Simple Network Vulnerability Testing

By Eric Uner

IT administrators have become too dependent on perimeter defenses. Instead, they should regularly run vulnerability scans against their networks. This article addresses the importance of testing and analyzing your network for security.

I recently met with an upper level bank IT executive who told me that their security is superior to all their competitors. I asked a simple question, "How do you know?" The silence that followed was less than reassuring. Like all too many victims (and future victims) of cyber-crimes, this executive's organization has left security testing of their network to the hacker community.

Vulnerability analysis of your network is too simple and inexpensive to ignore, especially when considering the potential damage of a security breach. Although there are literally hundreds of techniques that hackers can use against you, I have found that almost all of their tactics are rooted in a single fatal design flaw that almost every network architect makes—an over-reliance on perimeter defenses. Combined with a simple lack of testing, this flaw has repeatedly allowed hackers into every part of corporate networks, from Web servers to databases to desktops.

ANALYZE YOUR ARCHITECTURE

If you take nothing else away from this article, please, take this advice: Do not become over-dependent on your perimeter defenses. As illustrated in Figure 1, almost all IT environments in the world include the prolific deployment of perimeter defense devices such as firewalls, intrusion detection systems, virus scanners, and so on.

With the current situation in the Middle East, historians with expertise in French military history seem to be coming out of the woodwork. Let us turn our attention to the infamous Maginot Line. Named for Andre Maginot, the French constructed the Maginot Line in 1929 as a perimeter to protect against the invading German army. With all focus on this perimeter, the inside of France was virtually defenseless. France fell in just 35 days due to their reliance on this model.

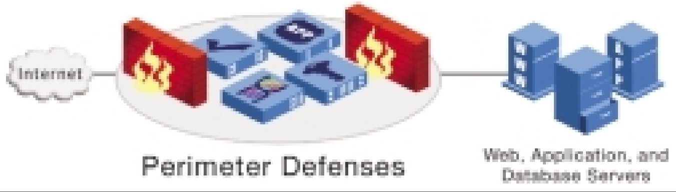
The flaw in the reasoning behind perimeter defenses is that they are usually violated or circumvented. Achilles had his heel, the French had the Maginot Line, Helm's Deep had that little water gate (slightly obscure reference for you Tolkien fans), and IT has the firewall and dozens of other ancillary devices and appliances. The result is always the same.

To avoid this pitfall, simply assume that your perimeter will be violated, prepare yourself for the intrusion, and look for servers that incorporate the latest in host-based security. Host-based security means that each device is responsible for itself, and assumes every other computer on the network is a launch point for an attack. Virus scanners are my favorite example. Many users assume that their e-mail must be safe since their e-mail server incorporates a virus scanner. This assumption is very wrong.

As illustrated in Figure 2, Server-based virus scanners cannot detect ciphered viruses (such as in a PGP encrypted e-mail sent directly to

FIGURE 1: TYPICAL WEB ARCHITECTURE

The typical architecture deploys a myriad of perimeter defenses around anything exposed to the Internet. Unfortunately, these defenses are usually not completely effective.



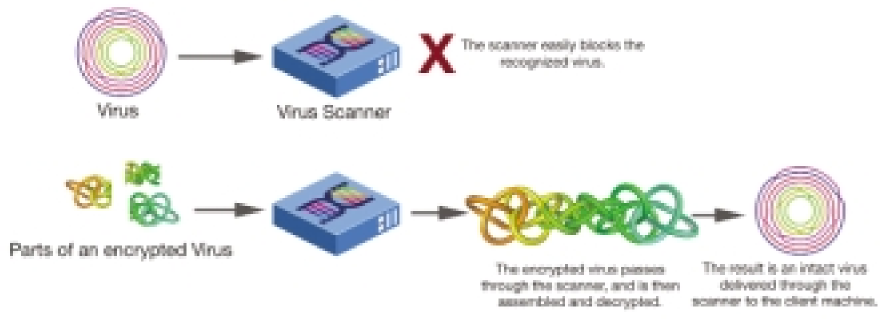
you), and many miss multi-part viruses sent from disparate sources in several messages. This is the reason for desktop virus scanners—one of the best and oldest examples of host-based security. Since only the computer opening the complete, deciphered message is guaranteed to have access to all relevant data, only that computer is truly suited for virus scanning.

