

# Hacking for Dummies

*(Access to other peoples systems made simple – & some extra database lore).*

## Introduction

The author is not responsible for any abuse of this information. It is intended for educational use only. You may be quite shocked at how vulnerable you are! As an afterthought I added a section on database access due to a number of requests.

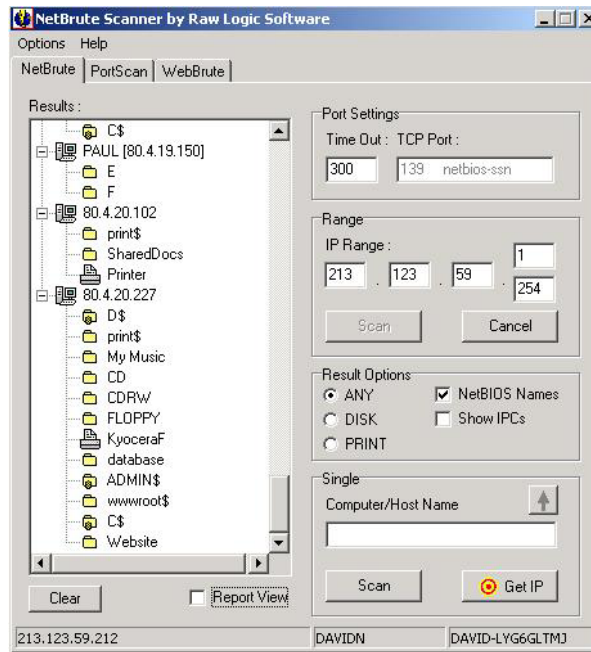
The majority of successful attacks on computer systems via the Internet can be traced to exploitation of security flaws in software and operating systems. These few software vulnerabilities account for the majority of successful attacks, simply because attackers are opportunistic – taking the easiest and most convenient route. They exploit the best-known flaws with the most effective and widely available attack tools. Most software, including operating systems and applications, comes with installation scripts or installation programs. The goal of these installation programs is to get the systems installed as quickly as possible, with the most useful functions enabled, with the least amount of work being performed by the administrator. To accomplish this goal, the scripts typically install more components than most users need. The vendor philosophy is that it is better to enable functions that are not needed, than to make the user install additional functions when they are needed. This approach, although convenient for the user, creates many of the most dangerous security vulnerabilities because users do not actively maintain and patch software components they don't use. Furthermore, many users fail to realize what is actually installed, leaving dangerous samples on a system simply because users do not know they are there. Those unpatched services provide paths for attackers to take over computers.

For operating systems, default installations nearly always include extraneous services and corresponding open ports. Attackers break into systems via these ports. In most cases the fewer ports you have open, the fewer avenues an attacker can use to compromise your network. For applications, default installations usually include unneeded sample programs or scripts. One of the most serious vulnerabilities with web servers is sample scripts; attackers use these scripts to compromise the system or gain information about it. In most cases, the system administrator whose system is compromised did not realize that the sample scripts were installed. Sample scripts are a problem because they usually do not go through the same quality control process as other software. In fact they are shockingly poorly written in many cases. Error checking is often forgotten and the sample scripts offer a fertile ground for buffer overflow attacks.

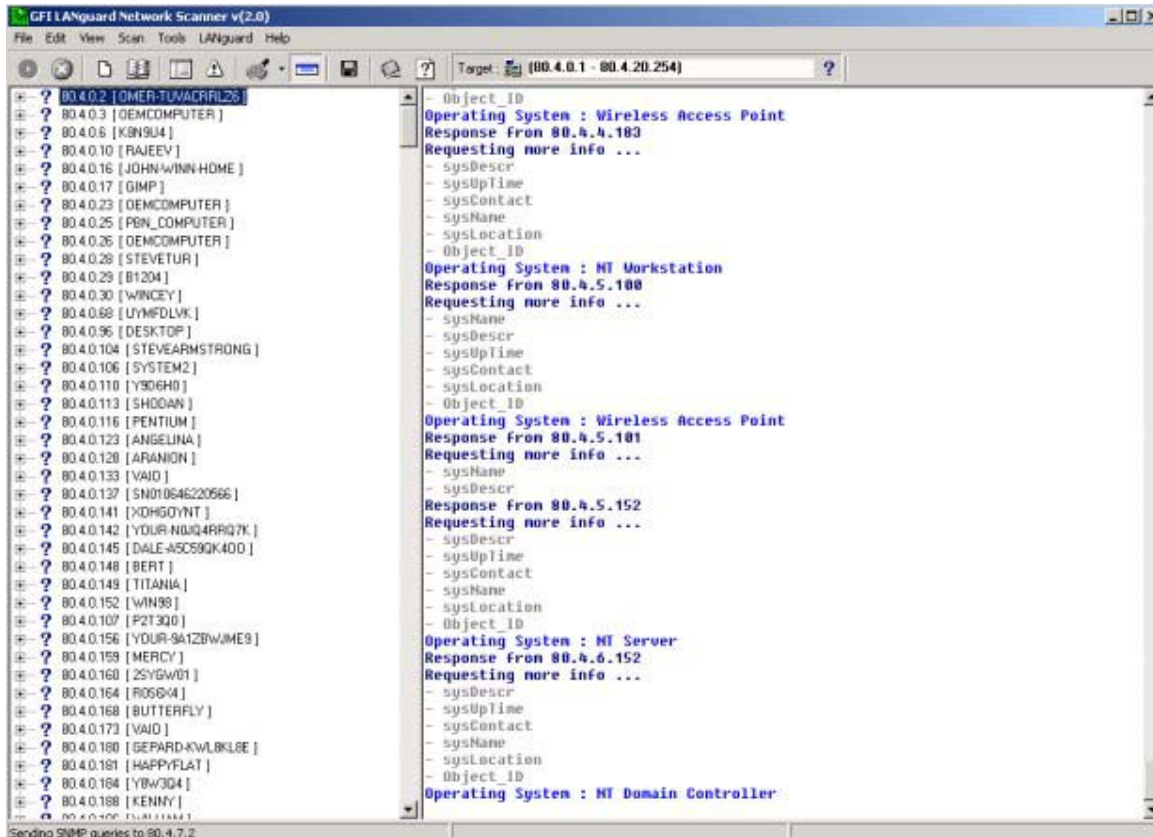
The simplest means to gain access to a system is by simple file and printer sharing. This is used to allow others on say, a home local area network share files, printers, and internet connections. If the computer having file and printer sharing enabled, this in fact allows these resources to be shared, and on offer, to the entire internet! This is largely due to the fact that Netbios was originally intended for use on local area networks (LAN's), where trusted sharing of resources made sense for many reasons. It was never intended to 'go global'.

