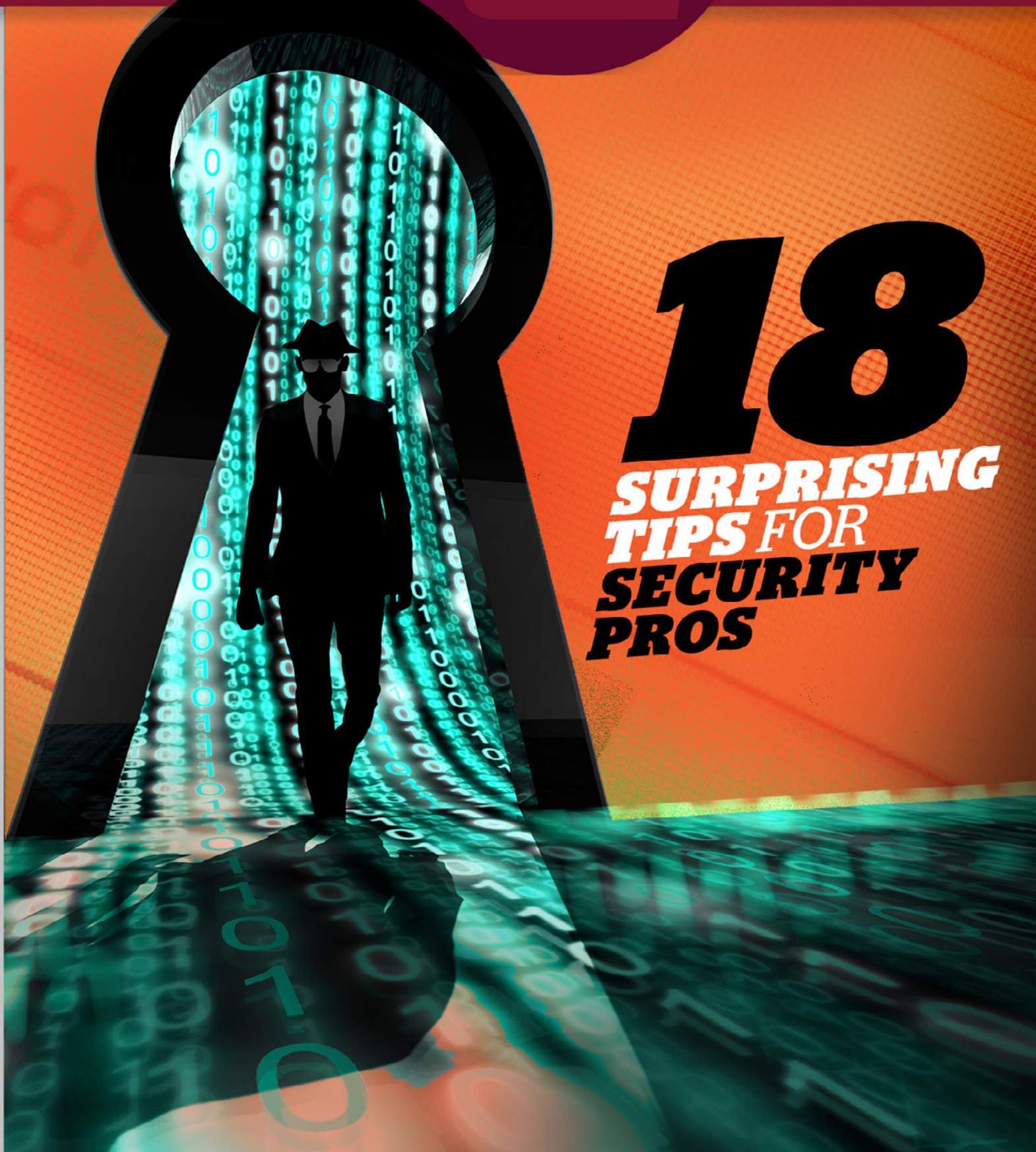
InfoWorld DeepDive
FROM IDG



18 SURPRISING TIPS FOR SECURITY PROS

THINKSTOCK

Deep Dive

9 popular IT security practices that just don't work

The security products and techniques you rely on most aren't keeping you as secure as you think. BY ROGER GRIMES

THINKSTOCK

Deep Dive

In IT security, FUD is more than just the tool of overhyping vendors; it's the reality that seasoned IT security pros live in, thanks in large part to the shortcomings of traditional approaches to securing IT systems and data.

