the system programmer is setting up a user and defining the minidisk, he or she will indicate minidisk link passwords. Thus, without a password, there is no linking to the minidisks.

## Password Options

In VM there are three options that may be specified when designating a password. The three types of passwords are as follows:

**1. READ.** This password permits access to data on a minidisk in a read-only mode. Users may look at the information contained on this minidisk, but cannot modify it. Also, this reading may be accomplished at the same time by multiple users.

**2. WRITE.** A WRITE password permits users to both read the data on the minidisk and modify it. In other words, users with WRITE access may make changes to the data. With this type of password, however, only one user at a time may write to the minidisk; other users may only read the data until the individual user who is writing has completed this task.

**3. MULTIWRITE.** This type of password has the potential for creating a nightmare, and because of this is seldom granted. As the term implies, it permits access to data on a minidisk in a non-exclusive write mode. When accessed in a multiwrite mode, data can be modified concurrently by other users. MULTIWRITE is not supported for CMS use because of the potential for data destruction.

The logon password, which may also be up to 8 characters in length, is required to log on to a userid, thereby gaining access to the system. CP audits these passwords, monitoring the number of invalid passwords issued by a user when in the process of logging on to the system, or attempting to gain access (link) to the data of another user. The installation can specify the maximum number of invalid passwords that will be tolerated, after which CP will prevent the user from further attempts.

The password in effect validates the logon request, ensuring that the person requesting access is really the owner of the userid, and subsequently the one to whom this privilege was given. This is a classic area of security failure. Users frequently allow their personal passwords to be known to others, or display them in conspicuous places, and a security breakdown results.

The CP directory itself is stored in plain text format in a file located on a minidisk. The file containing the directory is customarily called USER DIRECT, with USER being the file name and DIRECT being the file type. This is essentially a "humongous" file, continuing for pages and pages with directory entries and any associated comments. There is an entry in the CP directory for every user in the system. The unfortunate systems programmer—generally referred to as the Security Manager—who is responsible for maintaining this directory has a major task in keeping this large file updated. The directory file is stored in plain text format and edited by a standard VM editor, such as XEDIT. With the continuous maintenance of new users, keeping the directory updated using XEDIT can be a very time-consuming process. Access to the directory file is protected by a LINK password that is associated with the minidisk on which the directory file is located.

Each user also has information in his directory entry which shows the location of his minidisk. Defining the minidisk address requires caution. It is necessary for a site to keep a "map" of minidisks as a way of tracking where they are located. Some organizations develop a program to help with this task, by extracting minidisk statements and sorting them to make the job of mapping large numbers of minidisks easier. System software products will also assist

with this process at varying levels. If the locations of minidisks are not tracked carefully, what can result is two or more users being given minidisks at the same address. This situation is referred to as overlapping minidisks. More about this later.

Privilege class is an important concept in VM. Privilege class controls at a high level which VM commands individual users can issue. With VM/SP Release 4, installations have the ability to create up to 32 CP privilege classes as well as to dynamically assign capabilities. This illustrates IBM's recognition of the increasing role that VM plays within organizations, and the complexity of user roles within this environment.

A user's directory entry will contain many more lines of information, depending on the kinds of applications and data that the user is able to access in the system. Each user is also given an 8-character account code.

When the directory is modified, such as when a new user is added, the file is saved. Following this, the source directory is placed online using a command called DIRECT. Running DIRECT is essentially indicating to CP that this is a "new" directory. CP then reads the file in, recompiling the directory and re-writing the machine-readable version out to a designated area on the system-owned minidisk (the system residence volume). With an online directory now in place, users are defined, privilege classes are set, and minidisks are defined.

When the user is officially part of the directory and able to log on, he or she is in charge of a dedicated virtual machine. From his terminal, the user has access to DASD storage, a virtual card reader, a printer, tape drives, and possibly communications lines.

## The Password Problem

Users access VM through the terminal, which is either connected directly to the system (local) or, in a Departmental Computing situation, through a network (remote). Because the terminal is the point at which the user enters into the VM environment, it is also at this point that security problems might begin. Two scenarios illustrate the security problems that may occur within VM.

A fairly sly systems programmer, even without the necessary CP privilege class, can gain access to the plain text directory on minidisk storage and can subsequently gain access to the virtual machines of the user population. From here it is possible to accomplish anything, including going into the payroll database and redressing all kinds of grievances, real or imagined.

