Global Information Assurance Certification Paper

Curt Purdy
GSEC Version 1.4b

A Short History of Hacking and Case Studies of Cracked Windows Servers

Abstract

This paper is a case study of a series of Windows Servers I have dealt with over the years that were the targets of malicious attacks and code. Environments range from an SMB business to a multi-national corporation to an ISP. Operating systems include NT3.51, NT4.0, W98 and W2K Server and it spans a period of eight years. This paper documents real-world cases, solutions, and cites relevant research. I highlight the differences of each attack/compromise as well the means of discovery, method of sterilization, and verification of sanitization.

In addition I have encapsulated a historic perspective of the art of hacking and cracking. My personal experience with computing, communications, and networking closely parallel the evolution of the Internet, and I convey some of my own experiences along with the historic overview. I attempt to offer a unique perspective of evolving technology and psychology in the world of the Internet.

Glossary From the Jargon File

hacker n.
[originally, someone who makes furniture with an axe] 1. A person who enjoys exploring the details of programmable systems and how to stretch their capabilities, as opposed to most users, who prefer to learn only the minimum necessary. 2. One who programs enthusiastically (even obsessively) or who enjoys programming rather than just theorizing about programming. 3. A person capable of appreciating hack value. 4. A person who is good at programming quickly. 5. An expert at a particular program, or one who frequently does work using it or on it; as in `a Unix hacker'. (Definitions 1 through 5 are correlated, and people who fit them congregate.) 6. An expert or enthusiast of any kind. One might be an astronomy hacker, for example. 7. One who enjoys the intellectual challenge of creatively overcoming or circumventing limitations. 8. [deprecated] A malicious meddler who tries to discover sensitive information by poking around. Hence `password hacker', `network hacker'. The correct term for this sense is cracker.

The term `hacker' also tends to connote membership in the global community defined by the net (see the network and Internet address). For discussion of some of the basics of this culture, see the How To Become A Hacker FAQ. It also implies that the person described is seen to subscribe to some version of the hacker ethic (see hacker ethic).

It is better to be described as a hacker by others than to describe oneself that way. Hackers consider themselves something of an elite (a meritocracy based on ability), though one to which new members are gladly welcome. There is thus a certain ego satisfaction to be had in identifying yourself as a hacker (but if you claim to be one and are not, you'll quickly be labeled bogus). See also wannabee.

This term seems to have been first adopted as a badge in the 1960s by the hacker culture surrounding TMRC and the MIT AI Lab. We have a report that it was used in a sense close to this entry's by teenage radio hams and electronics tinkerers in the mid-1950s. [1]

cracker n.
One who breaks security on a system. Coined ca. 1985 by hackers in defense against journalistic misuse of hacker (q.v., sense 8). An earlier attempt to establish `worm' in this sense around 1981-82 on Usenet was largely a failure.

Use of both these neologisms reflects a strong revulsion against the theft and vandalism perpetrated by cracking rings. While it is expected that any real hacker will have done some playful cracking and knows many of the basic techniques, anyone past larval stage is expected to have outgrown the desire to do so except for immediate, benign, practical reasons (for example, if it's necessary to get around some security in order to get some work done).

Thus, there is far less overlap between hackerdom and crackerdom than the mundane reader misled by sensationalistic journalism might expect. Crackers tend to gather in small, tight-knit, very secretive groups that have little overlap with the huge, open poly-culture this lexicon describes; though crackers often like to describe *themselves* as hackers, most true hackers consider them a separate and lower form of life.

Ethical considerations aside, hackers figure that anyone who can't imagine a more interesting way to play with their computers than breaking into someone else's has to be pretty losing. Some other reasons crackers are looked down on are discussed in the entries on cracking and phreaking. See also samurai, dark-side hacker, and hacker ethic. For a portrait of the typical teenage cracker, see warez d00dz. [1]

own v.
To compromise a server or workstation through cracking, in which the attacker gains the ability to execute commands remotely, including up and down-loading of files.

Historical Perspective

Although my first experience with computing began in the early 70's with main-frame programming in Fortran and Cobol, my interest in hacking began in 1979 with assembly language programming on my TI 99/4, the home computer with the first 16-bit processor that came out 2 years before IBM's PC 8-bit processor. I was forced to use assembly because the only other language available was Basic that could not do what I needed.

