

Intrusion Detection

Contributed by LE Webmaster
Wednesday, 26 January 2005

If someone broke into your network, how would you know? There wouldn't be any muddy footprints. If you had a strong firewall that had good logging capabilities, you might find evidence of an attack in your logs, but a smart hacker could even get around that. To make the case for rigorous intrusion detection beyond that provided by firewalls and their logs, consider the case of a classic e-mail virus: A worker receives an e-mail from a coworker's home account saying that he's found a copy of a file that's been missing for a few months. The worker clicks on the executable attachment that says it's a zip file, which installs a Trojan horse that lies in wait until it detects a period of keyboard and mouse inactivity for long enough to assume that the worker isn't looking at the computer. The Trojan horse then opens a connection to a hacker's computer. Even if your firewall is designed to block outbound connections on unusual ports (the vast majority are not), nothing prevents the hacker from serving his attack software on a common port like 80 (HTTP). Your firewall will merely see what looks like an HTTP connection flowing out of the network to a web server, a type of connection it sees thousands of times a month.

This sort of attack will get right past even a strongly secured stateful inspection firewall like Firewall-1 or SonicWALL. Only proxy-based firewalls like Gauntlet and Symantec Enterprise Firewall can be relied upon to reject improper protocol data on standard ports. Even in that case, a clever hacker will simply use a binary data port like FTP that can only be filtered for initial connection data; the true binary file data cannot be filtered because there's no way to predict what the file should contain. The hacker designs the Trojan horse and attack server to transmit fake session establishment data, while the client appears to be merely uploading a file, but is in fact uploading screen images and accepting mouse and keyboard input. A well-designed Trojan horse could even work through an FTP proxy. Any other binary protocol could also be exploited. If you rely upon firewall logs to tell you when an intrusion has occurred, you'll never find this sort of attack because it will appear to the firewall as if it were a regular client-initiated FTP upload session. Nothing about it will set off any triggers or alarms. So we've established that even the strongest firewalls cannot prevent certain attacks. Any useful connection to the Internet is a potential vector for attack.

Direct Intrusion Before we discuss TCP/IP and Application layer intrusion detection, it's important to understand that intrusion takes many forms at many other layers in your network. Direct intrusion, where someone gains physical access to your facility and sets the stage for further networked intrusion, is a rare but important security problem that must be addressed to achieve holistic security. The attacks in this section are exceedingly rare; most companies need not worry seriously about physical security. But if your company performs any research and development activity, then you should use more stringent security policy to protect the product of your research. Real intrusion prevention begins with premises security, Physical layer security, and Data Link layer security. If your network is so fortified against Internet attack that a dedicated enemy cannot breach your defenses, they will change tactics and intrude more directly. Possible vectors for attack include:- Impersonating an employee- Impersonating service personnel- Wiretapping public data links- Adding devices to the network- Outright theft

Do you know everyone who works at your company? You don't unless you work at a small business. Does your company issue ID badges that everyone wears? They probably do not if you work at a small business. Employee impersonation is particularly risky, especially in medium-sized businesses-attacks of this sort are extremely rare. Are new employees subjected to a background check? Although it as rare as any of the attacks in this section (and more frequently the subject of movies than reality), if your organization had secrets worth more than \$50,000 to steal, it becomes worth the effort for an intruder to simply be hired in order to gain access. Impersonating service personnel is the easiest way to gain trusted access to a company. If a phone repairman walked in and told your receptionist or security guard that they were experiencing telephone problems in the building, would that receptionist or security guard call to

verify their story or would they simply escort them to the wiring closet? Would they know the difference between the attachment of a legitimate bit error rate tester (BERT) to a T1 line and an illegitimate wireless bridge? If a salesman showed up and offered and demonstrated a new laptop, and said his company would be willing to let your staff evaluate the device for a month at no charge, would you accept? If you hired a security expert to evaluate your network, would you bother checking her credentials? I've won a number of contracts to evaluate network security based on my experience and the fact that I've written a number of security-related articles but I've never had a customer check my driver's license to see if I was actually who I said I was. For some reason, companies go to reasonable effort to check out employees, but they let contractors and consultants parade around the company without so much as a look at their personal identification. If you fired an IT staffer, are you certain that he hasn't embedded a Trojan horse or opened a back door somewhere? Did you change every password in every device that the staffer had access to? This attack is by far the most common of those discussed in this section, and by far the most damaging because the attacker has intimate knowledge of your architecture, methods, and weaknesses. Any of these examples of lax facility security could lead to a network intrusion. A minute alone with a firewall is long enough to modify the policy to allow a surreptitious service port entrance for further exploits, or to change the policy for an existing service. The policy abstraction allowed by modern firewalls is nice, but nothing prevents a hacker from creating a service called SMTP on port 5900 that actually accepts VNC (remote control software) connections. All you'd see in your rule base is that SMTP allows inbound connections; you'd have to dig to find out that that SMTP wasn't SMTP at all. Intrusion Tools and the Techniques Hackers use a variety of tools and techniques to attack networks. A typical intrusion takes the following form, assuming that the intruder begins with no information about your site other than its address and lately, not even that. A constant barrage of address and port scans reveal hackers rummaging through the Internet looking for targets of opportunity. When our company recently installed a firewall on a newly provisioned, never-before-used IP address, it took only seven minutes to log and drop its first hostile port scan. Slashdot.org reports default IIS and Linux installations being compromised routinely within minutes of being exposed to the Internet without protection. You can no longer count on obscurity as any sort of security. Hacking attempts usually proceed as follows:

