SYSADMIN ADMONISHMENTS

Kenneth G. Olthoff
10 Terrace Drive
Linthicum, MD 21090
Olthoff@earthlink.net

ABSTRACT:
There are many articles, books, tools, classes, etc. that deal with security in terms of designing systems, or operating particular systems. The following is an effort to fill in more general thoughts about secure operations of a generic system. While some of the points presented assume that the system is a computer or telecommunications system of some sort, the same principles can in most cases be adapted to other systems as well. This paper will avoid the specific and attempt to discuss more general rules of what to think about, what questions to ask, and pitfalls to avoid when one is trying to actually use a system securely. Given the context of cyberterror, we will attempt to highlight areas where cyberterror considerations may add a different nuance to the usual security posture.

Author's ground rules: The reader is encouraged to question the author's assumptions and prescriptions. Think through the issues presented. Ask yourself if there is something I misstated, or a point on which I am wrong. Try to determine for yourself if there is an area of discussion, a line of reasoning, an example case, or a logical point that I missed. In this way, even if you totally disagree with my ideas and conclusions, the paper will still be a success.You will emerge from reading it with a better understanding - even if you found it for yourself.

Begin all deliberations with the question of whether, and why, you care if your system is secure or not - depending on the answer, you may save yourself some effort.

It may seem obvious, but the question should be posed up front. If the answer is that you *really don't care* about anything that could happen to the system, there is no need to go further, except perhaps for personal curiosity. The second possibility is a little trickier - if you care somewhat about the system and its contents, then you will need to make a decision about how much effort and expense a given amount of security or lessening of risk is worth. Instances where a system is truly worth a "secure at all costs" effort are likely to be very rare. Regardless of what the answer is, though, ask the question! The answer will help determine the rigor and the focus applied to any subsequent activities.

In all considerations though, be sure to consider not only the usual risks – loss of profit, inconvenience, physical damage, injury or loss of life, but also secondary and tertiary effects. Who else might be affected by damage to your system, or it being compromised? Be sure to factor in risks where your system or your organization may be at risk for reasons other than those directly related to you.

How do you spell security? (What is your policy? Your environment? Your priorities?)

This set of questions revolves around setting the parameters of what you want or need. The idea is not to prescribe a particular solution, but to state what you wish to accomplish, and under what conditions. Take care in getting back to the basic issues, but no farther. For example, a stated goal of "secure Email between A and B using <insert detailed legacy system configuration here>" may overlook the obvious - that it may be cheaper to seriously change or replace a poorly designed system than to try to patch all the holes in the existing system configuration. By drawing the boundaries too tightly, one ends up specifying a particular implementation, not the desired functionality. If the goal is revised to "secure communication of computer data between A and B", there is no boundary on the solution, yet one needs to be careful to not go off the deep end with such freedom. An extended analysis of the efficacy of carrier pigeons carrying CD ROMs or some equally unlikely solution is probably not worth the effort.

In any case, try to establish the policy at a generalized level, and to make sure that specific cases can be traced back to the general rule. Starting at the specific, whether in policy or in implementation, tends to make things inconsistent and confusing, both of which are properties that the adversary can make use of.

There are many systems which cannot be adequately secured - figure out if you have one, and be aware of the limits to damage control.

By now, you have pondered what you wish to secure and why. You have hopefully formulated some generalized notion what "secure" means to you. It is now time for some examination of your system, available technology, research, and your own intuition. It is possible that you will find yourself confronted with the words of the prophet "You can't get there from here!"

In such cases, be careful. Arthur C. Clarke once described sufficiently advanced technology as being indistinguishable from magic. Thus, when something looks impossible, it may simply mean you're not thinking about it in the right way, or that you're not being clever enough. It should also be noted that throwing up one's hands and declaring defeat just to get out of the hard work of doing things right is considered bad form.

