RC 12403 (#55644) 12/18/86
Computer Science 10 pages

Security Considerations for Dedicated Public Terminals

Steve R. White

IBM Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

87A000570

Abstract

Dedicated public terminals are computer terminals, available in public places, whose function is to provide access to a set of information utilities. This paper examines dedicated public terminals which access applications running under VM/CMS on IBM mainframes. On-line library catalog terminals are an example of such terminals. Security problems associated with this class of terminals are summarized. Various attacks are detailed, including novel attacks that arise from the use of personal computers as dedicated public terminals. Alternatives for improving the security of these terminals are examined. The author recommends that these terminals be attached to mainframe applications only over dialed lines. This isolates the user from the VM session console, which eliminates most of the security problems. If the dialed line is dedicated, and the application account is autologged, the application is automatically available whenever the host computer is functional.

Dedicated Public Terminals

The use of computer terminals as a means of accessing information within companies is growing. With this growth, there is an increased use of terminals, available in "public" locations, whose function is dedicated to providing access to a particular information utility. On-line library catalogs are an example of these information utilities. This paper examines the security problems associated with dedicated public terminals attached to IBM's VM/CMS operating system.

Often, the utilities which are accessed through these terminals are developed internally by the company in which they are used. They may be available from individual users' accounts as private utilities as well as being publicly available. In many cases, these utilities are developed as applications that run in a VM/CMS console session. This is a natural way to write such a utility, especially if it is used on individual VM accounts.

Exposures of Dedicated Public Terminals

Despite their growing utility, typical implementations of these dedicated public terminals have numerous security exposures. These can be divided into two classes: denial of service, and unauthorized use of a VM account.

Denial of Service

As computerized information utilities replace more traditional forms of making information available, reliance on the computing system grows. If a physical card catalog in a library is supplanted by an on-line catalog utility, for instance, the unavailability of this utility creates a severe problem for the users of the library. So, it is often very important that information accessed by dedicated public terminals be highly available. Security weaknesses in the terminal system may result in the utility becoming unavailable. This can be the result of accidental errors by legitimate users, who are unaware of the frailty of the terminal system, or it can result from malicious activity.

Unauthorized Use of a VM Account

More severe than denial of service is the ability of an unauthorized user to access the facilities of the VM account on which the information utility executes. If the VM session console can be accessed by an unauthorized user, the security of the host is compromised. The attacker can evade the password protection that usually protects the account, and can use the facilities of the host as if he or she were the legitimate owner of the account. This raises the following threats.

Direct Access to the Account's Information

Clearly, an attacker with access to the account has access to all of the files on that account. Information, which may be proprietary, may be accessed. Alternatively, it may be altered or destroyed.

Ability to Masquerade as the Account's Owner

An unauthorized user also has access to the privileges associated with the account, with complete anonymity. The attacker can send electronic mail, or data files, as if he or she were the account's owner. This ability to disseminate information with a false identity provides a number of opportunities for further subversion, especially in the presence of large networks of computers.

Many software repositories give almost every account the ability to contribute software. PCTOOLS and VMTOOLS are IBM internal software repositories which provide examples of such "public" repositories. The attacker may put software onto these repositories. This is particularly easy when the public terminal is a PC. The software may be developed at another location, and loaded onto the system through the PC in a relatively small amount of time, when the attacker has access to the PC.

The software placed onto software repositories by an attacker may be malicious, in the sense that it damages data on systems which execute it. Alternatively, the software may gather information covertly about the accounts on which it is run. Many users regularly try new packages on software repositories, and this can cause the damage to spread both widely and rapidly. A software package which gathered information covertly could gather proprietary information from each account on which it executed, including passwords, and transmit the information back to the unauthorized user. This could give the attacker unauthorized access to an every-growing set of accounts.

Finally, on-line conferences often allow any user to request information from them, as do software repositories. IBMPC and IBMVM are IBM internal conferences in this category. This can result in proprietary data and software being sent to the attacker, who could then print out or download the information. This represents a potential leak of information to unauthorized individuals, and is a path by which an attacker can evade the usual security controls placed upon such information.