This is the foundation of hacking and the concept has existed and been used long before there was a Jargon File [1] or even an Eniac. Cracking is another story, and though coined in 1985 referring to software crackers, I am going to say the modern concept of network cracker began on November 2, 1988. That is the day Robert Tappan Morris, Jr., son of the then current National Security Agency chief, released the first self-replicating code on the Internet that later came to be known as worms. [2] I am going to say it was a crack as opposed to the accidental release of a hack because he released it from MIT rather than his home university at Cornell.

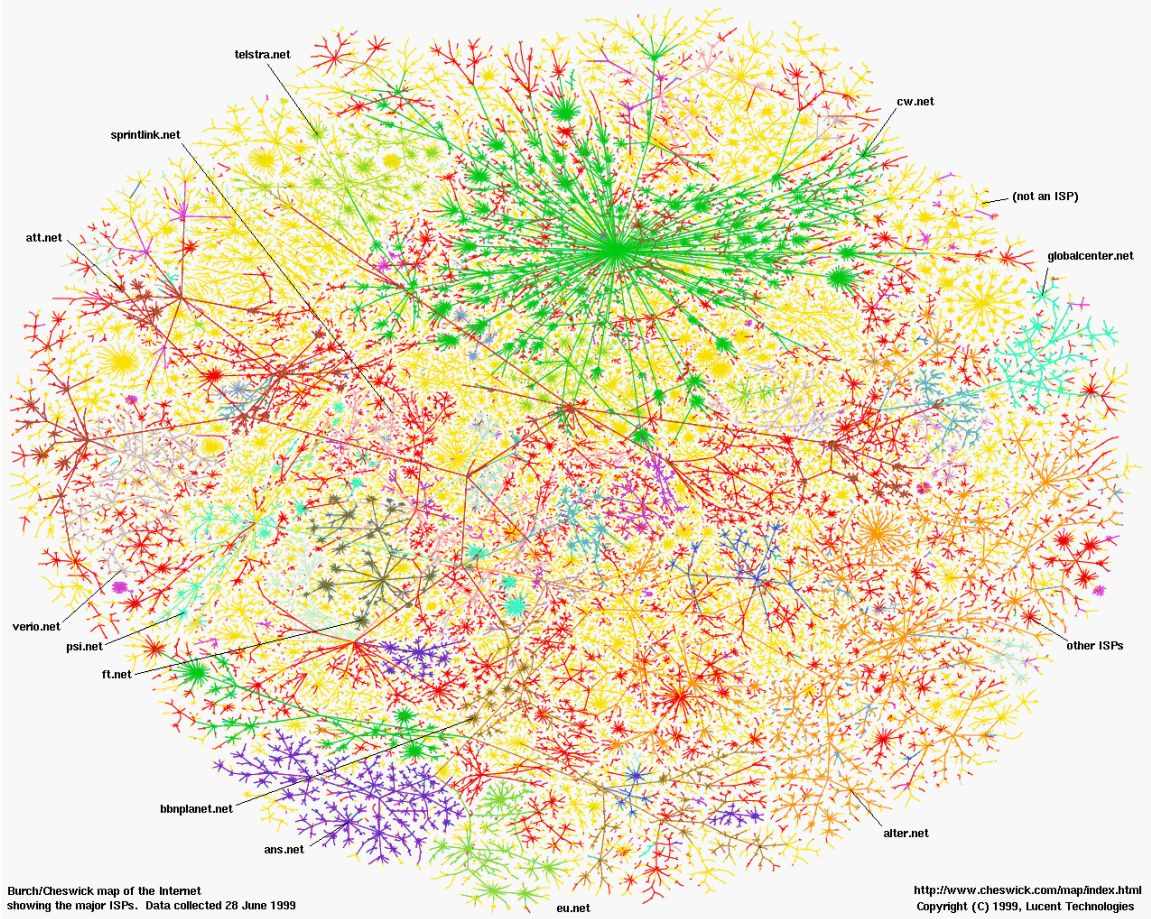
Regardless of how one feels on this point, that little worm has grown into megabyte monster trojans and come full circle with the SQL Slammer UDP worm of 2003. At 376 bytes, it is smaller and much faster than any worm in history, having infected 90% of all vulnerable servers on the Internet within 10 minutes. Cracking psychology also seems to have come full circle in the relatively short history of the Internet, as I describe at the end of this paper.

Since my own history seems to parallel that of the Internet's pretty closely, I may have a somewhat unique perspective to share with the community. Though not quite old enough to have been aware of ARPANET chief Dr. J.C.R. Licklider prophetically nicknaming it the "Intergalactic Network" [3] I am old enough to recall a member of our users group becoming the proud owner of the first 1200 baud modem in town and us all rushing over to his house to watch the BBS text scroll so fast up the screen you actually couldn't read it!

