The Answer is **42** of Course

**If we want our networks to be
sufficiently difficult to penetrate,
we've got to ask the right questions.**

THOMAS WADLOW, INDEPENDENT CONSULTANT

Why is security so hard? As a security consultant, I'm glad that people feel that way, because that perception pays my mortgage. But is it really so difficult to build systems that are impenetrable to the bad guys?

OK, I just threw you a ringer. Two, in fact. The first is that lovely word *impenetrable*. A nice absolute word that makes it seem like you are completely *safe* and *secure* (two more words that give similar impressions). If we are talking about the security of your network, or any system for that matter, and you tell me that you need it to be impenetrable, safe, secure, or any similar absolute concept, it suggests that there's money to be made. By me.

I don't want to sound too cynical, and I certainly don't want to make it seem like a scam. I provide good value to my customers, as do many other security consultants and professionals. I know that I have saved my customers

money, and made them safer. But "safer" isn't "safe," and until customers and developers understand the difference, security is going to remain a difficult and expensive proposition for them.

The difficulty is in the human hunger for absolutes. Many technology issues are based around absolute concepts. The airplane flies or it doesn't. The network router works or it doesn't. The equation produces the correct answer or it doesn't.

Issues where people are involved rarely have this clarity. That is the first fundamental realization necessary to really understand security as a technologist or as a manager. Security is a people problem, not just a technology problem.

When we design a network router, we are basically solving problems of logic and physics. We can define tests

The Answer is 42 of Course

for your device, and those tests can be repeated exactly so that we can debug the device. If we have enough tests, and our device passes them, our router works.

Security devices such as firewalls often physically resemble network routers (and indeed may be part of the same device), but good tests are a lot more difficult to develop. Why? Because now our tests have to take people into account. To reflect the real world accurately, they won't all be repeatable, and the people we are trying to defend against may not play by our rules.

If we want our network to be acceptably difficult to penetrate, we've got to ask the right questions. For many people, even technically sophisticated ones, thinking about the basics of network security boils down to a question of what kind of firewall to buy. But let's step back a bit farther and try to look at this problem from first principles. Absolutes won't work, so the first question we must ask is: How can we design a network that is safe enough?

Thinking about the phrase "safe enough" should quickly make us wonder, "Safe enough from whom?" In other words: Who are the people attacking our network?

Well, that was the second ringer in the first paragraph of this article. Remember the *bad guys*? The phrase conjures up an image of movie-type criminals hunched over computer screens, or maybe 14-year-old social misfits in the family basement. But in truth, they could be anybody. They might even be people we know. And there's a pretty good chance they are people we hired, and probably trust. A truism in the security industry is: if money is involved in a security issue, look for an insider.

In my work as a security consultant, I've had the opportunity to look at many different network security systems, both successful and unsuccessful. I've seen how a variety of organizations protect their computing infrastructures and proprietary information. I've spoken to many of my peers, who have had similar perspectives. Throughout all this, one recurring theme is clear: for the majority of successful intrusions, either the barriers are very low or the motivation level is very high.

This is not to say that there aren't complex and

dangerous penetrations, or that attackers aren't using so-called "zero-day exploits" (making use of newly discovered vulnerabilities early before defenders can patch against them). It is not to say that well-defended networks are not penetrated. It is still true, to paraphrase Thomas Jefferson, that the price of network security is eternal vigilance. But in an awful lot of cases, either the network designers made it easy enough, or the attackers wanted it really badly.

SKILL, MOTIVATION, OPPORTUNITY

Given all of the above, we might make this assertion: any computer system can be penetrated if the attackers have sufficient skill, motivation, and opportunity. From there, a strategy for protecting our network falls quite naturally into two major goals:

- Requiring a very high level of skill and motivation for an attacker and presenting as little opportunity as possible for the attack to succeed.
- Deciding where and how much effort to expend in the process.

Now those are goals that a technologist can embrace and expand upon. Furthermore, those goals can direct our security efforts from the micro to the macro level.

Let's analyze our first goal a bit more, starting with some help from a dictionary: *skill* is defined as "proficiency, facility, or dexterity that is acquired or developed through training or experience"; *motivation* is "the psychological feature that arouses an organism to action toward a desired goal"; and *opportunity* is "a favorable or advantageous circumstance or combination of circumstances."

Skill. How do attackers become more skillful at penetrating a network? One way is through practice. With off-the-shelf software, attackers can set up test networks of their own and learn about them until a weakness is found that they can exploit. This requires a fairly high level of skill and motivation, because it can take a lot of patience to find a good exploit. By packaging that exploit in software and distributing it, however, many more attackers can take advantage of the flaw without having

to be smart enough to find it.