Most common IT security products and techniques don't work as advertised, leaving us far more exposed to malicious code than we know. That's because traditional IT security takes a whack-a-mole approach to threats, leaving us to catch up with the next wave of innovative malware rolling out in plain view on the Internet.

In the vein of forewarned is forearmed, here are 9 common IT security practices and products that are *not* guarding your systems as well as you think.

Most common IT security products and techniques don't work as advertised, leaving us far more exposed to malicious code than we know.

Fail No. 1

Your antivirus scanner won't uncover real network killers

Back in the early 1990s, all-in-one programs, now known as antimalware scanners, could reliably detect every one of the *dozens* of viruses, worms, and Trojans in the wild. At the time, I volunteered for the PC Antivirus Research Foundation, started by Paul Ferguson, disassembling and testing newly found viruses. I remember everyone thinking antivirus programs had become so accurate and freely available that we all assumed computer viruses would be gone in a couple of years.

Boy, we were wrong. The professional bad guys now put out hundreds of thousands — if not millions — of new malware programs each month, far too many for any single antivirus program to reliably detect. This persists despite claims from nearly every antivirus vendor that they reliably detect 100 percent of the common malware submitted to them. Reality argues otherwise.

Every one of us is constantly faced with new malware that our antivirus engine doesn't detect. If you've ever submitted a malware sample to an engine checking site, like VirusTotal, you know it's fairly common for antivirus engines to miss new breakouts, sometimes for days. Weeks later, antivirus engines can still bypass a particular Trojan or worm.

Don't blame the vendors. With literally more bad files in existence than legitimate files, antivirus scanning is a tough job and begs for whitelisting programs. They have to store database signatures for hundreds of millions of devious, hidden programs and detect brand-new threats, for which there is no signature, all while not slowing down the protected host's operations.

Fail No. 2

Your firewalls provide little protection

As far as IT security is concerned, firewall protection is becoming even less relevant than antivirus scanners. Why? Because the majority of malware works by tricking users into running a forbidden program on their desktops, thus invalidating firewall protection. Moreover, the bad programs "dial home" using port 80 or 443, which is always open outbound on the firewall.

Most people are protected by multiple firewalls on the perimeter, on the desktop, and filtering applications. But all that bastion host-port isolation doesn't appear to be working. We're as exploited as ever.

Fail No. 3

Patching is no panacea

For many years the No. 1 security advice you could give anyone was to do perfect patching. All software has multiple vulnerabilities and must be patched. Despite the existence of more than a

Deep Dive

dozen patch management systems that promise perfect updates, it appears it can't be done.

Often it isn't the patch management software's fault — it's the managers. They only patch some items, but miss the most popular targets, such as Java, Adobe Reader, Flash, and more. Or they don't patch in a timely fashion. Or they don't follow up on why some percentage of their population doesn't take the latest applied patch, so there's always a vulnerable portion of users. Even in the best cases, getting patches out to the masses takes days to weeks, while the latest malware spreads across the Internet in minutes or hours.

Even worse, social engineering Trojans have essentially done away with that No. 1 advice. Consider this: If all software had zero vulnerabilities (that is, if you never had to patch), it would reduce malicious exploits by only 10 to 20 percent, according to most studies. If you got rid of the exploits that required unpatched software to be present, hackers relying on unpatched software would move to other strategies (read: social engineering), and the true reduction in cybercrime would be much less.

You know what works better than end-user education? More secure software and better default prompts.

THINKSTOCK

Fail No. 4

End-user education earns an F

Since the dawn of personal computing, we've warned users not to boot with a disk in their floppy drives, not to allow the unexpected macro to run, not to click on the unexpected file attachment, and not to run the unexpected antivirus cleaning program. Still, it does not work.

If our end-user education policies succeeded, we would have defeated hackers and malware by now. And if recent trends are any gauge, end-user awareness is worse than ever. Social-engineering Trojans are the biggest threat by far. Most end-users readily give up all privacy to any application or social media portal, and they do it without any thought of the repercussions, greatly increasing their likelihood of becoming a target and succumbing to social engineering.

I strongly fault the people behind most end-user education programs. In their hands, end-user education becomes a forced, unwanted childhood chore. Education is undertaken haphazardly, using spotty curriculum that usually doesn't contain information relevant to the latest attacks. If the No. 1 way end-users get tricked into running Trojans is through fake antivirus prompts, does your company tell your employees what their real antivirus program looks like? If not, why?

That type of disconnect puts IT systems in jeopardy. On average, it takes two years for the latest threats to show up in end-user education programs and only a minute for the bad guys to switch themes, putting us behind another two years.