A user might name his password after his dog, SPOT, and during lunch tell another user in the department the story of how he arrived at this password. If a person at the next table were to hear this information, he could log on after hours as the other user, using the SPOT password, and read or write on the data on that user's minidisk.

Passwords are contained in an easily readable form which can result in passwords, and subsequently data, being subject to unauthorized access. If an unauthorized user is able to obtain the password for the directory minidisk, this individual has access to a large amount of data because of having the passwords to other virtual machines and minidisks, including, for example, the owner of payroll information. He who can access the directory truly has the keys to the kingdom.

No matter how many locks there are on a door, if the keys are easy to steal, there is no security. VM, because it was originally positioned as a testing and time-sharing system, may have security "holes" where multiple users are concerned. People are human; the potential for security breaches is great. Users give their passwords to other users, or they tape the password to the side of the terminal, or they use easily guessable passwords, like the names of pets. Where multiple users are concerned, the security of any operating system is at risk.

Furthermore, the directory can only be maintained by the designated system programmer. With many users being maintained this can subject the system to the possibility of

human error, either in designating the minidisk address or in indicating the password. If a user's password is also given, either verbally or in writing, to the user's manager, confidentiality is totally lost. At least three people have the key at this point.

The password is the first major loophole in the security of VM. For an installation with few users, perhaps a scientific application, the security provided by VM may be more than adequate. However, the larger the VM population, the greater the opportunity for the potential weaknesses in the use of passwords to become security loopholes.

## Sharing Data

Even if logon passwords provide adequate protection, the security issue is further complicated once users start to share data. VM does provide a means of limiting data sharing to a subset of users, through link passwords at the minidisk level. It is important to make a distinction between minidisk and file at this point. Restrictions on data sharing between users go only as far as the minidisk level, so that once a user has given minidisk access to another user, that user can see everything on the minidisk. Users cannot protect individual files on the minidisk.

The link password is the means in VM of protecting the user against unwanted links, allowing the owner of the minidisk to attach a password to it. Again, because this password resides in the directory, the user may or may not have an easy way of setting the link password. It may involve coercing a systems programmer into going into the directory file, setting this password, and then recompiling the directory to reflect the change. Getting this accomplished in many organizations is the result of a complex series of work orders.

Assuming that the user is actually able to get a password set on his minidisk, sharing is now possible. In fact, as a means of being ready for any situation, the user may have had three passwords set on his minidisk: ALICE, a read access password, HENRY, the write access password, and FRANK, a multiwrite access password. With these three passwords, when another user attempts to link to this user's minidisk, the person will be prompted by CP to enter the appropriate password. The user with these passwords must have provided one of them to the user with whom he wishes

to share his data. If the other user cannot name the correct password, the link will not be allowed by CP.

Once the password is validated, the accessing user enters a CMS command called ACCESS. This provides access to the user's data, through CMS. Unlike CP, the minimal security built into CMS is easily circumvented. Thus, once access has been granted, a "hacker" can accomplish much more than the link-granting user intended to occur. An unauthorized write access can result in data that is altered or destroyed. This can be particularly unfortunate if the data is altered so slightly that the loss of integrity is not apparent until a much later time. Even an unauthorized read access can have major implications if sensitive data is involved.

It is also possible to gain access to minidisks owned by others through an implicit link. This type of link can be established between a userid and a software product, so that a user has automatic access to a software application at the time of logging on, without requiring a separate password. Though enhancing ease of use, implicit directory links can also be misused. A systems programmer with directory access can grant himself the privilege of sharing another user's data without using passwords. Once data sharing is granted with a directory, data can be accessed without password verification.

## Overlapping Minidisks

Getting into the data of others does not take a whole lot of investigative work. With overlapping minidisks, this process becomes even easier because it can be accomplished without the need for a password.

When users are defined in the directory, their minidisks are also defined by three parameters: the real DASD volume it resides on, the physical start address, and the length of the minidisk. As discussed previously, if this is not carefully mapped it is easy to assign the same minidisk space to two different users. In fact, if the mapping process is based on manually prepared lists, even if done carefully, there is opportunity for human error. There is nothing in native VM to prevent these overlapping minidisks from occurring. Furthermore, because of the ease with which this can be

accomplished, it is not uncommon for hackers to gain unauthorized access to data through defining an overlapping minidisk.

