Behavioral Analysis - Next Generation Threat Detection

by Lisa Vaas

Original intrusion detection solutions were based on whitelists / blacklists requiring extensive user customization and maintenance. Threat detection vendors today are, thankfully, moving to the far more proactive technologies associated with behavior analysis: an evolution to prevent malevolent guests from getting onto your networks and into your databases.

Malware, says Rob Rachwald, is weird. It has no manners. It's always sniffing around on the network. Before you know it, it's slipping off to connect back to an attacker, establishing a connection to your now-violated network. "That's not normal network behavior," Rachwald, senior director of research at FireEye, explained to SearchNetworking's Sally Johnson.

But it's exactly this sort of abnormal behavior that the next generation of information security technologies are using in order to identify potential data threats. FireEye, maker of malware detection technology, is just one of an increasing number of vendors trying to better tackle weird, and potentially malevolent, behavior, whether it comes over the web, through email, through mobile devices, or even, in the case of DB Networks, through SQL injection from a web application.

As it now stands, an intrusion detection system (IDS) that seeks out statistical anomalies can look at a range of factors when trying to figure out what's normal vs. what's abnormal behavior. At the network level, for example, an IDS can figure out pretty standard network activity by looking at metrics such as typical bandwidth use, traffic volume, which protocols are normally used, or which ports and devices generally connect to each other. An IDS looking for abnormal network traffic alerts an administrator when traffic strays from these normal parameters and instead starts acting funny, like that misbehaving malware.

Blacklist / Whitelist Flaws

Vendors' increasing tendency to sniff out weirdness through behavioral analysis makes perfect sense, says Eric M. Fiterman, founder of Spotkick. As it is, the traditional model of threat protection, based on barring entry via blacklists, has one big problem. Namely, during the time lag between brand-new bad stuff emerging, it being detected as such and, finally, the new signature being added to a database of blacklisted signatures, you've left the gate wide open for those new threats to successfully launch an attack.

"Where security companies, and the security industry at large, are at right now is the realization that it's easier to characterize intended behavior,"

©2013 • Reprinted with Permission

rather than trying to predict what new ways attackers are going to try to get past your defenses and keep your blacklists/whitelists up-to-the-minute, says Fiterman.

With blacklisting, Fiterman says, you're trying to keep up with what seems to be an infinite number of ways that attackers can get around your security defenses. "There are a thousand ways to walk around a wall," he says. "A lot of good people, or companies, are getting attacked, and [that can be] because they don't think like attackers. But it's harder to get engineers to think like criminals, who are very adept at accounting for every single vulnerability. When you build a system, it's easier to build a model or profile of how you expect it to be used than to catch every possible bad thing that can happen."

Network Behavioral Analysis

McAfee has thrown considerable heft behind behavioral analysis to address network threats. In 2012, the company decided to roll out a virtual, production-ready instance of its McAfee Network Threat Behavior Analysis (NTBA) to both new and existing McAfee Network Security Platform customers.

Tyler Carter, who heads up product marketing for McAfee's Network Security Program, in December went on to write that the modern threatscape is screaming out for improvements on the traditional security approach, which includes blacklists. "Network Threat Behavior Analysis gathers network flow information from all over your network, determines what is normal in your network, and then scrutinizes deviations from the norm that are likely to be dangerous," he wrote. "This becomes your sentinel on the wall, looking for targeted and stealthy attack behavior that just barely deviates from the norm — not significant enough for traditional defenses, but the only evidence that can prevent these attacks."

So, OK, McAfee's looking to improve network protection with behavioral analysis. But why stop there? Vendors aren't.

Antivirus Behavioral Analysis

Bitdefender, for its part, runs behavioral analysis in its antivirus offering. In May 2006, the company rolled out a new technology, now patented, called B-HAVE, intended to wean antivirus technology off its dependence on virus signatures by proactively detecting unknown threats. B-HAVE analyzes behavior in a virtual PC environment to figure out if an application will pull any of the tricks for which malware is infamous: modifying files, reading from or writing to a sensitive area of memory, or creating a file that's a product of a known virus, for example. Keeping the program at arm's length in this virtualized environment allows Bitdefender to vet the visitor. If the program behaves, without trying to pull a malicious stunt, the technology allows it to start normally. Otherwise, Bitdefender quarantines or deletes the app.

SQL Injection Threat

What's good treatment for networks and applications is just as good for one of the most persistent, always-at-OWASP's-Top-Ten list of the most critical web application security risks: SQL injection.

What is it with SQL injection? Why is this vulnerability so persistent? Can't companies just fix their code, already? How hard can it be?

