CALUG, 9/10/03

"We know what you do at night," or, My Life As A Pen-Tester

Raven Alder
Security Consultant

What is a pen-tester?

- Penetration testing – approaching a network as if one were a malicious attacker, and determining its vulnerabilities
- Requires specific negotiations with client before beginning the test – what is to be tested, when, what is off-limits
- Make sure you have authorization from an appropriate company officer on paper before starting the test. Keeps you from getting sued.

Why pen-test?

- Rest assured that there are hostiles out there who won't hesitate to poke at your network

- Knowing about your security holes is the best way to fix them.

- Bringing in someone who has seen and secured many networks increases your odds of finding problems.

Types of pen-testing

- Black-box – you don't know what the network you are testing looks like, what sorts of machines there are, or any details. You approach with the same information a brand new outside attacker would have – none.
- White-box – company officials give you information on network topology, operating systems, et cetera before you begin the test. You begin with more knowledge, and this often shortens the time it takes to test.

Black box testing – discovery

- Usually, you get a domain name or an IP address block, and that's it. Harvest information from public sources to fill out your knowledge of the company.
- Use whois databases from the registrars to determine ownership, contact names, and e-mail addresses.
- Check the DNS servers for your target domain – see if they will allow zone transfers. If so, get one. If not, start querying for every IP in the block.

Discovery countermeasures

- Use role accounts for whois contacts – System Administrator rather than Jane Q. Admin. This lessens the chances of social engineering.
- Don't give out more information than you have to. You may wish to consider Domains-by-proxy (http://www.domainsbyproxy.com) or a similar solution to hide your private information.
- If you have an internal network, do not share its DNS with the Internet. Have a separate internal server.

Packet level discovery

Nmap (http://www.insecure.org/nmap/) their network to identify hosts.

- Don't rely on just ping sweeps – that may miss machines configured not to respond to ICMP.
- Consider IDS systems that may be present, and whether you're trying to be stealthy or not. If so, slow scan, strange packets (not just a TCP connect or half-open).
- Check UDP as well. Do OS fingerprinting.

Packet level countermeasures

- Turn off ports and services that are not needed on each machine. Each one is a possible attack vector shut down, and one less thing to patch.
- Install firewalls to filter out traffic that has no business coming from the outside world. For example, there is no reason not to filter the SMB ports from the Internet, or the SQL database ports. Those should only be accessed by local machines that use those services.
- Install an IDS, and watch its logs to know when you are being scanned or attacked.

Service level attacks

Once you have identified ports and services that are listening on individual hosts, determine the software they are running and what version it is.

- Banner grabbing is a useful technique, whether automated or just telnetting to the port and seeing what is returned.
- Amap (http://www.thehackerschoice.com) identifies services running even on nontraditional ports.

Service level prevention

- Patch, patch, patch. Read Bugtraq, Full Disclosure, or some similar mailing list that advises you of new vulnerabilities. Update your system when appropriate.
- Change or obfuscate the banners displayed by your services. There's no need to give away information about your configuration if you can help it.

The surgical strike

- Once you have the versions of services that are available, it's just a matter of Google.
- Check online sites like CERT or SANS to see if any of the software versions are vulnerable.
- Check archives of mailing lists – often they include exploit code.
- Googling for “sploits”, “sploitz”, and the name and version of the service is also often fruitful.

Most of the time, you can download a ready-made hack. (Pen-testers that do this all the time tend to accumulate a small collection.)

The surgical strike – defense

- If the attacker has gotten this far and found a vulnerable service, you're already kind of screwed.
- Make sure you have a good incident response team on hand in case you do get hacked.
- Make sure logging on your devices is as verbose as possible, to have a better chance of figuring out how they did it. Correllating logs from perimeter routers, firewalls, servers, and IDS systems can help a lot.

I 0wn j00! Now what?