Similar in strategy to virus scanners, XML scanners are all the rage and the latest entry into IT buzzword bingo right behind Web services. Web services use protocols that operate on ports designed to go through the perimeter and right to the Web server. Now what? The perimeter defense is actually circumvented by design in this new technology. Although Web service security standards are in their infancy, it is clear that the ultimate defense against attacks in a Web services environment will be up to the presentation and application layers that ultimately process the XML. The defense will not be left to any perimeter device.

Not only are there new services designed to punch the perimeter, but guess what—your perimeter has already been violated by you. In addition, the person down the hall and in the cubicle next door has violated the perimeter. Many organizations with strong perimeters ignore the insider threat.

FIGURE 2: VIRUS SCANNING

Ciphered viruses can pass right through even the best virus scanners, no matter how simple the virus is.


Virus
Virus Scanner
The scanner easily blocks the recognized virus.
Parts of an encrypted Virus
The encrypted virus passes through the scanner, and is then assembled and decrypted.
The result is an intact virus delivered through the scanner to the client machine.

REGULARLY USE AUTOMATED TEST TOOLS

Enough about perimeters—keep those perimeter devices around if you must, but regardless of their presence, IT administrators should regularly run vulnerability scans against their networks. Sample scanners include the free tools Nmap (http://www.insecure.org), Nessus (http://www.nessus.org) and SARA (http://www-arc.com/sara/) or commercial products such as VIGILANTe's SecureScan (http://www.vigilante.com/) or ISS's Internet Scanner (http://www.iss.com). These tools scan networks for known vulnerabilities. There is, of course, a right way and a wrong way to use these tools.

First, be sure to run the tests against individual devices as well as the vendor's typically advertised system-wide network test. The reasoning here is to avoid falsely good results. The first time our security staff ran Nessus against our network, they decided to run it outside the firewall and see what damage an Internet user could do. The results were wonderful—the firewall was blocking all hacking attempts. Skeptical, one on the staff decided to go to the Nessus Web site and get the latest test suites. Trouble was, no one could get out to the Internet. The first test had brought the firewall down, and no traffic (good or bad) was passing through it. Further testing from the inside brought down routers, Web servers, e-mail servers, and in general wreaked havoc throughout our environment.

Only by running the test tools against each device on your network one at a time will you be able to create an accurate list of your trouble points. This takes less time than you would think. A single administrator walking around with a laptop, a small, inexpensive switch, and a scanner like Nessus can produce preliminary reports on even large corporate networks in just a couple of days. Of course, finding those couple of days may be difficult, especially since testing can interfere with normal business operations (as it did in my example). The choice that your company's risk manager must make is simple: is it more expensive to suffer from an unanticipated real attack, or to recover from a planned, simulated attack. The choice seems obvious.

Another popular vulnerability scanning mechanism is the use of outside firms to run assessments over the Internet against your network at regular intervals. There are many free services like those from Sygate (http://scan.sygate.com/) as well as commercial subscription services like those from SonicWALL (http://www.sonicwall.com). Though these scanners usually find only the most obvious vulnerabilities, and cannot scan the interior of your network for insider threat detection, they still offer some powerful advantages. Even the commercial services are relatively inexpensive, usually under $2,000 USD for a whole year's worth of testing and reporting. The most powerful advantage of a subscription-based service, however, is that it gets done. I conducted an informal survey that sampled from 70 plus companies with whom I have worked, and asked their security manager about the last security scan they conducted. The answers were similar, and usually sounded something like, "Scanning? I think Larry might be doing it now," or my favorite, "Thanks for reminding me." Regular scanning requires time and discipline, two elements that are rare in the fast-paced IT industry. Subscription services can ensure the job is done, and at the very least, the major vulnerabilities are reported.

