Understanding Ethical Hacking

Whilst "black hat" hackers need only find one vulnerability to exploit, an ethical hacker must find and document them all, and it's well worth having this expertise in-house.

By David Norfolk
Freelance Journalist

In a computer-dependent society, breaking through anybody's perimeter security is seriously anti-social - and arguments that intruders are merely curious, harmless e-tourists don't hold water. At the least, all intruders are a threat to computer availability, because systems are hard enough to keep up when used as intended, let alone under untested conditions such as attempts at intrusion. Some people suggest employing "reformed hackers" to test perimeter security. Well, people do reform, but anyone who has been actively hacking systems in the last few years (when the anti-social nature of hacking will have been apparent to anyone with enough brain to be worth employing) should have to give very strong proof of his or her reform before being allowed to play with any security systems. The problems with the "reformed hacker" are these:

1. How do you know they have reformed? If they find a really major vulnerability, how do you know that they won't exploit it, or boast about it on the Internet?
2. How do you know that they are any good? Many hackers aren't that bright but just "got lucky" once. The key to success as an unethical hacker is single-minded patience and some aptitude for problem-solving, rather than intelligence. Others are "script kiddies", following other people's exploit recipes. Do you know enough about breaking security to distinguish a clever hacker from someone who just knows the jargon?
3. Do you really want to encourage the idea that a life of crime is a short-cut to a well-paid job in the world of corporate computing?

The other approach is to teach a computer specialist about hacking - and this is almost certainly far easier than teaching a criminal hacker about ethics. However, you can hire intrusion vulnerability testers from companies such as Internet Security Systems (**www.iss.net)** with a reputation for probity to defend, so why would you want to teach your computer specialists about hacking?

It's partly about control and risk management. If you call in external experts, you want to be sure that you know enough to get the right people and to make sure that they're giving value for money. Security is about managing risk, and if you don't know how hackers operate how can you be sure how big a threat they represent to you? Also, at least some security consultants make a living spreading "fear, uncertainty and doubt", especially when their client isn't well-informed, and even the more ethical ones spend their lives surrounded by successful exploits and may unconsciously exaggerate the risk these breaches represent.

In-House Value

However, knowledge of hacking techniques is also a useful practical skill for a trusted senior IT specialist. For example, you are less likely to forget something when configuring your firewalls (and most successful hacking instances exploit nothing more than poorly-configured systems) if you're fully aware of why it is being configured as it is instead of merely reading instructions from the book. An "ethical hacker" is also useful to have around when someone inside your organisation is playing silly games. An insider, abusing their legitimate access to (and knowledge of) your internal operations, is far more dangerous than any external hacker.

Let us imagine a scenario and examine how you might deal with it. Suppose head office is hearing disquieting reports from a devolved branch office - private emails

being read, cases of sexual harassment via email (with serious legal implications), "war games" being played in which someone is hacking into staff accounts on the LAN and planting "joke" programs (far from funny if you're trying to get work done). You're tasked, as technical troubleshooter at HQ, with sorting it out. When you arrive it is clear that devolution has gone too far and that the branch isn't in control of its IT, which is managed by two friends with little business experience but some technical skill (they possess home computers and freshly-minted MCSE certificates). They know enough to be dangerous but not enough to set up security policies, control the proliferation of admin accounts and so on.

So, you tidy up. You appoint a trustworthy administrator, promulgate the company Security and Privacy policies, explain (and get staff to sign to) the penalties for not following the policies, give staff the access they need to do their jobs and no more and, in short, all the standard "good things". Then acts of petty vandalism occur - servers crashing, disks filling up, public files being deleted. You suspect the two friends who previously ran the branch office IT operation, who are showing signs of bruised egos after being displaced by a respected younger Unix wizard from head office. They will almost certainly be aware by this stage of any relevant legislation controlling the misuse of computers, so if they're guilty they're probably being careful - and proving anything against them using the facilities in Windows NT is turning out to be difficult. You wonder just how far they'll go.

This is where you need your black hat knowledge (the bad guys in "Westerns" wear black hats) - not to target these idiots but to assess and control the risk they represent. Since they obviously have more access privileges than they should (if they're guilty - you must be careful about accusations until you have proof) you wonder if Back Orifice (a well-known Trojan which puts a back door into Windows NT systems) has been installed. You consider running one of the Back Orifice detection kits available on the Internet - but you know that many such programs have a Trojan payload themselves and are effective weapons for the black hats.

With knowledge of "black hat" hacking techniques you make a systematic survey of the vulnerabilities in your system that an unethical hacker could exploit (anyway a useful exercise), and perhaps find the *modus operandi* of your current miscreants. When you highlight the issue of "social engineering" with the staff, for example, you find that this pair has recently talked their way into a position from which they could install Back Orifice, and a copy is found. You don't have enough proof to take legal action, but they have breached security policy sufficiently to merit an official warning, and they face dismissal next time. After all this, the sabotage stops.