The results of an overlapping minidisk can be disastrous. Suppose a user identified by the userid TOM has a minidisk that contains nothing of significance. Another user, identified by SUPRHUSH, has a minidisk that contains confidential information. If TOM has access to the directory he could conceivably change the definition of his minidisk to correspond exactly with the definition of the minidisk belonging to SUPRHUSH. For example, SUPRHUSH might reside on a DASD identified by VMPK51, starting at cylinder 281 and continuing for three cylinders. All TOM has to do is define a minidisk for himself with the same parameters. TOM would thus gain access to the "crown jewels." Though each user owns a separate virtual machine, and a different logical means of accessing the minidisk, the results are unfortunately the same.

The only situation in which overlapping minidisks are really necessary is for maintenance purposes. It is quite common for systems programmers to have deliberately overlapping minidisks for maintaining the system. For example, one of these traditional userids is MAINT, which typically owns a series of minidisks covering full DASDs, from beginning to end. By using MAINT, the systems programmer could then go in and perform maintenance on individual user's minidisks so that this space is better utilized. However, this also means that anyone having access to MAINT can subsequently log on and have access to the physical location of all other users.

VM has no native protection against defining these overlapping minidisks. Not only do they offer the same access as that afforded by a link, but use of the link command generates a CP accounting record which could later result in discovery. Without a very time-consuming audit, overlapping minidisks can continue on indefinitely. The most reliable protection against this situation is through system software.

The VM environment allows each user to play operator. This is one of the major strengths of VM and should not be compromised. However, this capability must be managed, with protection that does not restrict the availability of resources.

## Making VM Secure

Based on the potential security loopholes in the VM operating system, most installations find that it is necessary to "shore up" this situation above what is offered through native VM. This might include "homegrown" solutions that are developed in-house, as well as system software. Considerations for enhancing security include password protection and access rules.

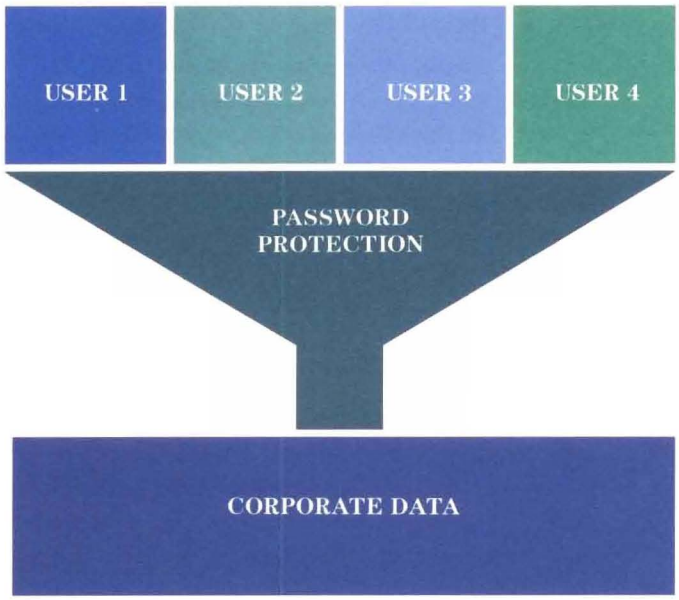
## Expanding Password Protection

Much of password protection is based on persuading, or coercing, users into being more responsible for their passwords. If users continue to tape passwords onto the sides of their terminals, or otherwise ignore the security implications, then of course security is a lost cause no matter what protection mechanisms are put into place. However, there are automated methods which, though not absolving the organization of responsibility in developing security policies, at least assist in getting the users more involved.

Here are the major password management needs:

1. Automatic expiration of LOGON passwords.
2. Enforcement of installation standards.
3. Providing users with the ability to change passwords.

If passwords stay in effect for months and even years at a time, their integrity will most likely decline with time. For example, if a user has given his password to another, the chances that this password has in turn been given to someone else, and even passed on further, multiplies. But with automatic expiration of passwords, once the password expires, the chain is broken. Password status needs to be constantly reassessed, both by the security manager and by users. To become complacent about passwords is to allow the security cracks to widen.