If you are sure that the cause is lost (and given the nature of the problem, that's quite possible), the best one can do is recognize the fact, do what you can to optimize your risk, and plan for damage control. Always make clear to whoever needs to know, though, that the system is in fact insecure, and provide as accurate an assessment as possible of to what degree and in what way it is insecure, and what damage control is possible. The proper decision may be that the likely damage is not worth the benefits, and the system might be reduced in scope, size or function to lessen the potential problems. On the other hand, it may be determined that the benefits are worth the risk - "Damn the torpedoes - full speed ahead!" Either way, it should be a case by case call, not an automatic default to one or the other extreme with no further thought.

If you are connected to anything not under your full and direct control, and do not take precautions assuming it to be hostile, you are betting your security on the hope that none of the designers, administrators or users of that system are adversaries, idiots, or prone to human error.

Let's ignore for a moment the possibility that one of your users, or one of the people who built or configured *your* system may in fact be an adversary, an idiot, or prone to human error. Let's also set aside for a moment the obvious problems if *you* fall into any of those categories. After all, it's unlikely that if you were an adversary that you'd be trying to catch yourself (though the mental image of somebody trying to do just that is quite humorous). If you are, in fact, an idiot or prone to human error, there's not much you can do about those problems in the near term, either. Education may help in the longer term, but even then, there is the possibility of being "book smart" but having the common sense and intuition of a gnat. And of course, everybody makes mistakes, no matter how good they are at what they are doing.

Self evaluation is left to the reader, with the hope that those who find themselves lacking in some area will recognize that fact, and take measures to get more expert help or compensate in other ways. A self-deluded designer or administrator is a *major* security problem.

With those possibilities on the back burner, let's look at the basic issue of trust in the unknown. No matter what your system/site is hooked up to, if you don't have full knowledge and control, it needs to be viewed as a potential source of trouble. Don't limit your thinking to the connections of the system itself - if your computer is in a basement, a clogged sewer and a faucet left on in the bathroom is potentially just as effective a denial of service weapon as any electronic signal.Such an attack has the additional benefit to the attacker that unfamiliar people entering the bathroom are not usually viewed with suspicion, while strangers entering the operations center might be. Few people thought of commercial airplanes as a weapon until it happened. Try to think of what unrecognized attack vectors you may be overlooking every day.

One of the most common errors in the field of security is to assume that somebody else's system (especially those of colleagues, partners, or allies) is secured in a way that will meet your needs, and to let down your guard in dealing with them. If a system is to have a chance of being secure, there needs to be a healthy sense of distrust in all interactions. A common obstacle to implementing such a strategy is that we feel that doing so may hurt the feelings of those running the other systems. Instead, we need to foster a view that being cautious indicates a prudence and diligence which shows our concern for others as well as our own system and users. We need to encourage others to be just as careful in connecting to our system for their own protection. This is not to say that such behavior guarantees security. It doesn't, but if you are cautious rather than blindly trusting, you at least have a chance of being less insecure.

As a rule of thumb, it helps to default to having connections and services turned off. Only after a need has been demonstrated (usually by a user screaming "How come I can't <fill in the blank> today? What did you do?") should you consider turning on anything. And even when a need is shown, the service or connection should not be allowed until an analysis of the risks has been done and a conscious decision to accept the risk has been made. Also, do not limit your analysis to just information systems or telecommunications systems. Anything, including electricity, water, gas, personnel, etc. which is tied to or interacts with your system or your site is a potential source of trouble. You may not be able to do anything about such risks, but be aware of them.

Note that the inverse is also true. Who or what are you connected to or otherwise interacting with that might be affected by problems with your system? What have you done to limit the downstream effects of problems at your site? As an example in the network realm, many people have intrusion detection systems to look for malicious traffic that is inbound. How many have similar systems to look for malicious traffic outbound? If your system were hosting a “zombie” would you have any way to detect and stop it?

The chances of a system being secure in any configuration are inversely proportional to the complexity of the system. Bonus penalty points if the system is too large for a single person to grasp in its entirety.