First, search using a Netbios scanner, for a system with sharing enabled. A program such as Netbrute, by Raw Logic Software, is ideal. These programs can help the would-be hacker, as well as the network administrator. Run the scan over a subnet at a time, for example an IP address range from 80.1.1.1 to 80.1.1.254. Choose a system which has, preferably, it's whole hard disk

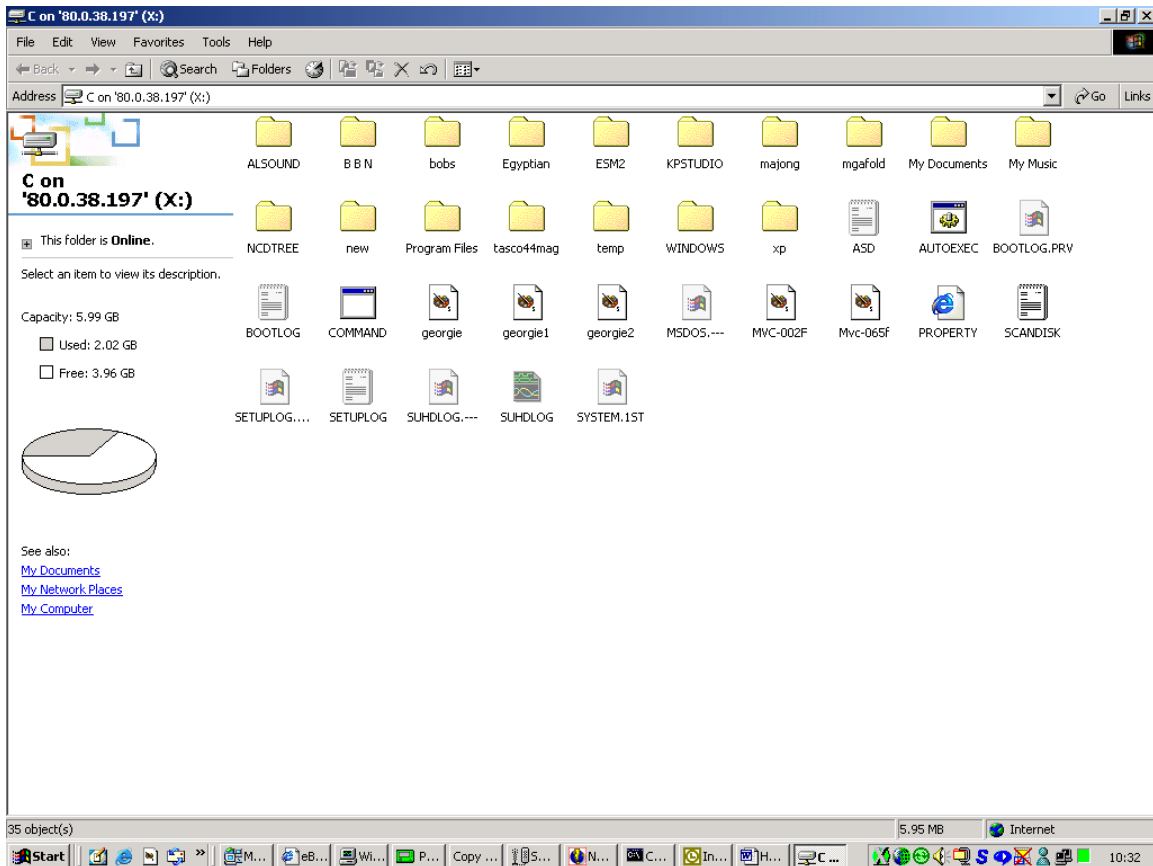
shared (You'd be amazed at some peoples stupidity!!!), this shows up as a result such as [\\80.5.7.2\C](http://80.5.7.2\C) or similar. Simply copy & paste this link into the address bar of Windows Explorer, and hit enter! This is a screenshot of Netbrute in operation:



For more comprehensive information, use a utility such as Languard Network Scanner. This returns a wealth of information such as domain names, login names, and more. Here is a shot of this in use:



Need I say more? If you find a system where the root directory of C: is shared, then on Windows 9.X systems, you'll be able to access the whole of the hard drive. On Windows NT/2000 systems, you will have only access as according to NTFS file access permissions. Here is a screenshot of Windows Explorer pointed at the root directory:



You can even map it to a network drive (use tools > map network drive), it's as easy as that!