Application rewrite

It can be very hard. As DB Networks brings up in a white paper, *SQL Injection Attack: Detection in a Web Application Environment*, when working to prevent or remediate SQL injection, enterprises strive to write or deploy highly secure code or to encrypt confidential data. That, unfortunately, isn't always possible. Sometimes the code in question was developed by a third party and isn't available to modify. Sometimes, fixing deployed code sucks up precious time and resources that need to be devoted to driving new business. Sometimes, there just isn't enough security domain expertise available in the developer pool, even if the application source code is available for fixing.

Web Application Firewalls

But why can't we rely on web application firewalls (WAFs) to detect SQL injection? Because they just don't do a good job at detecting this particular form of attack. The problem, says Dave Rosenberg, chief developer for DB Networks: the ubiquitous, ever-open port 443. If your enterprise has an important database fronting web applications, port 443 is wide open, because it has to be. All HTTPS URLs use port 443 by default. That's just normal. SQL flows through that open, expected port, most often via a web form nowadays, chugging along at its work of building various statements that access a SQL

database, passing them back to apps that need the data to do their jobs.

That makes for a web app that has to constantly build SQL queries from an input statement on a web form. If that web form wasn't built with scrupulous care, it's quite possible for attackers to inject SQL statements that misbehave, though they'll get interpreted as SQL right alongside polite, well-behaved SQL statements.

When that happens, anybody out in web land who has ample SQL skills and persistence can get right into the database, Rosenberg says. "It's a horrible problem," he says. "What are you going to do? [A successful SQL injection attack] can read anything. It can change anything. You'd think it's a problem that would have been solved long ago." And that's where behavior analysis comes in, this time targeted at advanced SQL injection attacks, this time specifically designed to protect the crown jewels that are the database.

Many companies attempt to address the SQL injection problem by using WAFs. The problem is, WAFs speak web input. They try to decipher web traffic to figure out if it's SQL injection, but SQL is in English.

Here's an example of how that language mismatch can get hairy: The first customer with which DB Networks deployed their behavioral analysis technology had previously placed a WAF in front of a very large database. The customer operates lots of large, public-facing stuff, Rosenberg says, but in the mix, they also have a small operation: an antiques business.

The minute they turned on the WAF, all antiques customers looking to buy a drop-leaf table found themselves blocked. Why? There's a rule set up pertaining to the word "table," and there's a word in that SQL query pertaining to the word "drop."

"Drop table" just doesn't seem like an innocent request to a WAF, Rosenberg says, given that the DROP statement enables all indexes, tables, and databases to easily be deleted or removed. And that, Rosenberg says, is why the web app firewall approach hasn't worked well: The technology just doesn't have enough context to understand what's actually going on in SQL language.

SQL Behavioral Analysis

Contrast that with what DB Networks is doing with behavioral analysis: the technology looks at the traffic between the web app and the database. It's interested because vulnerable apps may submit query statements, and somewhere in those statements there can be problems. Behavioral analysis technology is able to identify these rogue statements.

Again, it boils down to heuristics. Imagine a bunch of experts voting on whether a SQL statement should be accepted into the clan or not. The experts may have drastically differing opinions, but perhaps a majority of the experts vote to shun this particular statement each for their own reasons based on what they've learned as proper behavior.

Here a couple of the many SQL–behaviors DB Networks technology models:

Modeling what the app does at a fundamental level. Web apps build statements to talk to relational databases in SQL. That language has a number of rules. DB Networks has found it "incredibly helpful" to look at those little building blocks of SQL language. Just like English breaks down into words, so too does SQL break down into little semantic statements. That enables DB Networks to notice, for example, when an attacker has managed to get a rogue SQL fragment inserted within a proper SQL statement created by the web application.

Is X variation the kind of variation this app has shown before? This requires an understanding of the rules of language — the syntax, if you will. It requires understanding how all statements hang together in a tree-like pattern. That, again, is the behavioral model. At some point the technology says, Hey, is this SQL statement likely to have been actually created by the web app, or not? If not, it just might be flag-it time.

DB Networks initial customers came from the financial sector, Rosenberg says: businesses that need to check off the regulatory compliance boxes, of course, but primarily, the company's seeing uptake from those businesses who've been burned by putting their trust into old-school security technologies.

Will we see other vendors move toward grafting behavioral analysis onto their threat detection technologies? It's looking like that's the direction many are taking. Threat detection behavioral analysis is obviously a new and evolving technology. Look for the evolution to move our databases toward the day when, finally, SQL injection drops off the OWASP Top 10 list.

©2013 • Reprinted with Permission