Another way is what, in military circles, would be called intelligence. Learn the design of the target network. Learn the ways in which it is guarded and monitored. Learn about the people who run it. Knowledge is power, and if you want to open something tough, use power tools.

So we can analyze the skill part of our assertion by asking questions such as these: How can we maximize the amount of skill needed to penetrate our network? How can we minimize the amount of skill needed to operate the defenses of our network? How can we prevent attackers from gaining knowledge of our network? Given that some exposure of our network information to the Internet is inevitable, how can we make that knowledge less useful to an attacker? How can we know that we are under attack?

By answering these questions, we can begin to see how we can use technological means to make our networks hard to penetrate. What makes a defense hard to penetrate? Answers to that might include: good design; good implementation of that design; proper configuration; a continuing supply of up-to-date patches; and multiple, redundant layers. Now we are getting to something that looks like technical specs.

You can try to ask for those qualities from the manufacturers of equipment you are considering for purchase, but you should also ask them some other important questions: How can you prove to me that your device has good design, good implementation, etc.? How often is your design and implementation audited by independent third parties? How often do you issue security patches? Do you issue patches on problem detection or on problem resolution? When was your device last compromised, how was it done, how did you find out about it, and how long did it take you to issue a patch?

Motivation. What motivates someone to attack our network? Well, for a start, the same things that motivate pretty much anybody to do anything—such as anger, greed, love, hate, revenge, and, of course, everyone's favorite: money. Add to that more intellectual reasons

such as curiosity and the thrill of a challenge, and we won't have any problem finding reasons why people attack us.

Motivation is a two-way street, of course. Attackers are driven by their own desires, but they can also be encouraged or discouraged by how they are treated by the network. Tease them with tantalizing clues and they will treat it as a game to be won. Answer each challenge with disappointment and confusion, and perhaps they will seek fun elsewhere.

Some questions that explore the issue of motivation are: What circumstances or resources cause people to want to attack us? Do our actions in defending the network increase or decrease a person's motivation to continue an attack? What would motivate people *not* to attack our network?

Counterattacking an intruder,
while tempting, is almost always a really bad idea.

There is a famous quote from Sun Tzu, the Chinese general of several millennia past: "The general who is truly a master of war can wage the entire war in the mind of the enemy and win without bloodshed." One way to do this is to create a situation in which potential attackers are unaware of your network. If it never occurs to them to attack you, and nothing disturbs their blissful ignorance of you, then they never will bother you.

This practice, called *security by obscurity*, actually does work—as long as you remain obscured. And therein lies the problem. Because as those of us who have Googled ourselves know, it can be very difficult to remain invisible

The Answer is 42 of Course

in today's world. Invisibility certainly doesn't hurt, but it is not sufficient. You must plan for the day that it fails.

I once saw a stand-up comic whose act consisted, in part, of convincing audience members to loan him something personal, like a wallet or purse, on some innocuous pretense, only to rifle through the contents in public to get laughs at the expense of that person. He managed to get a woman's purse and started pulling things out, but when the first thing he found turned out to be a .38 caliber police handgun, he very wisely and quickly replaced it, closed the purse, handed it back to the woman, and moved on to the next part of the act.

Creating barriers that convince potential intruders that it is really unwise to continue is an excellent way of deterring some attackers. Telling them they will be monitored is nice, but demonstrating in some way that they really are being monitored is even better. You can take this too far, of course. Counterattacking an intruder, while tempting, is almost always a really bad idea. Why? Well, besides legal and ethical reasons, there is a good chance that it will actually increase their motivation to attack you. It becomes personal!

Imagine how your defenses appear to an attacker. In fact, you should try them out yourself and learn first-hand. Do they seem like a video game, urging players to try for ever-higher scores? Or do they make you seem cold, implacable, and serious, with no sense of humor and a big team of detectives and lawyers just waiting for someone foolish to step in the trap?

Which of those images would make you want to try again? Which would make you want to move away quietly, never to return?

Opportunity. We want to minimize the opportunities that an attacker has to attempt a penetration. What does an opportunity look like? It could be as simple as a program accidentally left running on an Internet-exposed network port. Or it could be a whiteboard visible through a window, or a white paper that is a bit too honest about the network design. It could even be something as indirect as an executive who lets his or her children play games on a company-issued laptop. It could be the temporary but forgotten wireless hub set up for a

now-defunct project. It could be some well-known commercial software with a not-so-well-known flaw. It could be a poorly chosen encryption algorithm or a mailroom worker who slips a backup tape in his or her pocket before leaving for the day.

