Security Tips and Techniques for Beginners

Terry W. Simpkins
Spectra-Physics
Retail Systems Division
959 Terry Street
Eugene, OR 97402

This paper is not the definitive answer for all security issues or questions. It is designed to be a relatively high-level primer for people who are either new to the 3000 environment, who have just taken over responsibilities for security, who have suddenly become interested and aware of security, or who are looking for some general tips to help them get started in the right direction.

I will not go into dramatic detail on any one of the areas, but rather give you several areas and parameters of security to look at, consider and evaluate. You decide which ones are the biggest risk for you and would have the most value for you to pursue further.

The general areas that I will talk about are: passwords; hardware access security; message system that MPE provides; specific programs to be aware of that represent a potential security risk; and monitoring activity and looking for trends--closing the barn door after the cows have escaped, so to speak.

First, let's talk about passwords. Everybody knows passwords are good, kind of like motherhood and apple pie. Auditors believe that everybody should have a password and I probably agree with them. At least one level of password security is appropriate for almost everything. Although there may be a couple of exceptions, they're few and far between. Should you use the MPE security system and the password mechanisms in it, or one of the aftermarket products? That's up to you. Your decision will be driven by your user's preference, your preference, your budget, and what kind of system you want to design and provide for your users. Any answer can be the correct one, if your logic is sound.

There are a couple things that are important: one, passwords should be non-trivial, that is, they should not be obvious, they should not be one letter, and they should not be the defaults from your vendor; two, they should be at least three letters long and relatively easy to remember. Passwords can be non-trivial, non-obvious, and still easy to remember. The worst scenario is to create a great password, just to have people write it down and tape it to their terminal. You have to strike a balance between security and workable security. Passwords should not be carved in stone, that is, they should be changed on a fairly frequent basis. I have found every three to six months is a reasonable timeframe. You want people to remember their password so don't change it every week because they'll have to write it down to remember it, and you don't want them to write it down. At the same time, you don't want passwords to become widely known. You should, if possible, utilize a scheme where passwords are only valid for certain people at certain devices. For example, "ASK," the software package that we primarily use, has a mechanism for restricting what commands are executable by which password, and this is not your MPE password, but rather your ASK password which you have to supply to get into the software. The ASK password allows you to use specific passwords to restrict users to specific databases. If you have a test database, a production database, and a development database, a given password may only allow access to the test database, or it may restrict access to the development and the test databases but not the production, or allow access to all three, or any combination. The password can also restrict you to what commands you can run and let you restrict what MPE user may log into the ASK system under this password. For example, if my ASK password is Terry and my MPE logon is Simpkins.Manman, I can setup the system so that to use the ASK password, Terry, one has to be logged onto the MPE user Simpkins.Manman. Now, "Terry" (my first name) is a lousy password to have in ASK, but it is a good <u>mechanism</u> to use so that you can control more closely what users are doing in your system. Don't misunderstand me, I'm not doing a sales pitch for ASK--I'm only one of their customers. I'm merely using ASK features as examples of what you can do in an applications security scheme. There are other methods, there are other applications security schemes, I'm not encouraging one over the other.

The next security area we need to talk about is hardware access. Hardware access can be gained three ways: direct connect, i.e., to an ATP or ADCC; through a phone line via a modem; or a DS line from another system. Hardwired--pretty straight forward. If people have access to your building and if they have access to a terminal, they're in. At that point, you're faced with controlling physical access to your building and physical access to the terminal. Are your terminals in a locked room, or do you encourage terminal use and set them on peoples' desks? You probably do the latter--you set them on peoples' desks. If you do that, you have to rely upon an informal security network which is someone walking in on an unauthorized user and saying, "hey Bob, what are you doing on this terminal, you're not supposed to be here." Or, you rely on your password and applications software security mechanisms to prevent unauthorized access.

Telephone lines are a different issue. Dial back modems, modems with passwords, etc., are all options you may choose. What you use will depend upon how many phone lines you have, how frequent the access is, and how varied the audience or the user base is. If only your programmers use the modem, it's probably less of an issue, or actually, maybe it's more of an issue, because you don't want those call-back mechanisms or passwords to have to be used. One scheme that I've seen used successfully is to write a very simple SPL program, that is a logon, no break, UDC, that every user goes through when logging on. All this program does is ask, "am I running on LDev21? Yes or no." If no, I'm done. If yes, "Is the person logging on authorized to use LDev21?" If yes, fine, let them proceed, if no, bounce them. You can define what is an acceptable logon as narrowly or broadly as is appropriate for your organization. This is a pretty simple program to write. All you need is the "WHO" intrinsic, and the "WHO" intrinsic is relatively simple to call from SPL, or COBOL.