This is not a matter of “bigger is bad”. It is possible for a small system to be very complex, or for a large system to be relatively simple in design. The issue is one of permutations. The more complex a system is, the more possible ways there are for the pieces of the system to interact. As the permutations increase, the possibility of there being a state, configuration, mode of operation, or sequence of events that has not been adequately tested or analyzed increases. If timing issues are thrown into the mix, it gets even harder to say with certainty what the system might do under any given set of circumstances.

In simple terms, what you don’t know might hurt you, and anything that increases the list of things you don’t know about your system is potential trouble. Many is the system which has come to harm in a situation which a designer dismissed by saying “Nobody will ever do that!” or “That set of events will never occur.” or “It’s not possible for it to get into that state.”.The classic systems engineering cartoon with a diagram containing a box labeled “Magic occurs here” can be relabeled for security purposes by replacing the word “magic” with “havoc”. It’s hard enough dealing with the security problems in the stuff we understand - we really don’t need to add any more problems by increasing the number of things we don’t understand.

If you care about the contents and operation of your system, you should have a written plan for dealing with the destruction, failure (catastrophic or otherwise), corruption, or subversion of anything within the system, connected to it, or involved in its use and operation, including carbon based life forms.

It’s not a question of *if* something or someone is going to break, degrade, explode, quit, defy the laws of physics or otherwise ruin your day, but *when*. While such events are part of life, their annoyance level can be minimized if you have actually worked out what to do before the fact. Waiting until all hell is breaking loose almost guarantees that you will rush, not think clearly, or otherwise do something leading to mistakes, which will make your day of crisis far more exciting than it needs to be. Plan ahead so that you can think things through calmly in a relatively stress free atmosphere. Even if your plans don’t cover all the possibilities, they will give you a significant head start over the typical damage control plan of “just make it up as you go along”. Besides that, being able to respond quickly and professionally when things go bad tends to *really* impress those around you.

Backups are good, but independent backups in a physically separate location are even better.

The usual case made for backups is to be ready in case the computer or the disk drive fails. The threat of viruses, hackers or program glitches going undetected for some period of time while they corrupt the system is the scenario usually thought of to argue in favor of keeping a backups extending far enough back in time. The third case which is less often thought of is the situation where flood, fire, that weirdly quiet guy from the next office suddenly going berserk, or some other calamity destroys the system and its surroundings. At that point, having backups stored in the same room may not be of help, since the water, flames, giant arachnids from outer space, or "good old unstable Olthoff" probably got to the shelf next to the computer at the same time that they trashed the computer itself.

As always, the possible loss from rare events must be weighed against the costs of recovery measures, but be sure to cover all the possibilities when doing the trade off analysis. Overlooking the possibility of major facility damage has been the downfall of many an otherwise adequate recovery plan. Be sure not to forget to plan for backups of the various services, comunications links, information sources, etc. that are not part of your system, but that may be taken down in a large scale event. We previously touched on planning as if everything your system or organization is connected to or rely on is hostile. Now you must plan to take into account the implications of one or more of those things suddenly not being there, or at least not operating as expected.

Action is generally better than reaction, but inappropriate action may be worse than doing nothing.

The day after a security incident is a poor time to begin formulating a security policy and an implementation plan, but it is probably the most common one. Security needs to be actively pursued, instead of waiting until something happens and reacting to it. That said, it is also important to not take action haphazardly. Security measures that are poorly coordinated or out of proportion to the threat and the potential consequences may serve as more of a hindrance than a help. Be sure to think through what you are planning to do, and make sure you know what you hope to achieve and what the side effects might be. Doing something just to be doing something may be hazardous to your system.

As noted before, don't limit your thinking to solving your own problem. Make sure you also have plans for limiting the spread of damage to other systems that you intereact with, and have plans in place for notifying your partners and those you communicate with if they need to be warned of potential risks from an incident at your site.

If your system was mucked with, how would you know, and how soon?