Access to Any Linkable Disks

Once an attacker has access to the VM account, he or she will typically have access to a large collection of linkable disks. The conferencing disks and software repositories mentioned above are usually accessible to any user of a system on which they reside. If the attacker must send requests to service machines for files from these disks, there is at least an audit trail showing which files the compromised account received. If the attacker links to these disks, there is no audit trail at all, and information can be gathered without any records.

Individual users often have disks without password protection. These can provide a wealth of information to the attacker, especially if proprietary information has been carelessly placed on these disks. It is straightforward for the attacker to search for all linkable disks, link to them, and search for key words such as "proprietary". Such a search can be quite widespread before it consumes enough computing resources to be noticed.

Methods of Attack

We now examine the methods by which an attacker can cause denial of the services of the information utility to its intended users, and gain unauthorized access to a VM account.

Denial of Service

Denial of service can be caused, either accidentally or maliciously, in a number of ways. If the host computer goes down, for instance, the account on which the utility runs will be disconnected or logged off when the host comes up again. If the terminal is a VM console terminal, authorized personnel must log onto the account again to restart the utility. This requires the presence of these personnel every time there is a problem with the host. It will also typically require that the password to these accounts be shared among a number of people, so that any one of them can restart the utility. This sharing of passwords is, in itself, a security exposure; the more people who know a password, the more likely its disclosure is.

Similarly, naive users may mistakenly turn the terminal off. This also results in a disconnected session, and requires attention by authorized personnel.

If the utility allows access to CP/CMS commands by whatever means, a naive user may disconnect the account or log it off, perhaps to log on to a different account from that terminal. Similarly, if it is possible to break out of the application, and into CP or CMS, the session may be left in that state, and naive users may be unable to restart the utility.

Unauthorized Use of a VM Account

We now deal with the attacks which open the VM account to unauthorized use by the attacker.

Issuing CP/CMS Commands From Within the Application

Many applications allow the user to issue CP/CMS commands from within the application. These commands may be passed into the CMS subset, which handles them. Clearly, access to these commands enables the attacker to access the entire VM account.

Breaking Out of the Application and Into CMS Directly

This same end results if the attacker can terminate the utility and be left in CMS. Many applications can be terminated with the CMS `HX` command. If this command is effective in any mode of the utility, breaking into CMS is trivial.

It is often tricky to write an application which behaves gracefully under all possible input. In most circumstances errors from invalid inputs are merely annoying; in utilities accessed through public terminals, they can be security holes. IOS3270 is an IBM product used to provide full-screen I/O to applications using 3270 terminals. When presented with APL characters as input, however, IOS3270 complains about illegal input. When it does so, the application may leave full-screen mode, and enter a mode in which the user can terminate the application by entering `HX`. Thus, if

APL characters are available from the public terminal, an attacker may be able to use this feature of IOS3270 to break into CMS. There are doubtlessly many other examples of this problem.

Breaking Into CP, Then Restarting CMS

Even if it is impossible to break out of the application and into CMS directly, it may still be possible to break into CP, then restart CMS. If the CP BRKKEY ("break key") is available on the public terminal, it enables the attacker to get into CP. The attacker then need only restart CMS without restarting the utility.

Whether or not this is possible is dependent upon the way in which the utility is started, upon possible local modifications to CMS, and upon default IPL procedures when starting CMS.

If the utility is started manually each time the account is logged on, the attacker need only IPL CMS to have access. If the utility is started at the end of the PROFILE EXEC, which is typical, the attacker may be able to enter an `HX` command while the PROFILE is executing, in order to halt its execution before the utility is started.

If the account is set up without the AUTOCR parameter to the IPL command, the system enters a VM READ state before executing the profile. This requires the user to press the Enter key before the IPL proceeds. At this point, the attacker can simply enter:

```
ACCESS 191 A (NOPROF
```

instead of pressing Enter alone. This IPLs the session without executing the PROFILE, and the attacker now has access to CMS.