With dial-up-lines, I'm a firm believer in changing your telephone number to the modem on a frequent basis. The exchange that the phone number comes in on (the first three numbers in the telephone number) should be different than your company's voice telephone lines (your published phone number) just to make it a little bit more difficult for people to discover. It's not that difficult for your average grade school or teenage hacker to write the program described in War Games, one that dials all the telephone numbers in all the local exchanges and finds all the ones with carriers. If you wrote such a program, you'd probably be surprised how many carriers you would find--there are a lot of computers out there with a large number of dial-in lines. I recommend that you change the phone number on a fairly frequent basis and always change it when someone that knows the number leaves your company. When you change it, don't just increment the number by one or decrement it by two, take it to a different exchange. The price associated with changing the phone number is not very great, where we are it's about $50; therefore, we do it every three to six months. We notify the programming staff the day the number changes--we don't give them a lot of notice. If they happen to be absent that day, they don't find out about it until they return.

If you are more concerned about restricting access to your phone lines, you have a couple of other options. One, you can "down" the phone lines. If you have 24-hour a day, 7-day a week coverage by your operations staff, this is probably a viable alternative. For example, say a programmer wants to logon, he calls in and says, "hey Bob, up the modem, I'm going to call in and do some work." Then the operator can "down" the modem when he sees the programmer log off. If you don't have that much coverage, or maybe you don't have any coverage at all by the operations staff, (you're it) this is not the easiest solution. There are lots of ways you can get around it, they just take more effort. I would venture to say that at least 90% of us have at least one modem on our machine, and that's the one HP sent us. Even though you don't use the support link for anything other than sending your HPTREND Reports to HP, or the occasional Response Center call, someone out there is trying to find your system, and how to get into it.

DS access is a little bit tougher to control because you probably have a couple of machines DSed together, and you have batch jobs logging on back and forth all the time at random occasions to transfer files, or trip flags, or look up data for validation. In this case, "downing" the device isn't necessarily a very good option for you. The best alternative I can come up with if you're extremely concerned about access is a variation on the modem security program. Have another table that you check for valid logons that says, "oh, yea, he's coming in across the DS lines and he's logged on to that corporate account, that's ok, we'll let him go," otherwise bounce him.

If you have a person that you have reason to suspect as being a security risk at another division or location of your company, or on another computer, a variation on the modem security program might be a viable, and fairly intelligent solution for you. Remember this about DS access, on certain versions of MPE, capabilities of the session on the source machine travel with the user to his session on the target machine. I don't remember which version(s) of MPE had this "feature" but it used to exist. One of your alternatives is obviously to down your DS lines and require people on the remote machine to ask you to up the DS line. The problem is if you have batch-oriented processes this procedure can get a little bit cumbersome and hard to manage. You will need to either have the operators at the other machine call up, or you'll have to define a very specific time window for batch processes access your machine.

The point I want to make with all of these scenarios, Passwords, Modem Access, DS Access, and Local Hardwired Access, is not that any one particular security method is better or worse than another. They all have pluses and minuses and they're all appropriate in some instances and inappropriate in others. The message here is that to make points with your auditor, you first have to think through the issues as they relate to your installation. You need to have defined your objectives--what it is you're trying to accomplish--to spell out the alternative you're going to use and then have clear, concise reasons regarding why you're using one particular approach versus another to meet the needs as they're defined for your installation. If you have that, what you'll find is that the auditors will give you a lot of points. They may or may not agree with your approach, but they will at least understand what you're doing and why you're doing it. They will be able to score you in a rational, thoughtful manner, as opposed to reacting like, "you haven't thought about it, therefore, it's bad."

Best laid plans of mice of men often fail and, for some reason, somebody that you don't necessarily want to access your machine has, in fact, gained access. How can you minimize the damage? First, don't make it easy for them. Don't paint them a road-map. This is where the message file can be a real friend of yours, or, as it's delivered from HP, actually be somewhat of an enemy. Take a good look at some of those messages. The messages are very user-friendly and try to help people understand exactly what they are doing correctly and incorrectly. As System Manager in charge of security this is a problem for you because message files can be a road map to hacking. I would refer you to an article in the September 1987 issue of Interact on this very topic. Here again, let's refer to our War Games example. You have a kid that doesn't know a HP 3000 from a bag of M & Ms, but he's got a carrier and now he's going to try and get in.

The point is that with trial and error, it doesn't take very long with the way the messages are structured, and he's going to have the exact format and the context that he needs to logon. Now, it's a matter of hitting a valid combination of user and account name. Then he simply takes his little Basic program that he used to find your dial-up phone number, modifies it to try every combination of alpha numeric, eight character long words, and records the message that he gets back. Pretty soon, he's going to get you. Then he's going to be down to trying passwords, and guess what, now he can take the exact same program and try it with all 36th-to-the-8th power combinations of letters and numbers and he's going to come up with a valid password--he's going to get in. I've never really tried this, but I would guess that within the course of a couple of nights, he could break almost every machine. If you're diligent, and you happen to look at your console logs, you might catch this and you'll say, "oh, jeez, somebody's out there hacking away at my modem," and you'll down your modem. But, what if you happen to have the scenario where you can't "down the modem." Now, you have problems. Change the phone number! Quickly! Think about Police involvement and traces!

Ok, you need to keep an eye on what's happening on your machine. Like I mentioned, you get somebody hacking away at night and, unless you happen to read the console log every morning (which I'm sure is not on the latest best seller list) you're not going to see a lot of those messages. What we have done and I recommend that you all do, is to write several little programs to monitor what's going on in our system. Recap the information in a format that is easy and quick to read so you can make some sense out of it.

