[in]security

by Gary McGraw

How Bad Is Intrusion Detection? : : :

Current network-based approaches to intrusion detection leave me cold, even if they're renamed "intrusion prevention" in an attempt to respin the marketing message. The main problems are that signature-based approaches are too easy for an attacker to skate right by, and anomaly-based approaches set off alarm bells too often to be useful. Part of this has to do with where the monitoring happens and what Intrusion Detection Systems (IDSs) are presently designed to look for. Today's IDSs are riddled with error and cause too many false positives, falling short of commercial expectations.

SIGNATURES AND ANOMALIES

The physical world uses intrusion detection concepts all over the place. Burglar alarm systems are a prime example—burglar breaks in, alarm sounds, authorities show up. Reactive yet effective. False alarms are sometimes a problem, but not a real showstopper. Car alarms, on the other hand, are much less effective. Cars that blare, blink, and beep all by themselves in the parking lot are so common that nobody seems overly alarmed by the alarms, just supremely annoyed. Today's IDSs are more like car alarms than burglar alarms.

> "Attackers have zillions of tricks for tweaking input streams to slip under the intrusion detection radar."

One problem is that these systems commonly rely on signatures of known attacks. As long as you know something specific about an attack, you can sift through network traffic or logs with a list of bad things to look for. Signature-based technology detects only known bad things, so it's easy to avoid. Attackers have zillions of tricks for tweaking input streams to slip under the intrusion detection radar. Previously unknown attacks are even better at avoiding detection, though they do require some actual work on the part of the attacker.

Less commonly encountered, anomaly-based intrusion detection relies on learning what normal system behavior looks like, then detecting anything that doesn't fit the model. Anomaly-based technology detects "not good" things, where "good" is defined by the model. Anything that doesn't look normal sets off the alarms. Ask any administrator whether users are "normal," and you'll quickly see the problem. In practice, anomaly-based systems have a very hard time separating novel but good from novel and not good.

All IDSs can be used to create a diversion. One very common attack technique is to cause an IDS to light up red in one area, while actually carrying out a clever attack elsewhere.

A signature-based system can't catch anyone who's using the latest attacks, and an anomaly-based system falls prey to the car alarm phenomenon, crying wolf over normal users who are just trying to get their work done. Because impeding real work tends to get security people fired, anomaly-based systems are almost never used. And because people tend to forget about things they can't see, feel, or taste, signature-based IDSs are fairly widely adopted despite their glaring shortcomings.

SAVING THE BABY

Keeping an eye out for trouble isn't a bad technique. The only questions are what to watch and how to watch it. One idea is to stop worrying so much about packets of data on the wire, and start worrying more about the behavior of applications that eat the data. This gets beyond a host-based approach by getting into the code itself. The guys who wrote the application your business depends on are supposed to know how it works and how it should behave. By using intrusion detection technology to monitor the things we know should and shouldn't happen inside the application itself, a much more interesting and useful paradigm emerges. Of course, this idea won't work for off-the-shelf software until ISVs adopt it, but it'll work fine for applications you build yourself. If intrusion detection is going to work, a new approach like this is necessary.

Gary McGraw is CTO of Cigital, a software quality management consultancy. He is co-author of *Exploiting Software* (Addison-Wesley, 2004), *Building Secure Software* (Addison-Wesley, 2001), and *Java Security* (Wiley, 1996). Reach him at gem@cigital.com.