;login:

THE MAGAZINE OF USENIX & SAGE

June 2003 • volume 28 • number 3

inside:

USENIX & SAGE

The Advanced Computing Systems Association &
The System Administrators Guild

getting the problem statement right

by Dan Geer

Dan Geer is a USENIX Past President and is Chief Technology Officer at @Stake, Inc.

geer@world.std.com

All of us who even occasionally get to be engineers have at one time or another discovered that we were solving the wrong problem. Make that solved the wrong problem. To put it a little brusquely, it's about then we ask ourselves, "What good is the right answer to the wrong question?"

In science, sometimes the right answer to the wrong question means opportune discovery. In commercial engineering, the right answer to the wrong question just means inopportune overtime, opportunity cost, and/or negative rates of return. Science may be about invention or discovery, but the rest of the world is about execution.

Of course, there are a lot of situations where the problem statement is dictated by someone with a vantage point that bears no resemblance to the vantage point of those of us who have to execute. In my own line of work (security), "We need a firewall" almost always means something else, even if you can't discuss what the real problem statement ought to be until after the damned firewall goes live. Surely there are a raft of parallel examples.

Let's take a look at some problem statements that are not so obvious.

Among the techno-geek community, one sees pervasive antipathy to digital rights management (DRM) technology, while one sees just as pervasive affection for privacy enhancing technologies (PET). Emotionally, DRM equals the record industry equals profit enhancement equals badness, while PET equals cryptoanarchy equals self-actualization equals goodness. Yet the problem statement for both DRM and PET is one and the same, viz.: *controlled release of information you own at a distance in space and time*.

One can perhaps argue endlessly over whether you "own" a tune or whether you "own" your bank account number – "endless" is probably what any genuine argument is and ought to be – but if you accept the problem statement, then you have to conclude, at least with respect to the technology itself, that DRM = PET or, in outcome terms, with respect to privacy and digital rights we get both or we get neither. Arguing in favor of one and against the other is to argue for a solution space that is the null set. If you find this distasteful to your worldview, then dispute the problem statement or at least apply yourself to adaptive re-use of the available means to the ends you favor.

Let's try another one.

The intrusion detection paradigm says that you want to know when an attempt is made to cross a network perimeter from the outside of the company to the inside. Is that what you care about? An intrusion is definitionally the illegitimate *acquisition* of legitimate authority. Is that where the risk is? No, the risk is the illegitimate *use* of legitimate authority, which is precisely why just about everyone knowledgeable in security acknowledges that the biggest threats are on the inside. The problem statement for intrusion detection is: *keep dishonest people honest*.

In implementation terms, an intrusion system is about manning a guard station at the network perimeter of the firm. So, by analogy, which do you think is bigger: (A) the sum of US border protection manpower or (B) the sum of all the police departments in-country? Obviously, the correct answer is B, so the real problem statement is: *keep honest people honest*.

That is a high enough goal, trust me. Interdiction of the illegitimate acquisition of legitimate authority, i.e., the effort already put into intrusion systems, ought to grow

no more than it already has. Intrusion detection's sunk costs are just that, sunk costs, but that doesn't mean you have to keep sinking more. It is time to put more effort toward the behavioral analysis of surveillance data collected when operation is normal rather than signature analysis of intercepted data when operation is exceptional. Surveillance is consistent with keeping honest people honest even if the surveillance logs are only used forensically.

On the other hand, if you have all those intrusion systems creating masses of logs (that you never read), what can you do with them? The usual intrusion detection problem statement has an explicit subtext: *Never let anyone know that attackers tried to get in, did get in, how it was that they did get in, or what they did while they were in.*

This is just another variant on the most venerably stupid problem statement in all of security: *security through obscurity*.

The correct problem statement is: *threat identification and mitigation*.

It is not: *threat identification and hiding*.

In other words, share your logs with other firms like yours. If you fear debate in the boardroom over sharing this sort of data with selected peers, then here's how to win that debate in a single sentence: Unless you share your intrusion logs with like firms you will not and you cannot ever know whether you are a target of choice or a target of chance, and you will therefore waste needless cash or incur needless risk.

Let's try yet another one.

This one is harder and it doesn't have a solution until you stir in your own situation data. Many of the readers of ;*login:* have run systems at one time or another. Some do nothing but run systems. Let's take just the client side: Nothing is so easy to manage as systems that are, by design, identical and dataless. "Identical and dataless" is the right answer if you can get away with a problem statement (goal) like: *low cost to manage with minimum time to repair (local) failure*.

Of course, identicality means that a local failure can escalate to cascade failure rather easily; think Slammer with its 8.5-second doubling time. Unless you must simulate dumb customers or something like that, a better problem statement would be: *maximize net cumulative productivity*.

That's a problem statement which would naturally lead to operational strategies like: *All applications must be platform independent*; and *No OS can control a majority of platforms*.

You get the idea.

If all this sounds obvious, then great! It is obvious – just as obvious as the Emperor's absence of clothing. It is harder to do day-to-day. It is way too easy to say, "I have a hammer" and to conclude, "Let's find some nails to pound." What's hard is to think big enough to find a problem statement that is at once doable, elegantly simple, and which can be communicated to others who could care less what the hell you are talking about, but who "know" that they want a firewall, who "know" that they want to copy music but don't want you to copy their documents, who "know" that all the bad guys are outside and that that's a secret, and who "know" that if everything were exactly the same on every desktop the company would be better off because that's precisely how you get the best upfront purchase price. A little thought is a dangerous thing . . . especially in problem statements.

Unless you share your intrusion logs with like firms you will not and you cannot ever know whether you are a target of choice or a target of chance, and you will therefore waste needless cash or incur needless risk.