Even if the account has the AUTOCR parameter specified for it, there may be local CMS modifications which can be exploited. At some installations, the user may IPL the account by saying:

```
IPL CMS PARM WAIT
```

This overrides the AUTOCR parameter, if it is specified in the account, and causes a VM READ state before the PROFILE is executed. The attacker can then access the 191 disk as above, and is in again.

There are a plethora of ways to break out of an application and into either CP or CMS. This discussion only touches on a few of the more obvious ones. It is difficult, and perhaps impossible, to list every possible avenue of attack, and this remains one of the richest ways for an attacker to gain unauthorized access to a VM account.

PC-DOS Session Modifications

A possible solution to the problem of damaging input (BRKKEY, APL characters, etc.) is to use a PC with a terminal emulator, set up in such a way that the offending characters and keys are not accessible to the attacker. This is a plausible approach, but it can be tricky. Essentially, the attacker may try to use the resources of that same PC, which can be substantial, to mount an attack.

Many terminal emulators give the user the ability to switch between VM and PC-DOS sessions, without terminating either session. If the attacker can switch to

the PC-DOS session without disconnecting the VM session, all of the facilities of PC-DOS are available for the attack. Suppose, for instance, the security of the terminal system depends upon the PA1 key (which is the default BRKKEY for CP) not being defined in the terminal emulator. If the PC-DOS session is available, it is possible (and perhaps even trivial) to redefine the available keys by modifying the contents of the PC's memory while the terminal emulator is resident. The attacker can define a PA1 key, switch back to the VM session, break into CP, and re-IPL CMS as described above.

PC-Based Trojan Horses

Even if the PC-DOS session is not available while the terminal emulator is resident, the PC provides further opportunities for attack. PCs are typically started by booting off of an internal hard disk, or off of a floppy diskette which is left in the diskette drive for just this purpose. In a publicly accessible terminal, the attacker can change the software used to boot the system in order to set up favorable attack conditions.

Software that appears to do one thing, which would not violate security, but in fact does something which will violate security, is called a Trojan Horse. Like the original Trojan horse, it is designed to lull the parties to be attacked into a false sense of security. At the appropriate moment, the attack can then be launched.

In the case of a PC-based public terminal, for instance, the terminal emulator may be replaced by one favoring the attacker. The substituted terminal emulator could have the PA1 key defined, whereas the original one did not. It is very easy to do this in such a way that no suspicions are aroused on the part of the personnel responsible for logging onto the account. By installing a new terminal emulator, the attacker can make it appear that the account was disconnected for some obscure reason (not at all an uncommon event). When the account is re-logged on, the attacker then has access to the VM session by virtue of the replaced terminal emulator.

Methods of Limiting Attacks

We have already mentioned special PC-based terminal emulators as one potential method of limiting attacks. A specific example is given, and two other methods are discussed: creating a special version of CP/CMS, and accessing the utility only over dialed lines.

Special PC-Based Terminal Emulators

The idea behind having a special terminal emulator is to isolate the VM session from damaging input by limiting the keys available from the terminal emulator. MYTE (an IBM Internal Use Only terminal emulator), available from the PCTOOLS software repository, offers enough reconfigurability to be useful in this regard.

Keys in MYTE can be defined as follows.

- [jump a] is the only session-switching key defined. [jump a] causes the terminal emulator to switch to the first (and, in this case, only) VM session, regardless of the current session. This allows initial access to the VM session from

PC-DOS but, once in the VM session, there is no way to return to the DOS session. Once in the VM session, pressing the [jump a] key further will not switch to any other session.

- There is no APL ON/OFF key defined. When CMS is IPLed, APL is OFF by default. By not defining the APL ON/OFF key to MYTE, the attacker is unable to transmit potentially damaging APL characters to the VM session.
- There is no PA1 key defined. PA1 is the default CP BRKKEY. By not defining the PA1 key to MYTE, the attacker is unable to use the BRKKEY to enter CP.
- The Ctrl-Alt-Del sequence on the PC is deactivated. This is an additional precaution, which requires a full power-on restart of the PC in order to reenter a DOS session. This guarantees disconnection from the VM session.
- All keys not explicitly defined as useful functions are explicitly defined as [dead]. Keys defined as [dead] to MYTE send no keystroke information if they are pressed. MYTE has a set of default keys defined, so that most keys send keystroke information even if they are not defined explicitly. Defining all other keys as [dead] is a precaution to ensure that there are no potentially damaging keystrokes available to the attacker.