First thing we've done is write a program that looks at the log files. You tell it which log files you want to look at and it will scan those log files and report out facts or information I want to see on a regular basis.

First thing you do is modify your Cold Load configuration to log everything. It doesn't take up that much disc space and you're not going to keep these log files in your system for very long anyway. All you need to do is use this information once to solve a problem and you're going to pay for a lot of $15 tapes used to store these log files.

What I do is log all console messages. This lets me go back, read the log files, and report security violations that have appeared on my console. I can use log files, then, to look for trends. Do I see a lot of them coming for one particular LDev? Do I see people trying to hack a given user? Does there seem to be a lot of them coming late at night when there's nobody here but the security guard? Things like that. Do a lot of them come in across my modem? Ah, maybe somebody has learned my phone number, I need to go change my modem phone number--that's the first thing.

Secondly, I can keep track of who's purging files. By logging file closes, you can record all file purges. This might be real interesting to help you discover if the purging of a file was an accident. It might also show you that somebody was trying to cover their tracks. If you see somebody purging one of your log files, then you might want to go look at that log file very closely and find out why somebody would want that particular log file to disappear. What kind of incriminating evidence is in that log file?

Lastly, certainly not least, is that I look at all logons on certain LDev's. Specifically, the ones that I look at very closely are, anybody logging in on my dial-up modem, and anybody logging in across the DS line. Are those the people I expect, or is there something funny going on?

All these programs I've just talked about are on the swap tape here in Orlando. These programs were originally written by Harold Jensen. Harold used to be a tech support programmer with Spectra Physics. He is now an SE at HP. I do not know if Harold has ever contributed these programs before; they were a part of a system that he used to track and report resource usage and trends. We have cloned some of his software, combined a couple of programs, taken out some functionalities to fit our specific needs.

The next thing we have done is to make use of a part of MPEX to read the system directory. We used MPEX because it uses Privmode to go right into the directory, we did not want to write that kind of code and then have to maintain it. We had the MPEX product already and it was a relatively simple thing to create some job streams to do what we're looking for. We use MPEX to look at all programs on the system that are prepped with PM, tell us what those are, tell us if those have lock words on them, tell us when they were last accessed, and tell us if they're released. The way we do that is through a series of "listf" steps with our own defined listf mode which MPEX allows.

The last thing we did was to use the indirect reference to a disk file of the LISTDIR5 command. We list off all accounts, groups, and users to a disk file. Then we have a straight forward little COBOL program that reads the file created as output from LISTDIR5, and reports any user, any account, or any group that has specific capabilities, and whether or not it has a password. By definition, all of our users should at least have one password. This method lets us track our compliance to our standards. We want to make sure that we have certain accounts and groups passworded because of the relatively sensitive nature of information in those groups and accounts. At this point, I'd also like to remind people, just because you have a password on a group, a user, an account, doesn't mean it's secure. A lot of the third party vendors (HP falls into this category) have default passwords that go with their software. That password is the same on every system in the whole wide world with that piece of software. Why would that make your system secure just because there's a password on it? Everybody in the world knows what the password is. So, in the article published in the September 1987 Interact, I've listed some of the more commonly used 3rd party software and the default passwords that come from the vendor. You should look at these and make sure you are using different passwords.

All these problems go away if you change your passwords on a regular basis as discussed earlier. Take a look at the list, if nothing else, pick up a phone and call up your vendor. Tell them who you are, ask them what their default password is, and make sure yours isn't the same. A perfect example of this would be your HP support accounts, like TeleSup and Support. These accounts are standard on every HP machine in the world. These accounts contain PM code and their passwords should be changed to one that is unique to your site. Also access to the accounts should be restricted to account users.

The last thing I want to mention is some of the PM (privilege mode) programs that you need to be aware of that could be on your system and represent a hazard to you. I mentioned how we used MPEX, to look for released PM programs that are not locked or protected.

Let me bring your attention to a few programs you should keep your eyes on.

The first one, named "God" is from VeSoft. "God" is a neat program because it'll let you do about anything you want to as far as giving yourself capabilities during your session. Fortunately, VeSoft lockwords that program when they ship it to you. First thing I do is "Restore @.@.VSoft," and as soon as that's done, purge "God." I would recommend the same thing, you have the tape, you can always restore it. Get it off the system.

The next one I would lock up is called MakeSmop. It comes from Kelly if any of you have their RAM disc. I don't have personal experience with this one, but I've heard that it will let you give yourself SM or OP capabilities no matter where or who you are so that's one you want to be aware of.

In addition, there are many such programs in the Telesup and Support Accounts. I'm not going to list all of them here, but I would recommend you sit down, talk to your SE about what all of these programs are and what do they do. These programs should be stored on a tape. Put the tape in your desk drawer; you can always restore the program if needed or restrict access to the Support and Telesup accounts to account users and keep them well passworded.

As mentioned earlier, this is not an all-encompassing security tutorial. Rather, use this as a starting point in evaluating your system's security needs. Security is much more than a password.