You know what works better than end-user education? More secure software and better default prompts. Don't expect end-users to make the right decision; instead, decide for them. Macro viruses didn't go away until the default option was not to run the macro. File attachment viruses didn't minimize until most of them were blocked. Autorun USB worms didn't go away until Microsoft forced out a patch that disabled autorunning from USB keys as a default.

End-user education has never completely worked because it only takes one person, making one mistake, to infect your whole company. But you can reduce risk by producing better, more targeted end-user education.

Deep Dive

Fail No. 5

Password strength won't save you

Here's a frequently repeated security mantra: Create a strong password, one that is long, complex, and frequently changed. Never mind that users are famous for reusing passwords across multiple websites and security domains, for being tricked into typing log-on credentials into fake authentication prompts, and for giving their passwords to random emails. Heck, a large portion of the population will give out real passwords to strangers on the street for a few dollars. (This has been tested over many years, in various countries, by a variety of survey companies, and the results are shockingly the same.) Many of your end-users simply don't care as much about their passwords as you'd like.

The bigger problem now is that most hackers don't care either. They trick end-users into running Trojans, get admin access, harvest the password hashes, then reuse them. A password hash is a password hash, and one from a strong password looks and feels no different than one from a weak password.

Many of your end-users simply don't care as much about their passwords as you'd like.

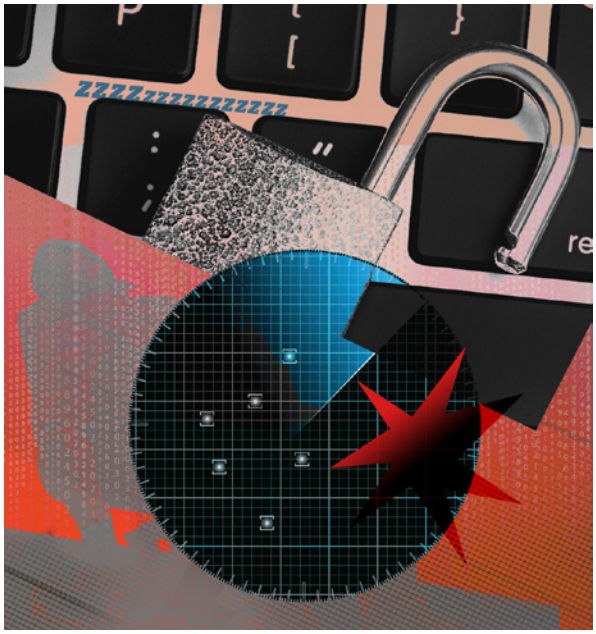
Fail No. 6

Intrusion detection systems can't determine intent

IDSes (intrusion detection systems) are the kind of security technology you want to believe in. Define a bunch of "attack" signatures, and if the IDS detects one of those strings or behaviors in your network traffic, it can proactively alert you or possibly stop the attack. But they simply don't work as advertised.

First, there's no way to put in all valid attack signatures needed to account for the malicious activity heaped on your enterprise. The best IDSes may contain hundreds of signatures, but tens of thousands of malicious attempts will hit your systems. You could add tens of thousands of signatures to your IDS, but that would slow down all monitored traffic to the point where it wouldn't be worth the effort. Plus, IDSes already put out so many false positives that all event alerts end up being treated like firewall logs: neglected and unread.

But the demise of the IDS is due to the fact that most bad guys piggyback on legitimate access. How can an IDS tell the difference between the CFO querying his financial database and a foreign attacker using the CFO's computer and access to do the same? They can't — there's no way to determine intent, which is needed to decide whether the network stream should create an alert or be passed as normal, operational business.



Fail No. 7

PKI is broken

Public Key Infrastructure is mathematically beautiful in every way. The problem is that many PKIs are hideously configured, woefully insecure, and mostly ignored, even when they function perfectly in the public sector.

Several legitimate public Certification Authorities have been horribly hacked in the past several years. They've allowed hackers to gain access to signing keys and to issue fraudu-

Deep Dive

lent keys for use by other hackers, malware, and possibly interested governments.

But even when PKI is perfect, people don't care. Most users, when warned by their browser that the presented digital certificate is untrusted, can't wait to click the Ignore button. They're happy to bypass the security inconvenience and get on with their computing lives.

Part of the problem is that the websites and programs using digital certificates have been lackadaisical in their use, allowing certificate error messages to become an everyday occurrence. Users who don't ignore digital certificate error messages aren't able to participate in a large segment of legitimate online life, sometimes including remote access to their own workplace systems. Browser vendors could enforce digital certificate errors so that any error, earned or mistaken, would result in the site or service not being presented, but customers would revolt and choose another browser. Instead, everyone blithely ignores our broken PKI system. On the whole, the masses don't care.