Automatic password expiration is a means both of forcing attention on the critical importance of password status, and stopping any chains that may have developed through users who have given their passwords to others. For example, a security system can provide the option of having users' passwords expire every three months, thus forcing each user to obtain a new password. This is more likely to work if the user is informed a few days ahead of time that the password is about to expire, and then provided with a full screen menu to assist with changing the password, as well as online help in doing so.

Password change can be enforced by preventing further use of the system until the password has been successfully changed. Merely reminding the user that the password should be changed, without backing this up with consequences, will most likely not result in action.

Password expiration implies that there is an overall organizational password policy. This includes not only the intervals at which passwords will expire, but any patterns to which passwords must adhere. For example, the possibility of guessing another user's password is made more difficult if users are required to have at least four characters in their

passwords, or if they are not allowed to include various symbols. Through security software, these standards can be enforced automatically without requiring constant monitoring by the security manager.

Depending on the needs of the organization, the password policy can go even a step further, past the issue of access control. Password information can be encrypted in the directory, making this information impossible to read. Thus, if the directory is the object of unauthorized access, the passwords cannot be read. The logon process involves reading the passwords on the employee's record, and then comparing the password entered to the one on the record before access is allowed. With this policy, it is particularly important that passwords be encrypted, because the password on the employee record would then be visible to others during an online query. Through security software, the password could be encrypted in both places.

Password encryption is generally based on the National Bureau of Standard's Data Encryption Standard (DES). Encryption can be very expensive, resulting in increased system overhead due to the process of encrypting and decoding. Also, data itself can be encrypted, especially if it is sensitive in nature. This adds even further system overhead. The costs must be weighed against the need.

A key to the success of a password expiration system is that the user needs to be involved both in choosing the password as well as making the actual change. Randomly generated passwords are difficult to memorize and irrelevant to users. Even is a user chooses the name of his dog as a password, after 90 days he will be forced to think of something else. When users have some level of control over what affects them directly, cooperation is enhanced. They are also more likely to make these changes if they can do so without having to go through a lengthy process of filling out forms and having to get special permission.

In addition, allowing users to change passwords creates a constant awareness of passwords and, by inference, system security. Rather than having an anonymous system with passwords assigned by a central technician, someone the users might not know, this system places accountability for system security more directly on those most responsible for

it. Some VM shops do, however, assign passwords from a dictionary rather than allowing users to choose them. It is a matter of organizational policy.

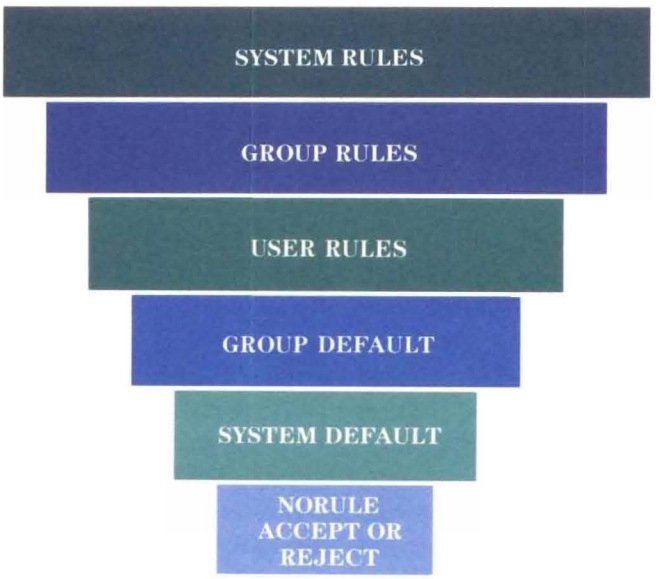
System security can also be enhanced through the development of user interfaces which, based on the user's userid, present a screen after successful logon which lists the applications that the individual is qualified to use. This interface is further enhanced if forward and backward scrolling, and selection by PF key or application name, are allowed. A caution lies in the listing of applications. Including those for which the user is not qualified may result in encouraging attempts at penetration.

## Access Rules

System security software packages that "hook" directly in CP allow the establishment of access rules which in effect supplant the need for minidisk passwords. Because of the cumbersomeness involved in indicating these passwords for each user's minidisk, as well as the potential security loopholes that these passwords create, the ability to establish rules saves time and enhances security.