These were the days of "sneakernet" when viruses were spread by sticking someone else's 8" floppy in your disk drive. Protocols were proprietary and few had heard of TCP/IP. There were other milestones in networking, like the development of X-Modem compression that let you download those huge 100k files, and ARCNet that let you connect computers with cabling, but the great dawning came with the Internet.

In the early 90's, I saw the promise of what was then not yet officially named the Internet, when I first telneted into USENET and ran FTP to download files and Pine to send email. Then this new protocol shows up called HTTP, and Tim Berners-Lee develops this thing called the World-Wide Web along with a neat little program called Lynx, and suddenly you could point and click your way across the Internet. When NCSA Mosaic came along in '93 it blew the top of my

head off. And it's been downhill ever since. Like I tell my wife, "I'm never gonna retire. They'll have to pry my cold dead fingers off the keyboard to get my 'puter!"



"Map of the Internet by ISP" by the Internet Mapping Project, Lumeta Corporation [4]

DoS Denial of Service Attack

My experience with Internet attacks began early in 1996. As a newly formed ISP with our first POP in East Texas, I had been following with interest the unfolding tragic story of the largest independent ISP in New York, Panix. I found it fascinating but frightening, watching the progression in a newsgroup visited by the two Panix techs explaining what was happening and looking for help. Seeing these guys go from an initial panic at being completely down for a day to almost resignation after being down for a week was hard to fathom. Alexis Rosin was just beginning to make headway at re-constituting service when I came in to work one Friday to a total loss of service ourselves.

By noon, I had both of the phones in my office tied up with Cisco support on one and our backbone provider MCI on the other. By 2pm I got my cell out of the car

and brought CERT in on the party. By 5pm, they all apologized for not being able to help with my plight. I thanked them for their time, hung up, and told my secretary that I would let her know if she still had a job on Monday.

I then got out all my firewall and TCP/IP books, tore down the firewall, and began rebuilding it brick-by-brick. With the help of what I learned over the past week from Panix, by 6am the next morning I had 80% of my service back, mainly by blocking TCP on the ports the attacker was sending the SYN-flood in on, even though at the time, I had no idea that our NT3.51 servers were spewing out hundreds of ACKs a second for each SYN received and not stopping because they were getting no SYN-ACK back.

The servers that port-blocking could not work on i.e. they were hitting port 80 on the web-server and port 25 on the mail-server, I simply changed IP addresses on them because we controlled our own DNS. Luckily they were hitting by IP and not by name.

It turns out that the summer before, 2600 magazine had published code for producing a SYN-flood attack under UNIX. [5] The weakness in the connection-oriented TCP transportation protocol was known and understood for many years prior to the Panix attack [6] and acted on by some UNIX vendors, but not Microsoft.

The attack is based on the 3-way handshake that initiates all TCP sessions. It uses the simple concept of sending many single SYN packets from a spoofed source address that would not respond back with an ACK. To make it worse, the victim will send back hundreds of ACKs for each SYN, desperately trying to shake that hand across cyberspace. In addition, each ACK occupies a holding place in the backlog on the server, waiting for a SYN-ACK that never comes. The result is that a 14.4k modem can bring down a t1. In effect, we were DoS'ing ourselves.

We didn't know it at the time, but we were experiencing the first form of Internet attack (if you believe the Morris Worm was really an accident). Through subsequent analysis of the logs, we determined the attack came from a former competitor whose customers we had taken when we came into town. So this was a form of business competitive information warfare with the very specific goal of driving us out of business.

Over the years, many sites have been SYN-flooded including the FBI, the White House, NASA, and the military. Today all OS's incorporate fixes that restrict the number of SYN-ACKS that a server will respond with. Some of the best resources on the subject, including the newer DDoS Distributed Denial of Service attacks are by Steve Gibson [7] and Dave Dittrich. [8] Cisco also offers a useful page. [9]

One of the “good netizen” practices we instituted as a result of this was putting filters in our firewall known as egress filters to prevent our users from attacking others using spoofed addresses. This experience was an eye-opener, hair-raiser, and the beginning of the long endless road known as a career in information security.