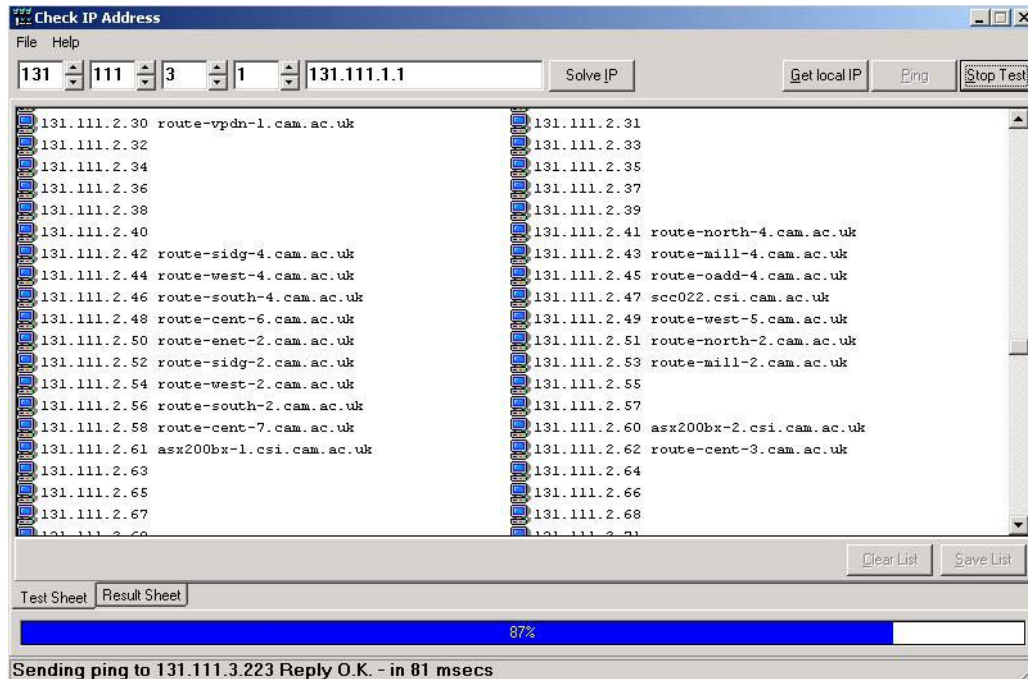
For best results, I recommend choosing systems with 'better than modem' connections. If you don't know where to start, try your own IP address. To get this, do the following:

- For Windows 9.X, go to start > Run and type 'Winipcfg' to get your IP address.
- For Windows NT/2000, got to start > programs > accessories > command prompt, and type 'ipconfig'.

This will return your IP address. If you are using a dialup connection, you will need to connect first. For 'always on' cable connection, omit this step. Then run your scan over the subnet; e.g. if your IP address is 164.99.34.212 then try a scan from 164.99.34.1 to 164.99.34.254. This should be enough to get you started. Have fun...

### IP Scanning

This simple scan simply pings a range of IP addresses to find which machines are alive. Note that more sophisticated scanners will use other protocols (such as an SNMP sweep) to do the same thing. This is a very simple technique which requires little explanation. It is however, useful for the domain name to be returned also.



## Port Scanning

This section introduces many of the techniques used to determine what ports (or similar protocol abstraction) of a host are listening for connections. These ports represent potential communication channels. Mapping their existence facilitates the exchange of information with the host, and thus it is quite useful for anyone wishing to explore their networked environment, including hackers. Despite what you have heard from the media, the Internet is NOT exclusively reliant on TCP port 80, used by hypertext transfer protocol (HTTP). Anyone who relies exclusively on the WWW for information gathering is likely to gain the same level of proficiency as your average casual surfer. This section is also meant to serve as an introduction to the art of port scanning, in which a host system can be persuaded to yield up its secrets. To accomplish this, you need to obtain a port scanner. There are many available both for free or for a small fee. It should have all these features:

- dynamic delay time calculations: Some scanners require that you supply a delay time between sending packets. Well how should I know what to use? You can always ping them, but that is a pain, and plus the response time of many hosts changes dramatically when they are being flooded with requests. For root users, the primary technique for finding an initial delay is to time the internal “ping” function. For non-root users, it times an attempted connect() to

a closed port on the target. It can also pick a reasonable default value. Again, people who want to specify a delay themselves can do so with -w (wait), but you shouldn't have to.

- Retransmission: Some scanners just send out all the query packets, and collect the responses. But this can lead to false positives or negatives in the case where packets are dropped. This is especially important for “negative” style scans like UDP and FIN, where what you are looking for is a port that does NOT respond.
- Parallel port scanning: Some scanners simply scan ports linearly, one at a time, until they do all 65535. This actually works for TCP on a very fast local network, but the speed of this is not

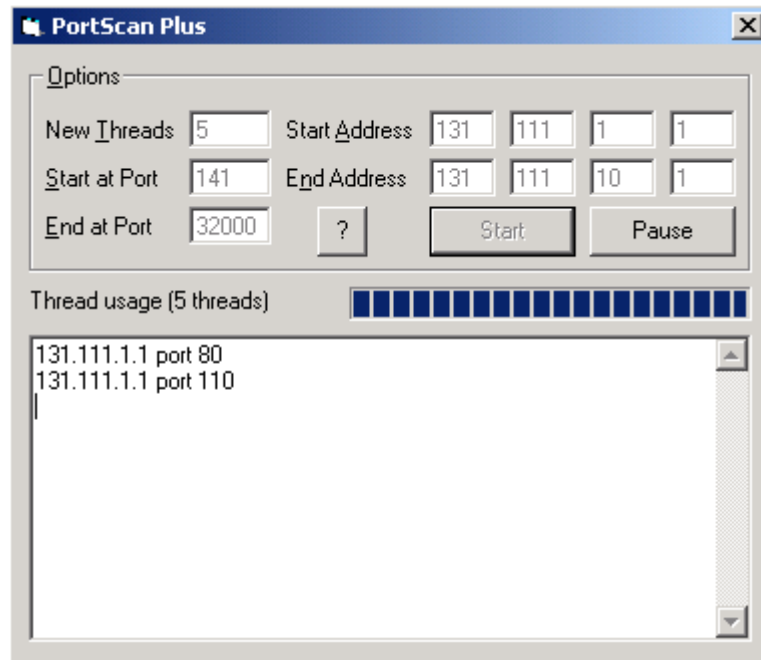
at all acceptable on a wide area network like the Internet. It is best to use non-blocking i/o and parallel scanning in all TCP and UDP modes. Flexible port specification: You don't always want to scan all 65535 ports! Also, the scanners which only allow you to scan ports 1 - N often fall short of my need. The scanner should allow you to specify an arbitrary number of ports and ranges for scanning. For example, '21-25,80-113' is often useful if you are only probing the most frequently running services.