These are the questions we must ask to evaluate opportunity: Do we know all the ways in and out of our network? How can we be sure that we really know them all? How do we know that our network is built the way we think it is built? Have we closed off all unnecessary paths into our network? How do we know that we've succeeded in closing them? How do we know that they stay closed? What do we do if conditions change? If a new vulnerability is discovered in previously trustworthy software? If that software is deployed on 1,000 servers?

I remember reporting the results of a detailed security audit to a customer. I began by saying: "Your firewall here in the home office is not bad, but your firewall in Switzerland is terrible!" The president looked at the IT guys. The IT guys looked at each other, and finally somebody said, "What firewall in Switzerland?" It turned out that the company had merged with another company a year before and linked their networks. Nobody had ever noticed that there was a wide-open Internet connection that had never been locked down.

Measuring your network is not enough. You have to keep measuring it, as often as possible, to make sure that nothing has changed. Tools such as Nessus are freely available to do this, but remarkably few people use them to good advantage. Custom tools, if you can afford to create them, are even better, because public tools are available to your attackers as well, but private tools are known only to you and your team.

Which brings us to the most difficult topic of all: insiders. It's painful to consider, but the people with the most skill, motivation, and opportunity to attack your network are often working for you, and unhappy about some aspect of their jobs or their lives. Some useful questions are: Who do you trust, and with how much? Have those people trusted others with access that we don't know about? How would we detect if someone abused that trust? How can we limit damage from a misuse of

that trust? Do we have any current or ex-employees who can cause problems for our security? Who on our own staff knows enough about our network to be dangerous?

Dealing with the potential for insider attacks in a way that does not alienate those same insiders is tricky, but it can be done. One way is to limit the people with critical access to the smallest number possible, then be bluntly honest with those people and enlist them to help. One way is to have them watch each other. It sounds draconian, but it's used in a number of sensitive situations. For example, the military employs a two-man rule, or a no-lone-zone approach, in which certain functions must be performed by two authorized people, side by side.

Probably the best way to deal with this problem is to keep the people who can cause a security problem for you happy and engaged and aware of the potential for trouble—and the consequences of a breach. Use their creativity and good will to improve your security.

TYING IT ALL TOGETHER

Our second major goal, deciding how much effort to expend on any given security measure, suggests that no technical project lives in a vacuum. The political and financial ramifications of any technical decision often weigh heavily on what is actually built. So for any particular security measure, a network designer must consider questions such as these: How much are we willing to pay for this security measure? Will our user base accept the security measures we think are necessary? Do they believe these measures are necessary? Will the improvement in security justify the cost? How will we know that our security measures are really working?

One of the odd aspects of security work is that it can be very difficult to cost-justify if it is successful. If your car was not stolen last week and you had a steering wheel lock, is the lack of theft because of the lock, or because nobody was trying to steal it? It's tough to know unless we have some way of measuring how many times an attack was attempted or contemplated. Measuring the system and knowing if it is working, and if we are really under attack, helps us get some sense that we are getting what we paid for.

Consider the following questions:

- What metrics tell us that attacks have been attempted?
- What characterizes an attack?
- What differentiates real attacks from accidents or unintentional attacks?
- What is the false-positive rate of our detection mechanisms?
- Who or what is responsible for reading the logs?
- How does the detection of an attack feed into our security response strategy?
- How can we characterize attack types and rates over time? How can we measure trends in those characterizations to know if our defenses are working?
- How can we detect new types of attacks that will arise in the future? How can we add known attacks to our repertoire of detectable ones? What about future zero-day attacks, whose profiles we don't know now?

Asking these questions can be a painful process, but the time to ask them is up front, so that we can think about their answers and make our plans accordingly. Given all these complex questions, we might be tempted to go look for outside help to help sort through the issues. Some questions we should ask about security consultants: Are they playing the FUD (fear, uncertainty, and doubt) card? Do they offer guarantees or certainty? (If so, double-check their credentials.) Are we getting a real analysis of our network, or just the (possibly reformatted) output of off-the-shelf scanning tools? Do they offer to fix the problems they are diagnosing? If so, how do we know that they have told us all the problems they have found? Maybe they just listed the ones they can fix...

Why is security hard? Well, to paraphrase Albert Einstein, the questions generally remain the same, but the answers change all the time. Q

LOVE IT, HATE IT? LET US KNOW

feedback@acmqueue.com or www.acmqueue.com/forums

THOMAS A. WADLOW is a network and computer security consultant and author of *The Process of Network Security* (Addison-Wesley, 2000).

© 2005 ACM 1542-7730/05/0600 \$5.00