The basic concept of rules in security management involves the establishment of rules at various levels, from the overall organizational level down to the user level, which are first evaluated by CP before an access request is granted. Rules allow for the control of access at organizational levels, thus ensuring that access to information reflects both organizational structure as well as the needs of users. The trend in security packages is toward access control by rules rather than by password. This is true in part because users are careless about minidisk passwords, and because rules are both easier to use and understand and much more flexible.

System software packages with a rules facility generally allow the system administrator to establish groups of users, perhaps based on departments, and then to set up access rights based on users being inside or outside the group of the minidisk resource owner. Users are also allowed to indicate to whom they wish to grant access privileges. This implies a hierarchy, as indicated below.




At the top of the hierarchy are system level rules, which override the rules at lower levels in the hierarchy, taking precedence over any other access privilege in the system. For example, a product designed to perform system backup would be allowed to link to the minidisk of every user in the system, thus allowing all data to be backed up. This rule would be set at the system level as an overriding rule, to ensure that even if a user has refused others access to his data, his minidisk would still be backed up.

Rules based on a hierarchy can then be set all the way down through the organization. For example, users in a specific department might be linked from certain applications. In addition, a user can also allow another user to link over to his minidisk, without having to designate and then divulge a minidisk password. A user might grant access to another user through this rule:

```
ACCEPT JIM LINK 191 RR
```

This user has allowed user JIM to access his 191 minidisk (where his data is contained) in read-only mode. This rule would have been entered through the system software package. Referring back to the earlier link example, rather than using, for example, a read-sharing password of ALICE and telling only the "right" people of this password, with the hope that they don't write it in a conspicuous place, rules allow access to be designated before the fact. There is no password involved, so there is no worry concerning to whom JIM might divulge a password.

Rules, including those written by users, are contained in a database which is maintained by the security system software. With the rules facility in place, CP intercepts certain access-granting CP commands, and passes the command to the system software where the command is evaluated in the rules database to see if there is a rule governing this request. The software then sends a message back to CP to indicate whether the request should be accepted or rejected.

At the bottom of the rules hierarchy are defaults. If a specific request has been evaluated down through the hierarchy and no rule has been indicated to govern whether or not it should be granted, it is passed down further to see if there is a default for the group level. At the group default level, if there is no rule governing the request, it is then passed on to the system default level to check that there is a system level default. These defaults do not imply an absence of rules, only that if a rule has not been specified at a higher level in the hierarchy, then rules have been specified at these lower levels to handle the request. Thus, the default can be either to grant or reject the request.

At the lowest level of the rules hierarchy is the system default level, which evaluates the request if no rule has been written at any of the previous levels. At this point, the VM system is either open or closed.

An open system is one in which, if no rule has been found, CP is allowed to accept the request. Conversely, a closed security system is one in which if no rule has been found to govern a request, the request is rejected. Organizations often find that it is safest to be a closed system, though in the early stages it may be necessary to be open, until after all the analysis is complete and the trial period over. After all eventualities and possibilities have been thoroughly analyzed, it may now be feasible to become a closed security system. The implications of being open, particularly with the creativity inherent in experienced users, are legion.

## How Do Rules Work?

The rules facility is provided by system software which hooks in the CP directory. These rules can be established because of the Access Control Interface (ACI) in VM. The ACI intercepts CP commands that include AUTOLOG, LOGON, LINK, SPOOL, TAG, and TRANSFER. ACI is essentially a package in which a system service called *RPI communicates with a userid by the Inter-User Communication Vehicle (IUCV). IUCV provides a means for two cooperating virtual machines to talk with each other—not the two humans running these machines—but the programs running on these machines. For example, a program running on one virtual machine can "shoulder tap" a program on another virtual machine, obtain needed information, and then return to what it was doing. IUCV also communicates with CP through system services, such as *RPI.

To make use of this facility, a userid is specifically authorized in the directory to connect to the *RPI system service. Once an authorized userid connects to *RPI, every time one of the included CP commands, such as LINK, is

issued by anyone else in the system, CP intercepts it and passes it back to that user's virtual machine. The request is then evaluated based on the rules structure and passed back with a reject/accept decision. CP is not actually aware of what checks are being made or the basis for the decision. A return code is sent back to CP to indicate the decision. ACI was actually developed for a security system software product marketed by IBM, but this facility has also been adapted for use by other software vendors.