Becoming An Ethical Hacker

So, if knowledge of what the black hats get up to can help control the threat they represent, how do you learn to be an "ethical hacker"? Not from an unethical hacker, that's for sure. However, you can find firms in the security field with the necessary skills and with an established reputation for probity. For example, the education division of ISS (mentioned above), SecureU, runs an Ethical Hacker course, taught by the people who would test the vulnerability of your systems to hacking if you employed ISS to do that; in other words, their trainers are not simply teachers with no more than theoretical knowledge. Whatever such courses (from various organisations) you may choose, consider the following issues before making your decision:

- The reputation and probity of those giving the course.
- Access to practical experience, both from the trainers giving the course and through hands-on exercises.
- Treatment of the ethical, political and legal issues associated with hacking.
- Use of a systematic method; an unethical hacker only has to break in once, an ethical hacker has to find and log every vulnerability.
- Some form of test or certification to distinguish you from the amateurs.

That last point is quite important and leads on to other issues. If you are going to train in "black hat" skills, don't do it in secret. There's no need to boast about it, but make sure that your knowledge doesn't come as a surprise to those that matter

"You can "war dial" your target, using a computer to generate and dial different numbers, looking for one with a modem attached and bypassing the firewall."

to your career. Otherwise, things may get embarrassing. Suppose a copy of Back Orifice is found. Ideally, you'll find it or be the official source of advice. Otherwise, people may remember that you seemed to know a lot about such things and you may become a suspect. Unfortunately mud sticks to the innocent too, and people don't always behave rationally when someone is breaching trust in an organisation.

The ISS Ethical Hacker Course

Let us now look in detail at the ISS course, how it is set up and what it teaches you. Even if you don't go on the ISS course it is a good benchmark to compare other courses against. First, ISS is an appropriate source for such a course. It is a global organisation, founded in 1994, and a trusted security provider, protecting digital assets and ensuring safe and uninterrupted e-business. ISS's security management solutions protect more than 6,000 customers worldwide, including 21 of the 25 largest US commercial banks, the top 10 telecommunications companies and more than 35 government agencies. In other words, ISS is not a fly-by-night company, and it has a valuable reputation it won't wish to lose. It works closely with companies like Microsoft (see **mspress.microsoft.com/-prod/books/3873.htm**, for the bible on Windows 2000 security, written by ISS), and it is clearly a fit company to advise on security and to give such a course. Sources from the company have also said, in personal conversation, that it provides the course only to members of genuine customer organisations rather than to anyone off the street; ISS prefers to avoid training the next wave of unethical hackers to hit the headlines.

There is a potential downside with using technicians as teachers, since they may be less proficient than teaching specialists at keeping an eye on students to make sure they are keeping up, and at designing tests or exams, but this need not be an issue in practice. In any case, the advantages of practical experience outweigh other considerations. The ISS course objectives are to enable attendees to:

- Describe how hackers are able to defeat security controls in operating systems, networked environments, and generally circumvent security systems;
- Identify how security controls can be improved to prevent hackers gaining access to operating systems and networked environments.

The course teaches a methodical process for gaining access to a target system, in the following stages:

- Educate the client in the risks associated with hacking and make sure suitable controls are in place. Consider the legal aspects of, for example, putting personal data at risk, and obtain appropriate authorisation for what you're doing.
- Gather information about the target passively (ie, without going outside the public domain).
- Use active but (if possible) carefully hidden probes to gather detailed technical information about the target and to map the topography of its network.
- Map out the target network, possible vulnerabilities and exploits, based on the information you have gathered.
- Exploit the vulnerabilities and gain access. Generally, you will gain access; install a back door such as Back Orifice; clear up any signs of your exploit; and leave before anybody notices you.

"An amazing amount of information useful to the intruder can be gathered from what companies put into the public domain. Obviously, this is an area which the security-conscious company will strictly control."

Preparation And Authorisation

The ethical hacker mustn't cause damage, which is easy to do while mucking about with security back doors. Even the ethical hacker may be abusing local computer misuse legislation (and possibly data protection legislation) unless he or she is operating with explicit permission. (In the case of personal data, the subject of the data has rights which must be respected.) ISS thus stresses the importance of having a proper contract (or memo of authorisation, if in-house) and complete, validated backups before starting an ethical hacking exercise. You must not stray outside the bounds set by the client or sponsor - which should be defined in writing.

The methodical approach recommended by ISS is specialised for the ethical

hacker - it is designed to find all vulnerabilities and document them, instead of running just one ego-boosting exploit. However, during the course you do actually complete an exploit against a carefully constructed live system, and at least this degree of practical experience would seem to be essential in any ethical hacking course. As anybody who has tried to find a file on someone else's hard drive can attest, even simple tasks become difficult when you're away from home.