Having survived the test of almost losing everything, I was no longer afraid, but I was also no longer as confident as I was. I know that at any moment, a cataclysmic event could occur bringing down networks all over the country including my clients and my own. It will have no precedence and there will be no experience or knowledge-base to rely on. There will only be my own wits and talents.

Warez

In 1999, I had a client call up complaining that his Internet service was dog-slow even though he had a full t1 through us. I got into our core router to look at his port utilization and his line was definitely maxed out. The unusual part of it was that it was all out-going traffic, whereas normally the traffic is 90% incoming.

I then went on-premises and began looking at servers, found unexpected services on their NT 4.0 server and realized it was owned. I found it running a kit with ServuFTP server along with smt, netcat, kill, psservices, info, cygwin1.dll and various other tools in C:\winnt\system32\spool\w42x86 as their initial location. The cracker found a local administrator with blank password, logged in, went to her TFTP server, downloaded nc.exe, and it was all over.

This was a different ethic than the first case, although just as criminal. She stole both bandwidth and disk storage that had been paid for by someone else for her own nefarious plans. To her cracker clan, she was ‘leat.

Gaining ownership of the server, the cracker proceeded to create the warez folders for file storage in a convoluted series of folders under C:\Inetpub\wwwroot\ _vti_txt. The directory tree under this was \tagged\by\##SLAMMER##\lpt\1\2\com1. I found that I could delete none of the folders. I opened a command line, ran dir /x on the directories and found their 8.3 Dos names were really lpt~001, 1~001, 2~002, and com1~002. I then ran the POSIX command RD \\?\d:\inetpub\wwwroot_vti_txt\tagged\by\##SLAMMER##\lpt~001\1~001\2~002\com1~001 to remove the final file-carrying folder and then repeated for subsequent folders, freeing up over 3gb of disk space.

Once I killed the illicit services, deleted the warez directories and files, and deleted the illicit folder, everything seemed to be back to normal. Needing further proof that the server was actually clean, I ran fport.exe from Foundstone and

pslist.exe from SysInternals (great free tools and ironically creator of psservices installed with the cracker kit above) to verify that no illicit processes were still running.

During forensics is when I determined the entrance point to be the local user with blank password giving "temporary" administrator rights that were never subsequently down-graded. It was easy to convince the client to install a firewall appliance, go to private NAT addressing on his local network, and attach to the server remotely through VPN.

Trojans

One day in 2000 while reviewing my company's firewall logs, I noticed one of our accounting staff visiting a chat room. As that was against company policy, I had a talk with her about it. She was so fervent in her denial, that I took a second look at the logs. I then saw that her Windows 98 workstation was going out precisely at 2 a.m. every morning. At that point I knew she had a backdoor Trojan and that our firewall had just been rendered a 40-pound boat anchor, at least to this particular cracker.

I went to her machine and saw that it had an (unnamed to protect the innocent) anti-virus on it but had not been updated in a while. I proceeded to update it and scan the drive. It found the PrettyPark backdoor Trojan that was obviously the culprit, but could not remove it.

It turns out that it was a very good thing that it could not be removed, because that forced me to put F-Secure on the box. It not only found and removed PrettyPark but also found and removed a variant of Sub-Seven that the other AV had missed, apparently because the code had been modified and customized by the cracker.

Apparently what happened was the workstation was originally infected with PrettyPark by the employee opening an infected attachment to an email. The virus then went to the prescribed IRC chat channel to announce to everyone on the channel it's availability to be used as desired. The particular cracker that took control of the box then installed Sub-Seven on it so that she alone could control this particular box.

At this point the cracker was in our network with the full rights of whatever user was logged on, including administrator while I was logged in troubleshooting it. She could have grabbed whatever HR, financial, or business-plan files she wanted to off the server. And all this could happen behind an enterprise-class firewall and in front of the IDS without a single alarm going off.

In addition she could have used it to attack other computers on the Internet with full safety from detection of her IP address. In this case, we did not have direct evidence of motive, but you can be sure it was not good. It may have been the workstation did not have enough disk space to warrant a warez server, or the owner my have used it for DdoSing with simple ping-floods that would not have been noticeable on our network. Or it may be that a competitor got in, grabbed what they needed, sanitized the logs, and we were none the wiser. The next section covers our discovery of a version of malicious code used for serious DDoS attacks.

DDoS Distributed Denial of Service

In 2002, I was remotely-monitoring the bandwidth usage of a customer when I noticed unusual usage, namely more outgoing packets than incoming, in fact at times reaching 1.54 mb, maxing their connection. I immediately drove over to see what was up, and found unexpected services on their Dell W2K web server and suspected it had been compromised. I found a service called m2kserve and thought that I had found the illicit service. After googling m2kserve, however, I found it was actually the Dell server manager which I was unfamiliar with.