Security system software that hooks into CP through the Access Control Interface is able to accomplish command verification without system modifications. There are three modules in native VM which are essentially "dummies" that don't do anything. As supplied, they are such that if accidentally called by a program there is no resulting error message, but nothing happens. These modules—labelled DMKRPI, DMKRPW, and DMKRPD—are replaced by system software offering rules capabilities with three modules of the same names. These modules that replace the dummy versions utilize *RPI and provide the capability to designate rules. Again, without a security product, these modules essentially just sit. Because of the nature of the ACI modules, replacing them with the vendor's versions is not considered an operating system modification. There are no negative consequences to be expected from this additional layer of checking.

The rules facility provided by system software not only provides more comprehensive protection without passwords, but is also much easier to use and manage.

## Special Auditing Considerations

The role of the EDP auditor is to ensure that the procedures for safeguarding the security of an organization's information systems are effective and efficient. Information is a valuable asset on which management relies for strategic decision-making. The EDP auditor must ensure that existing security measures meet management expectations.

Security methods for information systems should address accessing both the computer itself as well as the software, including data, in the computer. Physical access controls include door locks, terminal locks, and computer room visitor logs. These controls are easily tested by the EDP auditor.

Logical access controls are those security measures that exist "below the surface." They govern access to the programs and information stored in the computer. Comprehensive logical access security must provide what auditors refer to as preventive pre-processing and timely post-processing controls.

Preventive access control features typically found in a secure information system include, but are not limited to:

- Security administration
- Limited unsuccessful logon attempts
- Rules governing access to data
- Enforced periodic password maintenance
- No sharing of passwords and userids
- Automatic expiration of passwords
- User capability to maintain passwords
- Encryption of stored passwords
- Masking of password entry.

Post-processing controls include:

- Reviewing logs of access attempts to significant minidisks and restricted activities
- Confirming that all jobs executed were authorized
- Testing modified software.

The EDP auditor's scope may include the entire access control environment. He or she will confirm that the logical access security over the information system is adequate. Typically, the EDP auditor will review the security mechanisms in place, the password maintenance policies, and the appropriateness of access granted to significant data.

Testing procedures performed by the EDP auditor are quite often customized to the client's data processing environment. An all-inclusive list is impossible. Some examples of tests of preventive controls include:

- Confirming that a selected sample of computer user accounts were authorized
- Verifying that a sample of rules controlling access to data are appropriate and authorized
- Testing the security of password data files.

Examples of post-processing reviews might include:

- Minidisk accesses for authorization
- Programs executed for authorization
- Abend logs for proper supervisory review.

Access controls provided as part of the VM operating system have been discussed elsewhere in this booklet. The use of the VM operating system in a production processing environment which allows many users to share data and program files introduces additional security considerations.

The EDP auditor will generally focus on both preventive and post-processing controls when examining the security system. Understanding and anticipating these concerns, and communicating with the auditor, will streamline the review process. The security audit is yet another step in ensuring the security of critical information resources.

## Security System Software

The basic rule of thumb in considering any kind of security solutions for VM, whether developed in-house or purchased, is that they should protect the system while not discouraging users from subsequently using the system. VM is a user's system, and security should not interfere with the feeling of control and free access to information. By implication, system security functions should be as transparent as possible to the user, so that the security job is performed without interfering with the user's job.

Security software also allows an organization to further refine the access controls needed to make sure that system use is conforming to overall organizational standards. Software allows the ability to prevent users from logging onto the system on the basis of factors such as physical location, time of day, and the nature of the applications generated. In effect, software can enforce the rules and policies of the organization.

Based on the unique environment provided by VM, the major considerations for enhancing the security offered by native VM can be summarized as follows:

- Terminal and userid protection
- Hacker protection
- Last logon information
- No system modifications required
- Open or closed protection allowed
- Availability monitored by CP.

Password protection affects not only the overall security of the system from the organizational point of view, but also whether users themselves feel comfortable in the environment. Comfort is the result of both ease of use and overall trust. Passwords, assuming they are encrypted in the directory, should only be available to the security administrator, if anyone. Some packages go as far as to limit knowledge of the password to the user himself, with managers and the security administrator denied this knowledge. This reinforces the feeling of control that is important in the VM environment while placing responsibility for maintaining the password on the user.