- Flexible target specification: You may often want to scan more than one host, and you certainly don't want to list every single host on a large network! It is useful to scan, say a subnet at once, e.g. 131.111.11.0 – 131.111.11.254.
- Detection of down hosts: Some scanners allow you to scan large networks, but they waste a huge amount of time scanning 65535 ports of a dead host! Annoying! You are advised to choose a scanner which allows timeout intervals to be adjusted.
- Detection of your IP address: For some reason, a lot of scanners ask you to type in your IP address as one of the parameters. You don't want to have to 'ifconfig' and figure out your current IP address every time you connect. Of course, this is better than the scanners I've seen which require recompilation every time you change your address! If you are using a cable 'always on' connection, you may find that the IP address remains constant, as in my own case.

There are actually 65536 ports in all; however by convention services with which we are most familiar tend to use the lower numbers. Here are a few:

FTP	21
Telnet	23
SMTP	25
HTTP	80
POP3	110

Although the services can be configured to use other ports, this is very unusual. Ports above 1024 tend to be used by the operating system. Essentially a port scanner sends packets of data on each port in turn, and listens for replies to determine what services are running. A detailed list is available at the end of the document. This is an example of a simple port scanner in use:



### Network Topology Views

This may be useful on occasion. It provides a graphical view of the resources on your network. For example, it may show which systems are behind a firewall, and which routers are on-line.

A 'network viewer'.

### Packet Sniffing

A packet sniffer or protocol analyser is a wire-tap device that plugs into computer networks and eavesdrops on the network traffic. Like a telephone wiretap allows one to listen in on other people's conversations, a "sniffing" program lets someone listen in on computer conversations. However, computer conversations consist of apparently random binary data. Therefore, network wiretap programs also come with a feature known as "protocol analysis", which allow them to "decode" the computer traffic and make sense of it. Sniffing also has one advantage over telephone wiretaps: many networks use "shared media". This means that you don't need to break into a wiring closet to install your wiretap, you can do it from almost any network connection to eavesdrop on your neighbours. This is called a "promiscuous mode" sniffer. However, this "shared" technology is moving quickly toward "switched" technology where this will no longer be possible, which means you will have to actually tap into the wire.

There is no single point on the Internet where it is possible to 'see' all of the traffic. The connectivity of the Internet looks similar a fisherman's net. Traffic flows through a mesh, and no single point will see it all! The Internet was built to withstand a nuclear attack—and to survive any "single point of failure". This likewise prevents any single point of packet sniffing. Consider this situation: you have two machines in your own office talking to each other, and both are on the Internet. They take a direct route of communication, and the traffic never goes across the outside public portion of the Internet. Any communication anywhere in the net follows a similar "least-cost-path" principle.

Ethernet was built around a "shared" principle: all machines on a local network share the same wire. This implies that all machines are able to "see" all the traffic on the same wire. Therefore,

Ethernet hardware is built with a “filter” that ignores all traffic that doesn’t belong to it. It does this by ignoring all frames whose MAC address doesn’t match their own. A wiretap program effectively turns off this filter, putting the Ethernet hardware into “promiscuous mode”. Thus, Mark can see all the traffic between Alice and Bob, as long as they are on the same Ethernet wire.

Since many machines may share a single Ethernet wire, each must have an individual identifier. This doesn’t happen with dial-up modems, because it is assumed that any data you send to the modem is destined for the other side of the phone line. But when you send data out onto an Ethernet wire, you have to be clear which machine you intend to send the data to. Sure, in many cases today there are only two machines talking to each other, but you have to remember that Ethernet was designed for thousands of machines to share the same wire. This is accomplished by putting a unique 12-digit hex number in every piece of Ethernet hardware. To really understand why this is so important, you might want to review the information in section 5.4 below. Ethernet was designed to carry other traffic than just TCP/IP, and TCP/IP was designed to run over other wires (such as dial-up lines, which use no Ethernet). For example, many home users install “NetBEUI” for File and Print Sharing because it is unrelated to TCP/IP, and therefore hackers from across the Internet can’t get at their hard-drives.

Raw transmission and reception on Ethernet is governed by the Ethernet equipment. You just can’t send data raw over the wire, you must first do something to it that Ethernet understands. In much the same way, you can’t stick a letter in a mailbox, you must first wrap it in an envelope with an address and stamp.

Following is a brief explanation how this works:

Alice has IP address: 10.0.0.23

Bob has IP address: 192.168.100.54

In order to talk to Bob, Alice needs to create an IP packet of the form 10.0.0.23-->192.168.100.54. As the packet traverses the Internet, it will be passed from router-to-router. Therefore, Alice must first hand off the packet to the first router. Each router along the way will examine the destination IP address (192.168.100.54) and decide the correct path it should take.

All Alice knows about is the local connection to the first router, and Bob’s eventual IP address. Alice knows nothing about the structure of the Internet and the route that packet will take. Alice must talk to the router in order to send the packet. She uses the Ethernet to do so. An Ethernet frame looks like the following:

What this means is that the TCP/IP stack in Alice’s machine might create a packet that is 100 bytes long (let’s say 20 bytes for the IP info, 20 bytes for the TCP info, and 60 bytes of data). The TCP/IP stack then sends it to the Ethernet module, which puts 14 bytes on the front for the destination MAC address, source MAC address, and the ethertype 0x0800 to indicate that the other end’s TCP/IP stack should process the frame. It also attaches 4-bytes on the end with a checksum/CRC (a validator to check whether the frame gets corrupted as it goes across the wire). The adapter then sends the bits out onto the wire. All hardware adapters on the wire see the frame, including the ROUTER’s adapter, the packet sniffer, and any other machines. Proper adapters, however, have a hardware chip that compares the frame’s “destination MAC” with its own MAC address. If they don’t match, then it discards the frame. This is done at the hardware level, so the machine the adapter is attached to is completely unaware of this process.