I am often asked about exposing your network to white- or black-hat hackers. These folks will happily penetrate your network, then (upon payment) report the results to you so that you may close the vulnerabilities that they found. While the hacker community can be a good resource, reaching out to them is not usually beneficial. Believe me when I say that the hacker you have to worry about does not go to a hacking convention in Vegas and wear a "Hello My Name Is" badge. There is a general misconception that anyone who proclaims to be a hacker must be better than the automated tools at finding vulnerabilities, but this is generally not the case. Furthermore, if black-hat hackers become frustrated with challenges beyond their skill set, they often begin to attack your partners or ancillary services that may be more vulnerable. Professional teams, therefore, like those from Verisign (http://www.verisign.com/) or from product vendors (for example, Oracle offers Database Security Consulting) often offer the same services without the accompanying headaches.

DO NOT IGNORE SOCIAL ENGINEERING ATTACKS

I find it odd that companies would deploy Web cloaking software and appliances that disguise which Web server and operating system they use. These companies then post career opportunities for people with experience in those systems on Monster.com. A form of hacking called social engineering exploits such blunders.

Social engineering is the most intriguing hacking technique, and a very effective tool for cyber-criminals. The goal of a social engineering attack is to trick people, usually by phone, into revealing information that the hacker may use to compromise the security of the target's network. Rather than software vulnerabilities, social engineering relies on wetware (the hacker term for the human nervous system). Wetware vulnerabilities are more common than you would think. A college professor once challenged me and my fellow students to gain access to one of the university's restricted computer systems. Of course he knew many would get in, but the key was how fast and how traceable their intrusion was. When class was over, I walked right over to a campus telephone, called the support desk, and said, "I am in the lab here with Professor [name omitted], and he needs his password reset." The woman on the telephone happily obliged and gave me my professor's new password. In two minutes, a social engineering attack had both granted me access and denied access by an authorized user.

Proper training and policy is key to avoiding social engineering vulnerabilities. IT managers should set a policy of "information push" whereby all staff are prohibited from discussing IT issues of any kind with anyone unless the staff either initiated the conversation or can verify the identity of the other party. A recent attack against Intel's NetStructure 7110 E-Commerce Accelerator ultimately involved hackers calling companies that deployed these units, and posing as support staff. The hackers asked for seemingly harmless information (the MAC address of the server), and administrators easily gave up the information. Had the IT staff been trained to simply take the request down, then look in a file for an appropriate contact to call back, the hack would not have been successful. In this case, they would be the *source*, not the *destination* of the call.

Testing for social engineering vulnerabilities is actually fun. I am going to make a leap of faith and assume the reader has some friends, and those friends may even be in a similar business or at least in the IT field. Be they inside or outside your company, solicit them to call in and ask for the name of the IT manager (which could be used to impersonate that person), the operating system version they use on the Web server—anything at all.

ANALYZE AND ACT ON RESULTS

So you looked at your architecture, you ran all the scanners you could find, and had some friends try and get your name and e-mail address—now what? All this information is worthless without action. The first step is to consolidate all information into a report, and sit down with as many of your peers as you can to review the data. In my experience, at least half of your data is likely to be invalid—but that's OK. Scanners produce many false positives and negatives. Case in point, Nessus, when run against my company's primary Web server, indicated that the server is running multiple versions of an outdated Windows Web server. The server, however, does not run Windows, Linux, or an operating system of any kind. The report, therefore, was a case where Nessus was confused by how random the HTTP replies and TCP values were.

After sifting through the vulnerabilities that can be verified, I suggest submitting them to a tracking tool where you may assign a resource to fix the problem and a due date. I use Double Choco Latte, a GNU Enterprise package that provides basic project management capabilities (free from http://dcl.sourceforge.net/). DCL or any tracking tool will help organize your list of repairs, and can provide instant reports on the number of remaining vulnerabilities for your next security report. Since patches and updates often affect operational capability, using a tracking tool allows the rest of your staff to see your plan before it is executed, and point out any changes that may affect them.

NOW GO TO IT!

Start by looking at your security architecture, test against its goals, make the appropriate changes and rinse-repeat. The IT security issue is complex, certainly many books exist on the subject, and I recommend further reading at your local bookstore. My hope for this article is that you gained at least one factoid, link, or idea that helps. I can be contacted at uner@bodacion.com, and I'd love to hear your success story.

Eric Uner started Bodacion Technologies and is the chief software architect there.