Passwords can be further protected from hackers through terminating the access environment after a specified number of incorrect attempts at entering the password. Thus, if someone tried to log on as another user by guessing the password, the ability to guess would be terminated after, for example, three password attempts. This frustrates the hacker who may have compiled a repertoire of possible combinations and has been limited only by his own patience. This is an important consideration particularly for security at the departmental processor level, where access to facilities is even harder to monitor. Both violations and approved access to protected data need to be documented. These attempts can be recorded and maintained in a special file for later review by the security administrator.

System software with user exits further enhances security by allowing organizations to build in a means of ensuring that unique standards and policies are met by users. An example of user exit functions includes the validation of a new account number before it is updated in the directory, or the validation of a password to make sure that it contains a certain number of characters and ends in a certain letter. A user exit is basically a place in the program where it branches out, the information that has been entered is checked, and then the program continues on. For

example, when a password is being created, once it has been entered, the software program can branch to a user exit where the password is checked and, if standards are met, the program can continue. If the password does not correctly meet the standards, a message can be sent back telling the user to try again. Thus, user exits serve the dual purpose of enforcing standards as well as guiding the user in performing the task.

File level security continues to be an issue in security management. Security at the minidisk level offers very good protection. However, if a user gains access to another user's minidisk, he has access to all of the files contained on the minidisk. With the CMS file system, there is really no way to adequately and totally protect an individual file. With extensive modifications to CMS itself, some security system software vendors advertise this level of security. For a general user, the implications of the CMS modifications aside, this means of providing file level security may work. Experienced users, however, can easily implement a different version of CMS and bypass the system-defined protections completely.

## Rounding Out the Plan

Throughout these pages, the importance of VM-based security solutions has been stressed. As critical as the technical solutions are, in and of themselves, technical answers do not guarantee success. Behind those terminals are human beings, and ignoring the people considerations can leave the system as unprotected as it was before security was implemented.

When the security system is being introduced, or the current one changed, it is important that this be accomplished with as little shock to the users as possible. The system needs to remain operational, with security introduced gradually.

It is not uncommon for users to resist the implementation of security. Logon messages change. Password formats change. Users may be forced to update passwords more often. When users are not given enough advance preparation, with adequate training, the new system will be both confusing and frustrating. This leads to a lack of cooperation that may usurp any progress that is made as well as a loss of productivity while users attempt to maneuver through the perceived obstacles. Again, users are human. Technical innovation must be balanced by attention to the people considerations. Phasing-in security controls ensures that users are given ample time to adjust to these new responsibilities.

The right system software will provide the vehicle needed, a vehicle for a gradual phasing-in of security, so that controls do not have to go into effect overnight. For example, the directory can be automated first, followed by changes in the logon procedure, followed by more password controls. This approach gives the user community time to adjust to increased security, with system software features providing even more controls as necessary. Comprehensive system software, designed exclusively for the VM operating system, will provide the foundation needed for security that enhances the capabilities of VM and is tailorable to the needs of the organization, without adding an extra burden on the technical staff.

System security is everyone's responsibility. A solid security plan that is fortified by system software will ensure data integrity for the organization and accessibility for users.

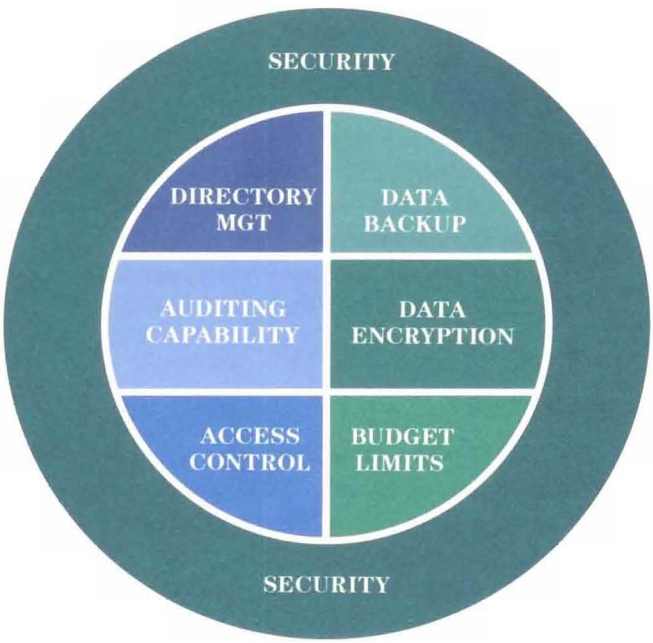
## A Comprehensive Approach

The system security needs in organizations that serve large numbers of interactive users are legion, extending past the issue of system access. And that's where VMCENTER II enters the picture.