THINKSTOCK

Fail No. 8

Your appliances are an attacker's dream

The main benefit of appliances — increased security — hasn't panned out. By having a smaller OS footprint, appliances promise to be less exploitable than fully functional computers running traditional OSes. Yet, in more than 10 years of testing security appliances for InfoWorld, I've only once been sent an appliance that didn't contain a known public exploit. Appliances are nothing but operating systems on closed hard drives or firmware, and those designs are innately harder to keep patched.

In the end, appliances often contain just as many vulnerabilities as their software-only counterparts; they're just harder to update and usually aren't. Instead of being hardened security devices, they are an attacker's dream. I love doing penetration testing on environments with lots of appliances. It makes my life significantly easier.

Fail No. 9

Sandboxes provide a straight line to the underlying system

I sigh every time a new security sandbox is announced. These sandboxes are supposed to make exploits against the software they protect impossible or at least significantly harder to pull off. The reality is that every security sandbox developed so far has fallen under hacker attention. However, that doesn't stop the dreamers who think they'll find one that will halt all exploits and put down computer maliciousness forever.

Unfortunately, a lot of computer security is more security theater than protection. Your job is to pick through the myriad solutions and employ the ones that truly reduce risk. The security practices listed above are overhyped. How do you know? Because IT is implementing every one of them and malicious hacking and exploitation is more popular than ever. You can't ignore the facts. ■

Deep Dive

9 crazy IT security tricks that actually work

IT security threats are constantly evolving. It's time for IT security pros to get ingenious. BY ROGER GRIMES

Network and endpoint security may not strike you as the first places to scratch an experimental itch. After all, protecting the company's systems and data should call into question any action that introduces risk. But IT security threats constantly evolve, and sometimes you have to think outside the box to keep ahead of evildoers.

Charles Babbage, the father of the modern computer, once said, "Propose to a man any principle, or an instrument, however admirable, and you will observe the whole effort is directed to find a difficulty, a defect, or an impossibility in it. If you speak to him of a machine for peeling a potato, he will pronounce it impossible: If you peel a potato with it before his eyes, he will declare it useless, because it will not slice a pineapple."

THINKSTOCK

Deep Dive

The world of network security is no different. Offer a new means for IT defense, and expect to meet resistance. Yet, sometimes going against traditional thinking is the surest path to success.

In that vein, we offer 9 security ideas that have been — and in many cases still are — shunned as too offbeat to work but that function quite effectively in helping secure IT assets. The companies employing these methods don't care about arguing or placating naysayers. They see the results and know these methods work well.

Trick No. 1

Renaming admins

Renaming privileged accounts to something less obvious than "administrator" is often slammed as a wasteful, "security by obscurity" defense. But it works. If an attacker isn't already inside your network or host, there's little reason to believe they'll be able to discern the new names for privileged accounts. If they don't know the names, they can't mount successful password-guessing campaigns against them.

Even bigger bonus? Never in the history of automated malware — the campaigns usually mounted against workstations and servers — has an attack attempted to use anything but built-in account names. By renaming your privileged accounts, you defeat hackers and malware in one step. Plus, it's easier to monitor and alert on log-on attempts to the original privileged account names when they're no longer in use.

Renaming privileged accounts to something less obvious than "administrator" is often slammed as a wasteful, "security by obscurity" defense. But it works.

THINKSTOCK

Trick No. 2

Getting rid of admins

Or, why not just get rid of all wholesale privileged accounts: administrator, domain admin, enterprise admin — every account and group that has built-in, widespread, privileged permissions by default.

Most network administrators laugh and protest when this is suggested, the same response security experts got when they recommended local Administrator accounts be disabled on Windows computers. Then Microsoft followed this recommendation, disabling local Administrator accounts by default on every version of Windows starting with Vista/Server 2008. Lo and behold, hundreds of millions of computers later, the world hasn't come crashing down.

True, Windows still allows you to create an alternate Administrator account, but today's most aggressive computer security defenders recommend getting rid of all built-in privileged accounts, at least full-time. Still, many network admins see this as going a step too far, an overly draconian measure that won't work. Well, at least one Fortune 100 company has eliminated all built-in privileged accounts, and it's working great. The company presents no evidence of having been compromised by an APT (advanced persistent threat). And neither IT nor users are complaining about the lack of privileged access. Why would they? They aren't getting hacked.