1. IP address scans
2. Port scans
3. Services evaluation
4. Target selection
5. Vulnerability probes
6. Automated password attacks
7. Application-specific attacks

Each of these attacks is detailed in the following sections:

- Address Scans** Scan across the network range, if any, to find service hosts. Hackers usually scan at least the entire range of IP addresses around your host and may use reverse DNS lookup to determine if those other hosts are registered to your company. For this reason, you should assume they'd find any public hosts you have on the Internet, even if you didn't publicize its address.
- Port Scans** Scan across responding hosts to find running services. This information tells the hacker what services are running on each publicly reachable host. Port scans typically work through firewalls as long as a host can be reached, especially if the scan is limited to service ports like 21 and 80 rather than scanning across all ports (which some firewalls are capable of detecting immediately and blocking on).
- Services Evaluation** Determine the operating system type of each host. After probing common service ports like Echo, Chargen, FTP, Telnet, SMTP, DNS, HTTP, POP, NNTP, RPC locator service, NetBIOS, NFS, etc., the hacker will determine what operating system each host appears to be running. Windows-based hosts typically respond on NetBIOS ports but do not respond on Telnet, whereas Unix hosts respond on Telnet but not on the RPC Locator service used by Windows NT. Linux hosts in their default configurations respond on a wide array of services and are easy to spot for that reason. It's a simple matter for any one of a number of text responding services like Telnet, FTP, HTTP, SMTP, or POP to receive a service banner indicating which specific application and version is providing the service. Since most applications have an affinity for certain operating systems, determining the operating system is trivial.
- Target Selection** Selects the weakest found host. Hackers will usually target the host with the most running services, in the assumption

that little to no work has gone into securing that host's default configuration. Windows hosts that respond on port 139 (NetBIOS) are certain to be attacked, since exploiting that service can lead to full control of the machine. Other services, like Terminal Services, VNC, pcAnywhere, or other broad-spectrum services that provide remote control are popular targets for attack.

Service-Specific Probes Uses vulnerability analysis tools like SATAN against Unix systems or the Internet Scanner from Internet Security Systems for Windows hosts. These probes check for a wide range of known service vulnerabilities that are easy to exploit, so they're checked first.

Automated Password Attacks Used against services like FTP, HTTP, NetBIOS, VNC, or others that allow access to the file system or a remote console. Hackers employ software specifically written to perform a high rate of logon attempts (like the NetBIOS auditing tool) using dictionaries of common passwords. Failing this attack, most hackers will concede defeat or resort to simple denial-of-service attacks if they hold a grudge against you.

Warning: VNC, the popular free remote control program, is especially susceptible to automated attacks. First, it typically installs on a unique and easily scanned address. Secondly, it is shielded only by a single password, not by a user account and password. Finally, all versions prior to 3.3r7 respond immediately to failed logins and do not lock out after numerous attempts. Hackers have created high-speed password crackers for VNC that can gain access to machines exposing the service in short order.

Intrusion Detection Systems Intrusion detection systems (IDS), also known as intrusion detectors, are software systems that detect intrusions to your network based on a number of telltale signs. Active response systems attempt to either block attacks, respond with countermeasures, or at least alert administrators while the attack progresses. Passive IDS systems merely log the intrusion or create audit trails that are apparent after the attack has succeeded. While passive systems may seem lackluster and somewhat useless, there are a number of intrusion indicators that are only apparent after an intrusion has taken place. For example, if a disgruntled network administrator for your network decided to attack, he'd have all the keys and passwords necessary to log right in. No active response system would alert on anything. Passive IDS systems can still detect the changes that administrator makes to system files, deletions, or whatever mischief has been caused.

Inspection-Based Intrusion Detectors Inspection-based intrusion detectors are the most common type. These intrusion detectors observe the activity on a host or network and make judgments about whether an intrusion is occurring or has occurred, based either on programmed rules or on historical indications of normal use. The intrusion detectors built into firewalls and operating systems, as well as most commercially available independent intrusion detectors, are inspection based. Intrusion detectors rely upon indications of inappropriate use. These indicators include:

1. Network traffic, like ICMP scans, port scans, or attachment to unauthorized ports.
2. Resource utilization, such as CPU, RAM, or Network I/O surges at unexpected times. (This can indicate an automated attack against the network.)
3. File activity, including newly created files, modifications to system files.