When the ROUTER Ethernet adapter sees this frame, it reads it off the wire and removes the leading 14-bytes and the trailing 4-bytes. It looks at the 0x0800 ethertype and decides to send it to the TCP/IP stack for processing (which will presumably forward it to the next router in the chain toward the destination). In the above scenario, only the ROUTER machine is supposed to see the Ethernet frame, and all other machines are supposed to ignore it. The wiretap, however, breaks the rules and copies the frame off the network, too.



To see your own Ethernet address, do the following;

Win9x: Run the program “winipcfg.exe”. It will tell you.

WinNT/2000: Run the program “ipconfig /all” from the command-line. It will show the MAC address for your adapters. This is an example result:

Windows NT IP Configuration

Host Name . . . . . : sample.robertgraham.com

DNS Servers . . . . . : 192.0.2.254

Node Type . . . . . : Hybrid

NetBIOS Scope ID. . . . . :

IP Routing Enabled. . . . . : Yes

WINS Proxy Enabled. . . . . : No

NetBIOS Resolution Uses DNS : No

Ethernet adapter SC12001:

Description . . . . . : DEC DC21140 PCI Fast Ethernet Adapter

Physical Address. . . . . : 00-40-05-A5-4F-9D

DHCP Enabled. . . . . : No

IP Address. . . . . : 192.0.2.160

Subnet Mask . . . . . : 255.255.255.0

Default Gateway . . . . . : 192.0.2.1

Primary WINS Server . . . . : 192.0.2.253

Linux

Run the program “ifconfig”. Here is a sample result:

eth0 Link encap:Ethernet HWaddr 08:00:17:0A:36:3E

inet addr:192.0.2.161 Bcast:192.0.2.255 Mask:255.255.255.0

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:1137249 errors:0 dropped:0 overruns:0

TX packets:994976 errors:0 dropped:0 overruns:0

Interrupt:5 Base address:0x300

Solaris: Use the “arp” or “netstat -p” command, it will often list the local interface among the ARP entries.



the underlying binary data. This is an explanation of the hexadecimal numbering system:

0000 = 0 0001 = 1 0010 = 2 0011 = 3  
0100 = 4 0101 = 5 0110 = 6 0111 = 7  
1000 = 8 1001 = 9 1010 = A 1011 = B  
1100 = C 1101 = D 1110 = E 1111 = F

In other words, when you encounter the hexadecimal digit “B”, you should immediately visualize the bit pattern “1011” in your head. It is much like memorizing multiplication tables as a kid, memorizing this table will serve much the same purpose. Hexadecimal is often preceded by a special character(s). For example, when you see the number “12”, is this “twelve” (decimal) or “eighteen” (hexadecimal)? If it is hex, it is often written as either “0x12”, “x12”, or “\$12”. The former is the preferred version, since that is how many programming languages represent it. Naturally, this isn’t needed for hex dumps because the fact we are showing hex is pretty much assumed. Computers represent everything as numbers. This means the text you are reading right now is represented as numbers within the computer. ASCII is one such representation. In ASCII, the letter ‘A’ is represented by the number 65, or in hex, 0x41. The letter ‘B’ is represented by the number 66/0x42. And the process continues for all characters, numbers, punctuation, and so forth. If you look at the normal (English) keyboard you will count 32 punctuation characters, 10 decimal digits, 26 letters, and 26 more letters when you take into account UPPER/lower case. This comes to 94 different characters. In binary, you need 7-bits to represent that number of combinations. This maps nicely onto the standard 8-bit bytes used in computers, with room left over. In hex dumps, note that the ASCII column contains lots of periods. A byte has 256 combinations, but we can only view 94 of them. Any character that is not one of these 94 visible characters is shown as a period.

Anyhow, if you want to try packet sniffing, I hope I have now provided the information you need to get started. You can download a packet sniffer free from the web as either shareware or freeware. Give it a go! By now, you must be feeling that there is a good chance that your boss may well have been snooping on your use of the corporate LAN and/or the internet all along! Is there no such thing as privacy at work nowadays? If you have a score to settle, the next section is for you...

### *Statistical Databases*

This may seem rather a departure from the ‘domestic’ hacking scene. But on reflection of some queries I have recently received relating to corporate databases, particularly relating to salary and employment details, I decided to give this topic a mention.

Have you ever wanted to somehow, obtain from your employer’s database, details relating to the personnel department? In this dreadful world of job insecurity and appraisal schemes, the author has just cause to explain a possible means to learn employer’s secrets.

A statistical database is, in its simplicity, a store of information relating to the infrastructure of entire organisations. This includes personal and employee details. These systems are implemented by means of Microsoft Access, MYSQL and other similar software, but what they all have in common is that one fact must be stored in one place. This is vital to ensure that queries return unique results. Please note that, in order to use this information successfully, a working knowledge of SQL (Structured Query Language) and relational algebra, is assumed. Some operand details are provided; however please note that this is not a SQL reference manual! This is a huge topic. I am simply suggesting possible means by which they may be manipulated in order to yield up details to which the database administrator has forbidden you access. The methods of trying to bypass access restrictions either may or may not work on all systems; the author merely

states that they have been successfully tried with success on *some* experimental databases.