Passive Information Gathering

An amazing amount of information useful to the intruder can be gathered from what companies put into the public domain. Obviously, this is an area which the security-conscious company will strictly control, releasing only information necessary to do business, and at this stage you can start to assess the target's general awareness of security issues. Company reports and press releases (often available on the Web) can highlight acquisitions, with the possibility of temporary and badly secured communications links or poor security integration. Press releases may tell you what technology you have to face. The actual source code of the Web site can be viewed in most browsers and can yield useful information on naming conventions and (possibly in comments in the HTML) the names and email or IP addresses of staff and host machines behind the firewall.

Other useful sources of information are the address systems ultimately managed by ICANN - the Internet Corporation for Assigned Names and Numbers, a non-profit organisation that manages IP addresses and domain names. It delegates IP address management to three Regional Internet Registries (ARIN, APNIC and RIPE, respectively for the Americas / sub-Saharan Africa, Asian Pacific region / Europe, and the Middle East / the rest of Africa), and domain name management to private companies. The information held by these organisations is mostly public domain and can be queried using utilities such as WHOIS. You can sometimes find names of key staff and their contact details; the geographic location of key sites; name servers and IP addresses; the ISP a company uses; and even when it last updated its Web site (an indicator of how Internet-savvy it is).

A lot of technical information can be gathered from legitimate queries directed at the DNS system and a company's name servers. Tools such as NSLookup and Sam Spade have been written to make this easier. Once you have an address, Traceroute and other utilities can be used to add new addresses and to start to map out the network topology. You can even analyse the SMTP headers in a rejected (misaddressed) email message in order to map out more of the network.

Many companies are careless with information because they don't realise the usefulness of information such as the name of someone in tech support when mounting "social engineering" attacks. If the Web site says contact Fred.Bloggs@megacorp.com for support (rather than contact support@megacorp.com), and the switchboard will tell you the name of someone's PA, and switch you through, you can say something like: "Fred Bloggs in Support has asked me to find out your password because one of your jobs is holding up the system, and we need your password in order to cancel it before you get into trouble". Unfortunately, hackers often succeed in obtaining passwords if they use this approach effectively. However, this is moving into the next phase, active information gathering.

"The methodical approach recommended by ISS is specialised for the ethical hacker - it is designed to find all vulnerabilities and document them, instead of running just one ego-boosting exploit."

Active Information Gathering

Active information gathering runs the risk of attracting attention, but activities like "dumpster diving" (searching waste bins for documents containing useful user names, addresses and so on) and social engineering often escape notice. In a security-aware organisation people will be careful about what they discard, and strangers hanging around the premises will be noticed, and all staff should know not to give out passwords on the phone. More active attacks can yield more information but increase the risk of putting the target on its guard. Since it's easy to find external phone numbers for the company, and companies usually have direct dialling in for blocks of adjacent numbers, you can "war dial" your target, using a computer to generate and dial different numbers, looking for one with a modem attached and bypassing the firewall; this scenario is not nearly as unusual as it should be.

Another attack is to scan all the ports in any hosts you can get to, looking for ports that are open, services waiting for input and so on. (A list of legitimate well-known services can be found at **www.isi.edu/in-notes/iana/assignments/port-numbers**, and a list of back door and Trojan default ports at **www.sans.org/newlook/resources/IDFAQ/oddports.htm**.) A TCP connect scan using the three-way handshake is the obvious approach: SYN packet sent, SYN/ACK received, ACK packet sent and connection established - these are standard TCP packets with various flags set. However, most perimeter security devices will detect this and a large number of connect attempts in a short time will ring alarm bells. Spreading the scan over a long time period may evade detection, although analysis of perimeter device logs can detect even this.

A stealthier TCP SYN scan (one that may not be detected) results from responding to the SYN/ACK with a RST packet, which breaks the connection, but most commercial security devices will log this these days. More stealthy still is to formulate packets with unusual flag combinations (TCP FIN, XMAS and NULL scans); the applicable RFC specifies that a closed port will respond with a RST while an open port will ignore the exploratory packet. (Note that the Microsoft TCP/IP stack doesn't follow the RFC, so this won't work with Windows NT, except that such behaviour helps to show that NT is running.)

Scanning UDP ports usefully is harder, although sometimes informative, because UDP is connectionless and you can't tell if a null response simply corresponds to a lost packet; besides, many services may simply not know how to respond to data on a UDP port. Another approach is to send mis-formed packets to a host and analyse the results to determine the operating system in use; some follow the RFC recommendations more precisely than others, so each operating system has a recognisable fingerprint, often down to the version level. This sort of operating system fingerprinting probe can even be run as a passive exercise, merely analysing the characteristics of routine packets sent out by the target computer (see **www.enteract.com/~lspitz/finger.html**).