Intrusion detectors monitor various combinations of those telltale signs and create log entries. The body of these log entries is called an audit trail, which consists of the sum of observed parameters for a given access object like a user account or a source IP address. Intrusion detection systems can monitor the audit trails to determine when intrusions occur. Intrusion detection systems include these variations:

Rule Based Intrusion detectors that detect intrusion based on sequences of user activities (called rules) that are known to indicate intrusion attempts, such as port scans, system file modifications, or connections to certain ports. The majority of intrusion detection systems are rule based. Rule-based intrusion detection systems cannot detect intrusions outside the realm of their programmed rules and are therefore usually ineffective against new types of attacks until they've been updated.

Statistical Intrusion detectors that detect intrusion by comparing the existing base of valid audit trails to each new audit trail. Audit trails that differ substantially from the norm are flagged as probable intrusion attempts. Systems like these have the potential to detect hitherto unknown intrusion methods, but may miss rather obvious intrusions that might appear to be normal usage.

Hybrid Intrusion

detection systems that provide the best of both worlds by combining statistical and rule-based detection systems. Some of these systems are capable of creating new permanent rules from detected intrusions to prevent the intrusion from happening again without the overhead of statistical analysis. IDS systems always require system resources to operate. Network IDS systems usually run on firewalls or dedicated computers; this usually isn't a problem because resources are available. Host-based IDS systems designed to protect servers can be a serious impediment, however. Rule-based IDS systems can only detect known intrusion vectors, so all possible intrusions cannot be detected. Statistical intrusion detectors stand a better chance of detecting unknown intrusion vectors, but they cannot be proven to detect them until after the fact. Because of these limitations, IDS systems generally require monitoring by human security administrators to be effective. Countermeasure technology and response systems that temporarily increase the host's security posture during attacks are all in the theoretical research stage. Current IDS systems rely upon alerting human administrators to the presence of an attack, which makes human administrators an active part of the intrusion detection system.

Decoy Intrusion Detectors

Decoy intrusion detectors, also called honeypots, operate by mimicking the expressive behavior of a target system, but rather than providing an intrusion vector for the attacker, they alarm on any use at all. Decoys look just like a real target that hasn't been properly secured. Because the decoy is not normally used by anyone within your organization for any legitimate purpose, any connection to it at all is an intrusion attempt. When hackers attack a network, they perform a fairly methodical series of well-known attacks like address range scans and port scans to determine which hosts are available and which services those hosts provide. By providing decoy hosts or services, you can seduce the hacker into attacking a host or service that isn't important to you and which is designed to alert on any use at all. Decoys may operate as a single decoy service on an operative host, a range of decoy services on an operative host, a decoy host, or an entire decoy network. Decoy networks are very rare. Most decoy software runs on an operative host. You can establish an effective decoy host by installing a real running copy of the operating system of your choice on a computer with all normal services active. Using your firewall's Network Address Translation, send all access to your public domain name to the decoy machine by default. Then add rules to move specific ports to your other service computers; for example, translate port 80 only to your actual web server. When a hacker scans your site, he'll see all the services provided by your decoy host plus the services you actually provide on your Internet servers, as if they all came from the same machine. Because the services running on the decoy host include services that are easy to attack, like the NetBIOS or NFS ports, the hacker will be immediately attracted to them. You can then set up alerts to alarm on any access to those services using the operating system's built-in tools. You'll be secure in the knowledge that if the hacker intrudes into the system, he'll be on a system that contains no proprietary information. You can then let the attack progress to identify the methods the attacker uses to intrude into your system. I suggest installing a network monitor (like the one that comes with Windows NT) on the decoy host so you can keep logs of specific packet-based attacks as well. Decoy hosts are highly secure because they shunt actual attacks away from your service hosts and to hosts that will satisfy the hacker's thirst for conquest, giving you plenty of time to respond to the attack. The hacker will be thrilled that he was able to break into a system, and will be completely unaware of the fact that he's not on your real Internet server until he browses around for a while. You might even consider creating a bogus "cleaned" copy of your website on the decoy server to maintain the illusion in the hacker's mind that the actual site has been penetrated. Any desecration performed on the decoy site won't show up on your actual site. Best of all, decoy intrusion detection costs only as much as a copy of the operating system (Windows Workstations can be used to decoy for Windows Server, Linux can mimic any professional Unix server), target hardware, and your existing firewall. You won't have to pay for esoteric software. ~Article by Jerom written for Linux Exposed