This is one of those obvious questions that we often forget to ask. After going through all the effort to set up a system which we at least claim to care about, we often sit back after we start using it and forget to follow through. No matter what level of security one is aiming for, and no matter what the details of that system may be, there is some amount of supervision and vigilance required.

If the system is designed to detect attacks and sound alarms, somebody needs to know what the alarm means and be responsible for responding in whatever fashion is required. Periodic *independent* audit should be done to make sure that the system is working properly and that the users and administrators are using it properly. As in prior sections, there is no "right" answer. The key is to know what level of security, detection, and responsiveness is desired, and then to make sure that your system and its operation match up with those goals.

If you detect an attack, what is your first priority?
Restoration? If so, to a working state, or to pre-attack state?
Prosecution?
Retribution?

The key here is the realization that the proper course of action when an attack is discovered will vary based on the answer to these questions. Not only that, but the right course of action for one priority may be exactly wrong for another. For example, if the object is to restore the system to the state it was in prior to the attack, one of the first things one might do is yank the system off line and start fixing things. If one is going for prosecution or retribution, however, one might wish to stay on-line until you can trace back to the offender and establish a convincing record of who it is and where they are operating from. If one is going for retribution or restoration, the imperative is to alter the system either back to where it was or to a more defensive posture. Additionally, in all three cases, you need to do adequate backups on a regular basis so that you can either quickly return to the "before" state, or in the case of prosecution, so you can show the evidence of the system being changed.

If you are going for prosecution, though, the standard is higher. You need to establish an unbroken chain of evidence.You will need to prove that whatever evidence is found on the system is not altered or destroyed and that it has been constantly in known custody from the time of discovery until it is presented. If anything is changed, the evidence can and often will be thrown out of court.

In all three cases, your first moves are critical. Whatever you do may destroy evidence, tip off the bad guy, trigger traps left by the bad guy, or further damage the system. You need to have carefully thought out beforehand both what your goals are, and the specific steps to take to support those goals. Remember, if you are under attack, the odds are that your adversary has planned what he is doing, and that he is counting on your panic and/or thrashing about in the event that he is detected. Try not to give him that advantage.

Especially in the case of cyber terror, don't assume that you know or understand what the adversary is up to. It may not make sense. It may be deliberately designed to not make sense. As noted above, the key is to know what you want to accomplish in various situations, and to have plans in place to allow you to respond with as little wasted time and effort as possible. It is entirely valid to have multiple plans for different scenarios, as long as part of your planning is a quick decision tree or other process to allow a quick decision to be made regarding which plan to execute when the time comes.

Documentation lies - when in doubt, learn from protocols, code, and empirical

demonstration. Get help from experts!

As anyone who has ever done development work can attest, documentation seldom exactly matches the final product, even with the best development procedures and intentions. As has been discussed before, any time you don't understand exactly what the system is doing, or how the system is connected, there is a risk. The best way to understand the system is to take nothing on faith.Dive in and look at the code, interfaces, and protocols. Try experiments on different things in a controlled environment. Keep current on any sources of information which might provide additional information about the components of your system and what others have found. Get expert help to explain what it all means if you are not sufficiently skilled. Even if you are sufficiently skilled, get expert help to give you an unbiased, or differently biased, view - you may be missing something due to your own tendencies, background gaps, or preferences.

The adversary does not care how your system is supposed to work, and will delight in finding functions, modes and ways of tweaking it that you never dreamed of.

This is tied to the discussion about learning from the code and the protocols. Even if you choose not to do so, rest assured that those probing your system will have done so. In some cases, they will be more familiar with the code than the person who wrote it, because they will be deriving their information directly from what exists - they do not have the mental block of seeing what one meant to write, rather than what actually got written. The attacker also has the advantage of not worrying about cleaning up any messes. There may be tricks that have potentially nasty side effects, which a normal user will avoid, not wishing to mess things up for himself or his coworkers. The attacker has no such qualms. Lastly, the attacker can view the derivation of all this arcane knowledge as a game or a challenge, and unfortunately the game is stacked in her favor - she can use any means to get in, and you must plug all the possible holes to keep her out.