### Hacking a Statistical Database

‘Views’ are used by a database administrator in order to hide certain data from those who do not need access to it according to their job description. For example, take this simple database for a small company having 10 employees:

Fname	Lname	Sex	dependen ts	occupatio n	Salary	Tax	audit
John	Harris	M	3	Program mer 25k		5k 3	
Lisa	White	F	2	Receptio nist 15k		3k 0	
Alison	Baker	F	0	Program mer	25k	5k	1
Emma	Foster	F	2	Secretary	13k	2.5k 1	
Steve	Smith	M	2	Manager	30k	6k 0	
Ann	Reid	F	1	Clerk	25k	5.5k 0	
Micheal	Roberts	M	0	Secretary	12k	2k	0
Tom	Reynolds M		3	Porter	11k	2k 0	
Pauline	Blackma n F		4	Program mer	18k	3.5k 1	
Sandra	Moore	F	1	Program mer	21k	4k	1

Suppose you wanted to find out John Harris’s salary. However, you do not have access to the salary and tax columns, as your administrator has excluded you from this view, as company policy states that only the personel department need access to this data. The key is not accessible to users. However, anyone with a limited knowledge of relational algebra can still get the information they seek...

We must arm ourselves with what we do know about John. We know that he is male and is a programmer. Without any protection other than the view set by the database administrator, these queries will flush out his salary:

```
SELECT COUNT (*) FROM Stats
WHERE sex = 'M' AND Occupation = 'Programmer'
Response 1
We have a single male programmer!
SELECT Sum(salary) Sum(tax) FROM Stats
WHERE Sex = 'M' AND occupation = 'Programmer'
```

Response 25k, 5k

We have found John's salary out. This single tuple attack is unlikely to work as, for security the administrator may have ruled that a query must say, more than one tuple. Therefore a single subject cannot be weeded out as before. However the multi-tuple manipulation can counter this as follows.

```
SELECT COUNT (*) FROM Stats
```

Response 10

```
SELECT COUNT (*) FROM Stats
```

```
WHERE NOT (sex = 'M' AND occupation = 'Programmer')
```

Response 9 (10 - 1 = 9)

```
SELECT Sum(salary) Sum(tax) FROM Stats
```

Response 195k, 38.5k

```
SELECT Sum(salary) Sum(tax) FROM Stats
```

```
WHERE NOT Sex = 'M' AND occupation = 'Programmer'
```

Response 170k, 33.5k

So  $195 - 170 = 25$ ,  $38.5 - 33.5 = 5$

Answer = 25k, 5k

We have still got John's salary! As the response in each case contained more than one tuple, it passed as an admissible query!

### **The individual tracker approach**

This method utilises predicates about John to construct queries.

```
SELECT COUNT (*) FROM Stats
```

```
WHERE sex = 'M'
```

Response 4

So there exist 4 males on the database.

```
SELECT COUNT (*) FROM Stats
```

```
WHERE sex = 'M' AND NOT (occupation = 'programmer')
```

Response 3

So there is only 1 male programmer.

```
SELECT Sum(salary) Sum(tax) FROM Stats
```

```
WHERE Sex = 'M'
```

Response 78k, 15k

```
SELECT Sum(salary) Sum(tax) FROM Stats
```

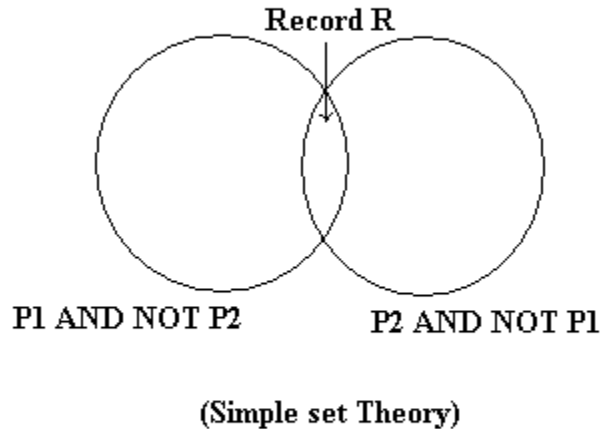
```
WHERE Sex = 'M' AND NOT (occupation = 'programmer')
```

Response 53k, 10k

So  $78 - 53 = 25$  and  $15 - 10 = 5$

Result 25k, 5k

So as before, we have John's salary. If we have a predicate about a specific record, i.e. John is male AND a programmer, we can formulate queries to obtain the results we wish to obtain. This can be summed up as P1 AND P2. The predicate P1 AND NOT P2 can be used as a tracker for that individual record.



### Hardware Tricks

For the hacker with some knowledge of computer hardware and general electronics, and who is prepared to mess about with circuit diagrams, a soldering iron and perhaps a voltmeter, logic probe or oscilloscope, still further possibilities open up. One of the most useful bits of kit consists of a small cheap radio receiver (MW/AM band), a microphone and a tape recorder. Radios in the vicinity of computers, modems and telephone lines can readily pick up the chirp chirp of digital communications without the need of carrying out a physical phone 'tap'. Alternatively, an inductive loop with a small low-gain amplifier in the vicinity of a telephone or line will give you a recording you can analyse later at your leisure.

By identifying the pairs of tones being used, you can separate the caller and the host. By feeding the recorded tones onto an oscilloscope display you can freeze bits, 'characters' and 'words'; you can strip off the start and stop bits and, with the aid of an ASCII-to-binary table, examine what is happening. With experience it is entirely possible to identify a wide range of protocols simply from the 'look' of an oscilloscope. A cruder technique is simply to record and playback sign-on sequences; the limitation is that, even if you manage to log on, you may not know what to do afterwards. Listening on phone lines is of course a technique also used by some sophisticated robbers. In 1982 the Lloyds Bank Holborn branch was raided; the alarm did not ring because the thieves had previously recorded the 'all-clear' signal from the phone line and then, during the break-in, replayed the recording up the line to the alarm monitoring apparatus. Sometimes the hacker must devise ad hoc bits of hardware trickery in order to achieve his ends. Access has been obtained to a well-known financial prices service largely by stringing together a series of simple hardware skills. The service is available mostly on leased lines, as the normal vagaries of dial-up would be too unreliable for the City folk who are the principal customers.