VMCENTER II is a comprehensive answer to the systems management needs of VM environments. In one easily installed package, it streamlines user access to system resources while automating a wide variety of data center management functions. VMCENTER II fortifies the VM environment with capabilities that include:

- Controlling system access, through site-defined rules
- Controlling access to data that is stored both online and on tape
- Directory management, with flexible implementation
- Password management, including automatic expiration and encryption
- Disk space allocation
- Auditing capabilities, with summary and exception reports
- Encryption of CMS files
- Full and incremental backup.

VMCENTER II works several ways to improve control over the VM environment. It automates a wide range of functions that would otherwise have to be performed manually. It coordinates related management functions such as scheduling, monitoring, and accounting. And VMCENTER II simplifies administrative procedures to a point where users can be made responsible for much of their own housekeeping.




The result is savings of both people and computer resources.

The security capabilities of VMCENTER II are fully integrated with the other VMCENTER II features, including DASD management, operations management, performance and capacity management, system monitoring, batch operations control, system accounting, and recovery management.

Don't jeopardize your systems management strategy. Put automated controls into place with VMCENTER II. It is the perfect partner for the VM operating system, for the data center staff, and for the entire organization.

VM Software, Inc., develops, markets, and supports software for the VM operating system. VM Software's team of experienced VM experts has earned a worldwide reputation of superior product design, customer service, and documentation. The company combines a strong record of innovation with a continuing commitment to provide the enhancements necessary to meet the future challenges of VM.

VM Software products are sold through agents and subsidiaries in over 50 countries worldwide.

## Our Products

**VMCENTER II**—a comprehensive Systems Management package offering security, DASD management, performance monitoring and capacity planning, production control, and operations management.

**VMACCOUNT**—an accounting, reporting, and chargeback system, with continuous collection and real-time validation of data.

**VMARCHIVE**—an end-user archive storage system for online and offline storage of CMS files.

**VMBACKUP**—an automated DASD backup and restore facility using full and incremental dumps.

**VMBATCH**—a full-screen batch processing system that reduces intervention by the data center staff and also increases end users' productivity by allowing them to run multiple jobs.

**VMMONITOR**—a monitoring system that runs in real time, producing online graphic displays, exception reports, resource evaluations, and trend analyses to solve performance problems by automatically tuning the system.

**VMOPERATOR**—an operator display management system providing message filtering and routing, scrolled console displays, multiple consoles, and online system log review.

**VMSCHEDULE**—a work scheduling system that optimizes use of computer resources by allowing tasks to be scheduled to run unattended during off-peak hours.

**VMSECURE**—a resource access control, directory management, and disk space management system, offering site-defined rules for system access.

**VMSORT**—a utility program to rearrange or merge data into a user-specified sequence.
**VMSPOOL**—a spool management system that monitors and reports on spool conditions, accounts for spool usage, and provides a flexible display of selected spool information.
**VMTAPE**—a tape drive and volume management system that substantially reduces operator tasks while offering complete control.
**VMSQL/EDIT**—a full function table editor for IBM's Structured Query Language/Data System (SQL/DS).
**VMSQL/REPORT**—a sophisticated report writer for SQL/DS that lets programmers build simple and complex reports quickly and easily.
**VMSQL/REORG**—a tool to help the Database Administrator reorganize and maintain SQL/DS databases.

## Headquarters

VM Software, Inc.
1800 Alexander Bell Drive
Reston, VA 22091
(703) 264-8000

## Subsidiaries

VM Software (UK) Ltd.
Phoenix House
1 Station Hill
Reading RGI 1NB
Berks., United Kingdom
44-(0)734-509001

VM Software Netherlands
The Hague Business Centre
Parkweg 2
2585JJ The Hague
Netherlands
31-(0)70-524119

VM Software GmbH
Falkensteiner Str. 75-77
D-6000 Frankfurt 1
West Germany
49-(0)69-590456

VM Software S.a.r.l.
27 Rue Garnier
92200 Neuilly Sur Seine
France
33-(0)1-40883575


VM Software, Inc.
1800 Alexander Bell Drive
Reston, Virginia 22091
(800) 562-7160 (703) 264-8000