Trick No. 3

Honeypots

Modern computer honeypots have been around since the days of Clifford Stoll's "The Cuckoo's Egg," and they still aren't as respected or widely adopted as they should be. A honeypot is any

Deep Dive

computer asset set up solely to be attacked. Honeypots have no production value. They sit and wait, and are monitored. When a hacker or malware touches them, they send an alert to an admin so that the touch can be investigated. They provide low noise and high value.

Shops that use honeypots get notified quickly of active attacks. In fact, nothing beats a honeypot for early warning — except for a bunch of honeypots, called a honeynet. Still, colleagues and customers are typically incredulous when I bring up honeypots. My response is always the same: Spend a day spinning one up and tell me how you feel about honeypots a month later. Sometimes the best thing you can do is to try one.

THINKSTOCK

Trick No. 4

Using nondefault ports

Another technique for minimizing security risk is to install services on nondefault ports. Like renaming privileged accounts, this security-by-obscurity tactic goes gangbusters. When zero-day, remote buffer overflow threats become weaponized by worms, computer viruses, and so on, they always — and only — go for the default ports. This is the case for SQL injection surfers, HTTP worms, SSH discoverers, and any other common remote advertising port.

Critics of this method say it's easy for a hacker to find where the default port has been moved, and this is true. All it takes is a port scanner or an application fingerprinter to identify the app running on the nondefault port. Most attacks, however, are automated using malware, which as stated, only go for default ports, and most hackers don't bother to look for nondefault ports. They find too much low-hanging fruit on default ports to be bothered with the extra effort.

Trick No. 5

Installing to custom directories

Another security-by-obscurity defense is to install applications to nondefault directories.

This one doesn't work as well as it used to, given that most attacks happen at the application file level today, but it still has value. Installing applications to custom directories reduces risk, because, once again, automated malware almost never looks anywhere but the default directories. If malware is able to exploit your system or application, it will try to manipulate the system or application by looking for default directories. Install your OS or application to a nonstandard directory and you screw up its coding.

On many of my honeypots, I install the OS to nondefault folders — say, in C:/Win7 instead of C:/Windows. I usually create the "fake" folders that mimic the real ones. When my computers get attacked, it's easy to find complete and isolated copies of the malware hanging out in the C:/Windows/System32 folder.

Changing default folders doesn't have as much bang for the buck as the other techniques mentioned here, but it fools a ton of malware, and that means reduced risk.

Trick No. 6

Tarpits

My first experience with a tarpit product was LaBrea Tarpit. It was developed during the outbreak of the Code Red IIS worm of 2001. Worms readily replicate to any system that matches their exploit capabilities. LaBrea worked by answering connection attempts for addresses not already assigned to legitimate machines. It

Deep Dive

would then answer and tell the worm to connect, then spend the rest of the time trying to slow down the worm, using various TCP protocol tricks: long timeouts, multiple retransmissions, and so on.

Today, many networks (and honeypots) have tarpit functionality, which answers for any nonvalid connection attempt. When I penetration-test these networks, my attacks and network sweep scanning attacks slow to a crawl — they're unusable, which is exactly the purpose. The only downside: Tarpits can cause problems with legitimate services if the tarpits answer prematurely because the legitimate server responded slowly. Remember to fine-tune the tarpit to avoid these false positives and enjoy the benefits.

THINKSTOCK

Trick No. 7

Network traffic flow analysis

With foreign hackers abounding, one of the best ways to discover massive data theft is through network traffic flow analysis. Free and commercial software is available to map your network flows and establish baselines for what should be going where. That way, if you see hundreds of gigabytes of data suddenly and unexpectedly heading offshore, you can investigate. Most of the APT attacks I've investigated would have been recognized months earlier if the victim had an idea of what data should have been going where and when.

Trick No. 8

Disabling Internet browsing on servers

Most computer risk is incurred by users' actions on the Internet. Organizations that disable Internet browsing or all Internet access on servers that don't need the connections significantly reduce that server's risk to maliciousness. You don't want bored admins picking up their email and posting to social networking sites while they're waiting for a patch to download. Instead, block what isn't needed.

Trick No. 9

Security-minded development

Any organization producing custom code should integrate security practices into its development process — ensuring that code security will be reviewed and built in from day one in any coding project. Doing so absolutely will reduce the risk of exploitation in your environment.

This practice differs from educator to educator, but often includes the following tenets: use of secure programming languages; avoidance of knowingly insecure programming functions; code review; penetration testing; and a laundry list of other best practices aimed at reducing the likelihood of producing security bug-ridden code. ■