However, each terminal also has an associated dial-up facility, in case the leased line should go down; and in addition, the same terminals can have access to Prestel. Thus the hacker thought that it should be possible to access the service with ordinary viewdata equipment instead of the special units supplied along with the annual subscription. Obtaining the phone number was relatively easy: it was simply a matter of selecting manual dial-up from the appropriate menu, and listening to the pulses as they went through the regular phone.

The next step was to obtain a password. The owners of the terminal to which the hacker had access did not know their ID; they had no need to know it because it was programmed into the terminal and sent automatically. The hacker could have put micro 'back-to-front' across the line and sent a ENQ to see if an ID would be sent back. Instead he tried something less obvious.

The terminal was known to be programmable, provided one knew how and had the right type of keyboard. Engineers belonging to the service had been seen doing just that. How could the hacker acquire 'engineer' status? He produced the following hypothesis: the keyboard used by the service's customers was a simple affair, lacking many of the obvious keys used by normal terminals; the terminal itself was manufactured by the same company that produced a range of editing terminals for viewdata operators and publishers. Perhaps if one obtained a manual for the editing terminal, important clues might appear. A suitable photocopy was obtained and, lo and behold, there were instructions for altering terminal IDs, setting auto-diallers and so on.

## *Linux & Unix for beginners*

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. But today Unix in all its flavours 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.

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 study. Even if you are not breaking the law, a shell log file that shows you doing lots of hacking 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 very good indeed.

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. 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 311te 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 stuff, 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](http://sunsite.unc.edu:/pub/Linux/docs/FAQ).

In the directory [/pub/Linux/docs](http://sunsite.unc.edu:/pub/Linux/docs) on [sunsite.unc.edu](http://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](http://sunsite.unc.edu) Web site at:

[/pub/Linux/docs/HOWTO](http://sunsite.unc.edu:/pub/Linux/docs/HOWTO). The directory [/pub/Linux/docs/LDP](http://sunsite.unc.edu:/pub/Linux/docs/LDP) contains the current set of LDP manuals. You can get "Linux Installation and Getting Started" from [sunsite.unc.edu](http://sunsite.unc.edu:/pub/Linux/docs/LDP/install-guide) in [/pub/Linux/docs/LDP/install-guide](http://sunsite.unc.edu:/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](http://tsx11.mit.edu) and [sunsite.unc.edu/pub/linux](http://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)."

What 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/cert_advisories/CA-94:01.network.monitoring.attacks)  
[ftp://info.cert.org/pub/tech\\_tips/root\\_compromise](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](mailto:cyberkid@usa.net)) hosts a channel called #LinuxHelp on the Undernet IRC server.

### Brief SQL Reference

To get all columns of a table without typing all column names, use: `SELECT * FROM TableName`; To get the total number of tuples (rows): `SELECT Count(*) FROM EMPLOYEE`  
To get the total number of female employees in reception: `SELECT Count (*) FROM EMPLOYEE WHERE sex = 'm' AND Department = 'reception'`;

### **Relational Operators**

There are six Relational Operators in SQL, and after introducing them, we'll see how they're used: = Equal <> or != Not Equal < Less Than > Greater Than <= Less Than or Equal To >= Greater Than or Equal To

For example, if you wanted to see the EMPLOYEE ID NO's of those making at least, or over \$50,000, use the following:

```
SELECT EMPLOYEEIDNO FROM EMPLOYEESTATISTICSTABLE WHERE SALARY >= 50000;
```

Notice that the >= (greater than or equal to) sign is used, as we wanted to see those who made greater than \$50,000, or equal to \$50,000, listed together.

The *WHERE* description, `SALARY >= 50000`, is known as a *condition* (an operation which evaluates to True or False). The same can be done for text columns:

```
SELECT EMPLOYEEIDNO FROM EMPLOYEE STATISTICSTABLE WHERE POSITION = 'Manager';
```

This displays the ID Numbers of all Managers.

#### More Complex Conditions: Compound Conditions / Logical Operators

The *AND* operator joins two or more conditions, and displays a row only if that row's data satisfies **ALL** conditions listed (i.e. all conditions hold true). For example, to display all staff making over \$40,000, use:

```
SELECT EMPLOYEEIDNO
```

```
FROM EMPLOYEESTATISTICSTABLE
```

```
WHERE SALARY > 40000 AND POSITION = 'Staff';
```

The *OR* operator joins two or more conditions, but returns a row if **ANY** of the conditions listed hold true. To see all those who make less than \$40,000 or have less than \$10,000 in benefits, listed together, use the following query:

```
SELECT EMPLOYEEIDNO FROM EMPLOYEESTATISTICSTABLE WHERE SALARY < 40000 OR BENEFITS < 10000
```

AND & OR can be combined, for example:

```
SELECT EMPLOYEEIDNO
```

```
FROM EMPLOYEESTATISTICSTABLE
```

```
WHERE POSITION = 'Manager' AND SALARY > 60000 OR BENEFITS > 12000;
```

First, SQL finds the rows where the salary is greater than \$60,000 and the position column is equal to Manager, then taking this new list of rows, SQL then sees if any of these rows satisfies the previous AND condition or the condition that the Benefits column is greater than \$12,000. Subsequently, SQL only displays this second new list of rows, keeping in mind that anyone with Benefits over \$12,000 will be included as the OR operator includes a row if either resulting condition is True. Also note that the AND operation is done first. This is a law of Boolean algebra. This is analogous to

the principle of mathematics which state that 'multiplication and division take precedence over addition and subtraction'.