Interpreting the results of host scanning isn't trivial and requires a fair knowledge of TCP/IP and the various applicable RFCs. Unfortunately, from one point of view, there are various tools available commercially and on the Web that will run the scans for you and interpret the results - Nmap, Vetescan, Nessus, ISS Internet Scanner, Firewalk and so on (see, for example, **www.insecure.org/nmap/nmap_manpage.html**). Any systems manager assuming that he or she is protected from this kind of attack because of the skill level needed for success is living in a fool's paradise.

Vulnerability Mapping

Once you have completed your active and passive information-gathering exercises you will have a vast amount of information about your target system. It is worth noting that the average "black hat", looking for any system that will provide a chance of an ego-boosting exploit, has an easier task than either the ethical hacker or criminal, who generally target specific systems. Turning this around, being easy to attack makes you a possible target in itself; having nothing worth stealing is no protection.

"If you are going to train in "black hat" skills, don't do it in secret. Make sure that your knowledge doesn't come as a surprise to those that matter to your career."

It is important to structure the information you have gathered. Try mapping out the target network against the standard patterns for network design and firewall deployment. Annotate the map with host names, IP addresses, operating system and version, services running and so on. Now you can try to identify vulnerabilities. There are many sources of vulnerability information, many of them legitimate. Many are on mailing lists or in newsgroups, but you might want to use an alias address before joining these, as some of those present may see fellow members as potential targets. Typical sources of vulnerability information are:

- Fix advisories sent round by software vendors. These probably won't tell you how to perform the attack, but if you can identify a host running the unfixed version of the software you have a point of attack.
- Full Disclosure Advisories. There are a number of mailing lists, such as Bugtraq, which will typically include full details of successful attacks on software vulnerabilities.
- Technical software training courses often highlight exposures, so that atten-

dees can correct them. Some attendees don't bother (or aren't encouraged to) correct anything on their return to work.

- Books such as *The Tangled Web: Tales of Digital Crime From the Shadows of Cyberspace* (Richard Power, Que, 2000, ISBN 0-7897-2443-X) or *Hacking Exposed, 2nd Ed*, (Joel Scambray, Stuart McClure, George Kurtz, Osborne / McGraw-Hill, ISBN 0072127481). You might think that systems managers would at least read the literature and correct published vulnerabilities, but many haven't.
- There are, allegedly, various secret hacker discussion groups and Web sites where information on successful exploits is exchanged. Little information on the Internet is guaranteed to be reliable, but this is possibly less reliable than most of it - people may deliberately mislead potential competitors or boast about what they haven't actually tried.

Obviously systems managers should try to keep up with such resources, and correct vulnerabilities, but this is a daunting task. Fortunately, the root causes of much vulnerability can be corrected at source because they are caused by one or more of the following:

- **Mis-configuration of purchased software**. Ensure adequate training and installation advice is available.
- **Poor programming practice**. Use more controllable languages such as Java, introduce code inspections and lifecycle testing - and more training.
- **Poor security culture**. Make sure that you have a security policy, that security awareness training is given to new recruits, and that security is included as a requirement early on in the development process.

The Practical Exploit

Typically this will exploit bad practice by an ISV or in-house developers. Take the "buffer overrun" exploit, in which you write past the end of your buffer and direct the next-instruction pointer on the stack to your own code. This then runs with the same privileges as the service you've broken out of. A hacker will probably use such an exploit to set up a shell from which he or she can install a back door for future access, explore the network for further vulnerabilities and more useful information to download, and erase all traces of the visit. Of course, none of this has been tested in the system it is running on, and may end up crashing the system and destroying data, no matter how benign the intruder's original intentions. Hackers may also boast of their exploit, which may attract other attacks and badly damage your commercial reputation.

These days, most hackers won't code up an exploit from scratch. They'll try to pick up on new vulnerabilities as they are reported on the Internet and exploit them before systems managers get around to installing the fix, assuming they do; configuration management systems in the better organisations protect production systems from untested or unauthorised code, but can slow down application of security fixes, which aren't always bug-free themselves. Within hours of a new vulnerability being publicised a script or code to exploit it will probably appear on a site such as **www.hack.co.za**, **packetstorm.security.com** or **www.root-shell.com**, nicely classified by operating system and version. The ethical hacker can download the code, compile it, check it for errors, logic bombs, hidden Trojans and so on (black hats won't bother with the checks) and try it out. The ethical hacker then documents both the vulnerability and ways of addressing it.

"Teaching a computer specialist about hacking is almost certainly far easier than teaching a criminal hacker about ethics."

PCNA