- So what did it get you? Not all exploits are remote-root. If you're not root, you will want to be. Check out what's available on the local machine, and see if any of those programs are locally exploitable.
- You now have another point of view – one from inside the network. If the machine you have compromised is behind a firewall, it may be worth taking another look at DNS, portscanning, et cetera. Local machines are often trusted with more information.

I am 0wn3d – local defense

- Consider a host-based intrusion detection system on your machines, to alert you when compromise happens.
- When possible, run services in a chroot jail and/or as a non-root user. This means that if Apache is overflowed, they get the user "nobody" or "apache" rather than "root".
- Log verbosely, read your logs, and log to another machine. That way if the attacker "rm -rf"'s /var/log, you still have another copy.

I am 0wned – network defense

- Make sure your IDS and your sysadmins look at internal activity as well as external activity. Most corporate hacks I've seen happen from the inside.
- Limit trust relationships between machines on the local network – consider DMZs and separate subnets with a firewall in the middle.
- Don't use cleartext logins on the local network if you can help it. One local compromise and a packet sniffer can yield tons of authentication information.

I am 0wn3d – what do I do?

- Consider forensics. Unplug the network cable to limit further damage.
- Get state information – process lists, what's in memory, what ports are listening.
- Take disk images with dd. Use The Coroner's Toolkit (http://www.porcupine.org/forensics/tct.html) to analyze your system post-hack.
- Format, reinstall the OS, patch it, and restore your data from a known good backup. Do NOT reinstall binaries from backup or you may be handing your system back to the attacker.

I 0wn j00 – securing your base

- You want to be able to get back into this system without having to hack it again.

- Install a backdoor.
- Create yourself a login with root priviliges.
- Cover your tracks with log editing, loadable kernel modules, or rootkits. If you just change the root password, they'll know you're here.

Preventing consolidation

- Check regularly for new logins, particularly with a UID of 0, 00, or something similar.
- Host-based IDS systems will sometimes require a separate password to change system binaries. Consider LIDS for Linux (a kernel modification), setting files immutable with chattr (will baffle new attackers but not stop seasoned ones), or disabling modules in your kernel.

Assessing your booty – the compromised system

- Determine what sort of a system it is. Netstat from the inside, look for running processes that may have valuable data.
- Do you have anything of worth on this system? Databases, Web content, e-mail?
- Can you get authentication information from this system? Trojan ssh to save passwords, which may be in use elsewhere on the network. Install a packet sniffer to catch cleartext logins.

Newer attacks – application hacks

- Look at Web-based forms and applications for bad coding and security vulnerabilities. You may be able to get that database without having to bypass a firewall or get root on the system at all.
- SQL injection, lack of input validation, and verbose error messages returned to the client all help the attacker.

Application hacks – defense

- Have your Web code written with security in mind, and/or have it audited.
- Err conservatively on the input you'll allow. It's not enough just to automatically parse ' and “ -- that wouldn't stop Unicode attacks, for example. Only allow what you need.
- Log errors, and have someone read and check those logs.

Other tricks of note

- Firewalls can make your life difficult – try firewalk (http://www.packetfactory.net/projects/firewalk/), lft (http://www.mainnerve.com/lft/), and interesting nmap options to bypass or circumvent them.
- If you can find a *nix machine with Xwindows behind the firewall, install Nessus (http://www.nessus.org) on it and run that to get a fairly detailed report of vulnerabilities on the local network.

Reporting in to the client

- Keep very thorough records of what you did, when you did it, and be able to prove it. Many companies love to blame the pen-tester for any network or system problems that they have while you're testing.
- Give them a thorough report of the vulnerabilities you found, and your recommendations for fixing them.
- Obviously, be professional and don't harm their systems. Duh.

Networking in many senses

- It's good to know subject matter experts. Hang out with other security geeks, or work in teams. This allows you to get excellent in-depth knowledge of whatever systems you might come across in a test.
- Know good sysadmins (or be one), in case the client asks you for a recommendation to fix their problems once you've found them.
- Know good code auditors that you can refer them to if they need it.

In conclusion...

Thanks for listening. Go forth and secure your networks.