Finding nothing obviously wrong in services, I begin to look closer at the file system and find services.exe under \WINNT instead of under \system32 where it belongs. That is when I realized the server was owned. Sure enough, taking a second look at services showed that there were two SERVICES.EXE running. I ran strings.exe, another free util from SysInternals on \WINNT\SERVICES.EXE to read the ASCII and Unicode contents and saw that it was actually ServuFTP server. Found it!

I then found an epic folder under system32 containing hundreds of files and over 7mb, a very elaborate crack that turned out to be multi-purpose. She had not only placed 7 full-length DVD movies on the server gobbling up 30gb but was also using it as a DDoS member to attack UNIX servers with identd.exe. No wonder the t1 bandwidth was being swamped!

This one was tough to kill. I could not delete services.exe because of permissions, even though I was administrator. One of the things about Windows is that even administrator is still just a user and does not have the power of System that runs as a separate user controlling all system processes. This is unlike UNIX where root runs as System, allowing root to do everything, including trashing the system if root is not careful. Luckily the OS was on a fat partition, so I booted into DOS, deleted it, rebooted and deleted the epic folder and all warez folders. I ran Vision, a newer GUI program from Foundstone that tells you just about anything you want to know about your server. It was clean.

Forensics was very tough on this one. We religiously patch our customer's

Microsoft servers remotely 48 hours after hotfix release and I verified administrator accounts had secure alphanumeric passwords.

It turned out to be a non-updated Matt Wright's formmail cgi script that was susceptible to a buffer overflow allowing command line execution in a folder with execute permission (cgi-bin). I installed 1.9 version and have had no more trouble. Just goes to show you, you have to patch everything, not just OS's.

This case had the interesting psychological motivations of both aggression and charity. After all, she was giving everybody a free ticket to the movies!

Zombie/Bot Cyberwarfare

My last example occurred earlier this year on our own network (blush). Luckily it was found quickly before any real damage could be done. We use Cricket to monitor all our lines and noticed a big jump in outgoing bandwidth on our t3 Internet backbone line from our DMZ.

We began looking at the most likely suspects first and found a newly installed SQL Server box, unplugged it, and the bandwidth immediately dropped. After roundly chastising the tech for putting a server online before fully patching it, I began forensics. The steps are outlined as follows:

1. I connected with a cross-over cable to a laptop with tcpdump sniffer and determined the box was scanning the network for windows shares on port 139.

2. I ran netstat -an and got:

```
  UDP    0.0.0.0:3456           *:*
```

3. From the laptop I attempt to ftp to suspicious port 3456 and got:

```
ftp> open 127.0.0.1 35394
Connected to 127.0.0.1.
220 Serv-U FTP Server v4.0 for WinSock ready...
```

So, we have servU installed.

4. I look in services and find DNS in Services (not DNS Server as you would expect).

I then open Task Manager and found nothing at first. Then scrolling down through services again, amidst the several svchost.exe that you normally find on a server, I found scvhost.exe. Luckily it was sitting right on top of one of the svchost.exe's or I would have probably never found it. Those sneaky crackers!

5. Going through the files, I found scvhost.exe under system32 along with tlist.exe and kill.exe, two helpful hacker tools from the Windows Resource Kit (also known as the hacker's best friend). On examination of the hidden Recycler director (not Recycle Bin) I found a yellow folder with the same type of name you would expect for the blue trash cans that belong there, namely S-1-5-21-141173530-1530468107-900494708-1500. There I found 10 subfolders destined to be the warez folders but currently containing nothing but sfind.exe, another great tool from SysInternals. It seems the black-hats and white-hats like the same tools but for different reasons.

6. I run fport.exe, from Foundstone, and find:

9496 scvhost -> 35394 TCP c:\winnt\system32\scvhost.exe

7. I run Sysinternal strings.exe program on scvhost.exe and find that it is iroffer IRC server and calls c_28619.nls which is servU that has been reprogrammed to appear as DNS in services.

8. Lastly I find a file called dos.exe with the same date/time as scvhost.exe. Run strings.exe on dos.exe and find it is really pjam2 udp denial of service program and is the program that was eating up our bandwidth. I felt bad that someone was getting hammered with a t3 but felt good that we got it off-line within 2 hours.