Using a special terminal emulator, by itself, does not deter Trojan Horse attacks. If the terminal emulator is kept on a hard disk, or on a floppy diskette that is left in the PC at all times, an attacker can substitute a more favorable terminal emulator for it. This can be prevented by putting in place the following policy.

1. A bootable floppy diskette should be prepared, which contains PC-DOS, a special terminal emulator (as described above), and any other files necessary to run the dedicated public terminal. (It certainly should *not* contain the password to the VM account.)
2. When not in use, this floppy diskette should be kept locked up in a place where only personnel authorized to log on to the VM account have a key. This prevents tampering with the diskette.
3. Whenever it is necessary to log onto the VM account, an authorized person removes the diskette from its locked location, inserts it into the PC, turns the PC's power off, then back on, and allows the PC to boot from the diskette.
4. The account is logged on and the utility is started as usual.
5. The diskette is removed from the PC and returned to its locked location.

It is critical that the account *never* be logged on without following this procedure. Even if the VM system's logon message (or logo) is already present on the PC's monitor, indicating that there is a host connection, it is possible that the terminal emulator is a Trojan Horse. The way to guard against this is to restart the PC before logging on, with a terminal emulator that is known to be safe.

In conjunction with the above policy, hard disks should be removed from PCs used as dedicated public terminals. They seldom serve a useful purpose in these PCs, and their availability only increases the chance that someone will accidentally boot the

system using the hard disk, which increases the chance that a Trojan Horse will be introduced.

The advantage of this approach to securing dedicated public terminals is that it is both simple, and can be implemented immediately with available software.

There are several disadvantages. Clearly, a special terminal emulator, and a rather inconvenient rebooting and logging-on policy are necessary. This approach does not deal with the denial of service problem, and the services of the utility can be denied by simply turning the PC off, which disconnects the VM session. More worrisome that these, though, is the fact that this approach is still likely to have security exposures associated with it. There are a large variety of ways to break into CP/CMS from a VM application, and it is difficult or impossible to anticipate them all. Using a special terminal emulator guards against certain kinds of attack, but gives the attacker access to the VM session should a new, unanticipated, attack be tried.

Special Version of CP or CMS

Another approach to the problem is to use a special version of CP/CMS on the account. The idea here is to limit the privileges of the account, and the available CP/CMS commands, so that the damage that an attacker can do is limited even if the attacker has access to the VM session console.

This approach has the advantage that access to certain system resources can be restricted. For instance, the account may be unable to send or receive VNET messages. Or, the ability to link to other disks may be eliminated.

This approach has a number of disadvantages. The special version of CP/CMS must be maintained, and this may put additional burdens on the system administrators and/or the owners of the account. As in the first approach, the denial of service problem is not addressed. There may be technical problems with this approach, in that utilities often need access to particular system resources such as VNET or other disks. Since the account must allow access to these resources by the utility, it necessarily allows the attacker access to them. Note that this approach explicitly allows the attacker to have access to the VM account, without either proper identification, or knowledge of a password.

Accessing the Application via a Dialed Line

It is possible to set up the utility so that the public terminal is not a VM session console at all. Rather, the terminal is connected to the utility via a dialed line, and is regarded as a simple I/O device by the utility. The idea of this approach is to deny the session console to the user altogether. It is still the utility's responsibility to deal with all input gracefully. But, even if it fails to do so, and the application exits, the user's terminal is still not a VM session console. Thus, an attacker may be able to cause a denial of service if the utility is not well written. But, as long as CP/CMS commands are not directly available from within the utility, the attacker has no way to obtain general access to the account.

Because the user does not have access to the VM session console, dedicated public terminals used with this approach may be PCs, and may run any terminal emulator.

Special terminal emulators are not required. The terminal emulator may be kept on a hard disk for convenience, and the PC may be booted from the hard disk, since there are no effective Trojan Horse attacks. Similarly, there is no need to have a special version of CP/CMS.