A fool with a tool (especially a security tool) is still a fool.

"Pride goeth before a fall" says the old proverb, and in this realm there are three big problems. The first is thinking you understand something when you really don't. The second is knowing that you don't know, and not doing anything about it. The third is trying to get expert help and picking an expert that falls victim to problems one and/or two. In short, if you think you understand the situation in its entirety, you probably have not thought about it hard enough. A new, fancy tool is unlikely to solve this problem, and may in fact give you more ways to get yourself into trouble.

Anything that is not explicitly prevented is exploitable.

The corollary is that even those things you think you've prevented may still be done. Somebody may see a different way to do it than the one you thought of. Your job, in general, is to plug all the holes that you decide need to be plugged, and to know about the rest of them. It's a thankless task, and there will almost always be something you miss. Given that, one should at least be very vigilant in dealing with the things one is aware of, and always assume there is at least something one is unaware of. Keep looking!

The attacker's motives and/or actions may be subtle, nonintuitive, or just plain weird.

It's hard to list all the reasons that a system may be attacked or probed. It's even harder to grasp how the attacker's various actions will add up to reaching his goal if one can't even figure out what the goal is. In light of this, one must throw out preconceptions. Don't assume that somebody won't do ABC, because rest assured, somebody somewhere will, even if it makes no sense to you. Remember, the activities on your system may not even be the main event. Your system could be a training ground, a reconnaissance mission to familiarize somebody with a particular type of hardware or software, you could be a target for reasons which make no sense to you, or the attacker may just be doing things "because he can". It definitely helps if you can figure out what the opponent is trying to do, just don't get too hung up on trying to understand it, and don't project your assumptions, values and motives onto the adversary. If you do, you'll probably miss something, because the adversary most likely doesn't think exactly like you.

In martial arts, one technique is to use the adveersary's strength and momentum against him. What are the preconceptions and biases that you have that might be turned against you by a clever and determined foe whose idea of "winning" is not what you are conditioned to expect?
For example, most of the systems looking for irregularities in the financial system are looking for people trying to make illicit gains. What if instead the whole object was to destabilize the market? Actions designed to incur large, intentional, and very public losses might get past many of the conventional protective measures, yet still have the desired effect of devastating investor confidence.

Fixed configurations are a luxury, not a given - be prepared to analyze and respond to changes.

A common fallacy to avoid is the myth of the static system. Most ratings, evaluations, reviews, etc. are based on the premise of looking at a specific configuration, and telling you about the properties of that configuration with little or no concern for the security or behavior of the system in any other configuration. The problem is that *very few, if any,* operational systems are static. The configuration this morning may change by lunch. Even the most static systems eventually get upgraded, changed, reconfigured, broken or otherwise tweaked. Even if the change is not to your system, a change in the systems that your system is connected to may have significant impact.

The key is to not rest on any evaluation or specific configuration, but to pursue an understanding of what change will mean. Does resetting that parameter reduce security or increase it? What is the magnitude of the change? What are your procedures for thinking through changes? You may not be able to avoid change, but you can prepare yourself to deal with it in a well-reasoned and hopefully effective way.

CONCLUSIONS:

In summary, these concepts can be applied to almost any system where security is a concern. It's not that specifics don’t matter - they do. It’s just that once you develop a way of thinking about security in general practice, you can more readily adapt to the specifics of whatever situation you find yourself in.

Learn about your own limitations and goals, and those of your system. Work diligently to improve both. Always question your assumptions. Engage in healthy suspicion. Learn to recognize patterns and deviations from them. Never underestimate the cleverness, tenacity, and capacity for seemingly odd behavior of the adversary. Realize that thoroughness and cooperation is key to limiting the spread of damage, especially when dealing with adversaries who may be deliberately trying to create unanticipated domino effects. Work for the best, and prepare for the worst. Lastly, try to have fun - the task is tough, but it can be an enjoyable challenge.