9. Find and delete adm1n under administrators group.

10. Because this was a newly installed server, just re-format and re-install to insure it is a clean machine.

So, it turns out that this server was destined to be a warez server but, probably due to the high bandwidth available to her, the cracker could not help but immediately use this DDoS zombie as a soldier in her cyberarmy of IRC bots to spew millions of rounds of UDP bullets at her enemy.

This is the closest crack I've come across to characterizing as evil. She became so drunk with power, she could not contain herself and ended up losing the best resource she ever had. This is actually representative of the many bot cyberarmies out there. CERT says it is monitoring 5 very large armies, one currently with 200,000 owned bots. The Internet is rapidly becoming a 21st Century battleground.

Cracker Psychology

It seems that over the past 8 years, we are coming full circle on the psychology of cracking. My first experience with SYN-flooding was from a business competitor trying to bring down my business. Cracking then seemed to be taken

over by script-kiddies who were given the tools by true hackers, trade them among themselves on IRC and then use them to crack servers and workstations around the world.

That trend now seems to be reversing with hackers keeping their tools closer to their vests and turning into crackers with financial and political incentives in mind. This is evidenced by the 0-day exploits that are emerging. It used to be that a vulnerability would be discovered, and an exploit would come out a few days or weeks after it was publicized on bugtraq, and hopefully after the vendor came out with a patch (that too many sysadmins never applied.) Today, we are seeing cracked machines being discovered with a 0-day exploit of a vulnerability that no-one knew about, and then the vendor develops the patch.

I believe the recent WebDav vulnerability was just such an occurrence. A military customer of ISS X-Force was found with a compromised server, and ISS brought in Microsoft who discovered the actual vulnerability was in the ntdll.dll file invoked by WebDav and worked feverishly through the weekend to develop the patch for release by Monday.

Conclusion

Although I have focused on Microsoft Windows OS's in this paper, all network-based operating systems have vulnerabilities that can be exploited to compromise a machine. I have seen both UNIX and Linux boxes owned, though not nearly as many as Windows. I am definitely not a Microsoft-basher as I have an MCSE in addition to my CNE as well as CISSP and CCDA and try to make recommendations to clients based on need, cost, and functionality in addition to security.

But in my experience of 8 years in the information security field, I have found all encountered Operating Systems remotely cracked except two, Netware and AS-400. But there are even documented cases of these happening in small numbers. So network administrators must practice the advice of caveat emptor. And this does not just entail the rigor of patch, patch, patch and put a firewall and IDS up, although this is a security baseline.

Security is an ongoing, human process; not a product. Sysadmins need to continually inform and educate themselves, from layer 1 hardware to layer 3 and 4 protocols to layer 7 applications. In addition they need to develop their people-skills so they can enroll their users as soldiers in their cyber-army. Instead of looking at users as problems, they need to teach them to be solutions. Once users understand the need for security and how to do it, they can be the sysadmins eyes and ears instead of their ball and chain.

The Internet today is like the wild-west was 200 years ago. There is no law out there, just the white-hats and the black-hats at 50 paces at high-noon on main street. The white-hat better be quick on the draw and a sure shot.

List of References:

[1] Eric Raymond et al, **The on-line hacker Jargon File, version 4.2.0**
January 31, 2000 - http://www.eps.mcgill.ca/jargon/jargon.html

[2] Larry Boettger, **The Morris Worm**
December 24, 2000 - http://www.sans.org/rr/malicious/morris.php

[3] Michael Hauben, **History of ARPANET**
May 1997 - http://www.columbia.edu/~rh120/ch106.x07

[4] Bill Cheswick and Hal Burch, **Internet Mapping Project**
June 1999 - http://research.lumeta.com/ches/map/

[5] Avi Freedman, **Stopping the Flooding**
March 1997 - http://avi.freedman.net/bw/mar97.html

[6] Adam L. Rice, **Defending Networks from SYN Flooding In Depth**
December 6, 2000 - http://rr.sans.org/threats/SYN_flood.php

[7] Steve Gibson, **Distributed Reflection Denial of Service**
February 22, 2002 - http://grc.com/dos/drdos.htm

[8] Dave Dittrich, **Distributed Denial of Service (DDoS) Attacks/tools**
March 5, 2003 - http://staff.washington.edu/dittrich/misc/ddos/

[9] Cisco Systems, **Characterizing and Tracing Packet Floods**
February 20, 2003 - http://www.cisco.com/warp/public/707/22.html