There are several examples of such systems being used internally in IBM. The GUARDIAN call-back system for remote access to VM systems is run as a disconnected virtual machine, which is accessed through dialed lines. This prevents users of this utility from breaking into the GUARDIAN account itself, and limits them to their own accounts, which are accessed *through* GUARDIAN.

The PVM system, through which users can log on to any VM machine on VNET, is another example. Like GUARDIAN, it limits the users to their own host sessions, and denies them access to the PVM session.

The CAPTURE system, which controls the output of proprietary material on publicly-accessible printers, is perhaps the best example. A user sends a document to be printed on a public printer, and specified that it should be "captured". The file is held at the printer until the user arrives. A dedicated, public terminal attached to the printer allows the user to enter a password to release the file for printing. The CAPTURE system accesses these terminals via dedicated, dialed lines. This service is available 24 hours a day to all users, without giving them access to a VM session console.

There are a variety of CMS I/O packages which support the use of dialed access. Many of them allow the input and output streams to be specified as parameters, so that a single program may be used to support both direct, and dialed access, depending upon the parameters supplied to it.

Several IBM Internal Use Only packages support this kind of access. The DISPIO package can be used with any language supporting PL/S subroutines. The FSX package supports REXX. Both of these packages support full-screen I/O.

The use of dialed lines can provide additional benefits. With session consoles, applications are limited to one account per terminal, since CMS is a single-tasking operating system. If a utility is to be provided from several terminals, it requires an equal number of VM accounts. With dialed lines, a single account may control an unlimited number of terminals, since these terminals are simply I/O devices. The utility could then exist on only one account, regardless of the number of terminals it serviced (subject, of course, to performance constraints). This can reduce the system overhead associated with dedicated public terminals.

The denial of service problem can be solved by using dedicated, dialed lines into autologged service machines. A dedicated line is one which is always at a fixed I/O address, known to the host. The utility can always find the terminal at that address. An autologged service machine is a VM account which is logged on as a disconnected virtual machine automatically whenever the host is IPLed.

When the account is logged on, the utility can start automatically, and connect itself to the dedicated line that serves the terminal. The utility is then available via the terminal whenever the terminal is turned on and connected to the host. A PC used as a dedicated public terminal can boot off of a disk which remains in the PC, and

switch to the host session automatically. In the event of a power failure, the terminal is automatically reconnected to the utility when the power is restored. If the terminal is turned off, any user can turn it back on, and the utility will be immediately available.

The use of dialed lines, and especially dedicated dialed lines, also eliminates the necessity of sharing VM account passwords among a number of authorized people. Since passwords are not necessary to access the utility, any user can connect the terminal to the utility. The connection can even be done automatically, as already noted. This eliminates the further security exposure associated with a number of people knowing an account's password.

Thus, there are a variety of advantages to this approach to dedicated public terminals. The most important is that there is complete isolation of the user from the VM session console. This prevents the user from gaining access to the VM account. Another important advantage is that the denial of service problem can be minimized, and can be eliminated entirely by the use of dedicated, dialed lines and autologged service machines.

The primary disadvantage of this approach is that the I/O of the utility must be written (or rewritten) to take advantage of packages that support dialed I/O. There are, however, a number of such packages available.

Conclusion

Potential security problems with dedicated public terminals are readily solvable with existing tools. Among the alternatives considered, the author recommends the use of dedicated, dialed lines as a means of accessing the utility, and the use of autologged service machines to run the utility. This combination provides a high degree of security against attack, and provides the highest possible degree of availability for these utilities. Both factors are increasingly important as dedicated public terminals become a common means of accessing information.

Acknowledgements

This paper is a compilation of a number of observations and techniques about public terminals, only a few of which originated with the author. The remainder came from lively discussions on this topic on the SECURITY and MYTE forums on the IBMPC conferencing system, and on the SECURITY forum on the IBMVM conferencing system. The author is grateful to the participants in these discussions for their many good suggestions. The author is, of course, solely responsible for any mistakes or misinterpretations which have found their way into this paper.