To perform OR's before AND's, like if you wanted to see a list of employees making a large salary (>\$50,000) or have a large benefit package (>\$10,000), and that happen to be a manager, use parentheses:

```
SELECT EMPLOYEEIDNO
```

```
FROM EMPLOYEESTATISTICSTABLE
```

```
WHERE POSITION = 'Manager' AND (SALARY > 50000 OR BENEFIT > 10000);
```

IN & BETWEEN

An easier method of using compound conditions uses *IN* or *BETWEEN*. For example, if you wanted to list all managers and staff:

***SELECT EMPLOYEEIDNO*** FROM EMPLOYEEESTATISTICSTABLE WHERE POSITION IN ('Manager', 'Staff'); or to list those making greater than or equal to \$30,000, but less than or equal to \$50,000, use:

***SELECT EMPLOYEEIDNO*** FROM EMPLOYEEESTATISTICSTABLE WHERE SALARY BETWEEN 30000 AND 50000;

To list everyone not in this range, try:

***SELECT EMPLOYEEIDNO*** FROM EMPLOYEEESTATISTICSTABLE WHERE SALARY NOT BETWEEN 30000 AND 50000; Similarly, NOT IN lists all rows excluded from the *IN* list. Additionally, NOT's can be thrown in with AND's & OR's, except that NOT is a unary operator (evaluates one condition, reversing its value, whereas, AND's & OR's evaluate two conditions), and that all NOT's are performed before any AND's or OR's.

**SQL Order of Logical Operations (each operates from left to right) 1. NOT 2. AND 3. OR**

### Using *LIKE*

If you wanted to see all people whose last names started with "L"; try: ***SELECT EMPLOYEEIDNO*** FROM EMPLOYEEESTATISTICSTABLE WHERE LASTNAME LIKE 'L%'; The percent sign (%) is used to represent any possible character (number, letter, or punctuation) or set of characters that might appear after the "L". To find those people with LastName's ending in "L", use '%L', or if you wanted the "L" in the middle of the word, try '%L%'. The '%' can be used for any characters in the same position relative to the given characters. NOT LIKE displays rows not fitting the given description. Other possibilities of using LIKE, or any of these discussed conditionals, are available, though it depends on what DBMS you are using; as usual, consult a manual for the available features on your system, or just to make sure that what you are trying to do is available and allowed. This disclaimer holds for the features of SQL that will be discussed below. This section is just to give you an idea of the possibilities of queries that can be written in SQL.

### Joins

In this section, we will only discuss *inner* joins, and *equijoins*, as in general, they are the most useful. For more information, refer to an SQL manual.

Good database design suggests that each table lists data only about a single *entity*, and detailed information can be obtained in a relational database, by using additional tables, and by using a *join*.

First, take a look at these example tables:

### AntiqueOwners

OwnerID	OwnerLastName	OwnerFirstName
01	Jones	Bill
02	Smith	Bob
15	Lawson	Patricia
21	Akins	Jane
50	Fowler	Sam

## Orders

**OwnerID ItemDesired** 02 Table 02 Desk 21 Chair 15 Mirror

## Antiques

**SellerID BuyerID Item** 01 50 Bed 02 15 Table 15 02 Chair 21 50 Mirror 50 01 Desk 01 21  
Cabinet 02 21 Coffee Table 15 50 Chair 01 15 Jewelry Box 02 21 Pottery 21 02 Bookcase 50 01  
Plant Stand

## Keys

First, let's discuss the concept of *keys*. A *primary key* is a column or set of columns that uniquely identifies the rest of the data in any given row. For example, in the AntiqueOwners table, the OwnerID column uniquely identifies that row. This means two things: no two rows can have the same OwnerID, and, even if two owners have the same first and last names, the OwnerID column ensures that the two owners will not be confused with each other, because the unique OwnerID column will be used throughout the database to track the owners, rather than the names.

A *foreign key* is a column in a table where that column is a primary key of another table, which means that any data in a foreign key column must have corresponding data in the other table where that column is the primary key. In DBMS-speak, this correspondence is known as *referential integrity*. For example, in the Antiques table, both the BuyerID and SellerID are foreign keys to the primary key of the AntiqueOwners table (OwnerID; for purposes of argument, one has to be an Antique Owner before one can buy or sell any items), as, in both tables, the ID rows are used to identify the owners or buyers and sellers, and that the OwnerID is the primary key of the AntiqueOwners table. In other words, all of this "ID" data is used to refer to the owners, buyers, or sellers of antiques, themselves, without having to use the actual names.

## Performing a Join

The purpose of these *keys* is so that data can be related across tables, without having to repeat data in every table—this is the power of relational databases. For example, you can find the names of those who bought a chair without having to list the full name of the buyer in the Antiques table...you can get the name by relating those who bought a chair with the names in the AntiqueOwners table through the use of the OwnerID, which *relates* the data in the two tables. To find the names of those who bought a chair, use the following query:

```
SELECT OWNERLASTNAME, OWNERFIRSTNAME
```

```
FROM ANTIQUEOWNERS, ANTIQUES
```

```
WHERE BUYERID = OWNERID AND ITEM = 'Chair';
```

Note the following about this query...notice that both tables involved in the relation are listed in the FROM clause of the statement. In the WHERE clause, first notice that the ITEM = 'Chair' part restricts the listing to those who have bought (and in this example, thereby owns) a chair. Secondly, notice how the ID columns are related from one table to the next by use of the BUYERID = OWNERID clause. Only where ID's match across tables and the item purchased is a chair (because of the AND), will the names from the AntiqueOwners table be listed. Because the joining condition used an equal sign, this join is called an *equijoin*. The result of this query is two names: Smith, Bob & Fowler, Sam.

*Dot notation* refers to prefixing the table names to column names, to avoid ambiguity, as follows:

```
SELECT ANTIQUEOWNERS.OWNERLASTNAME,  
ANTIQUEOWNERS.OWNERFIRSTNAME
```

```
FROM ANTIQUEOWNERS, ANTIQUES
```

```
WHERE ANTIQUES.BUYERID = ANTIQUEOWNERS.OWNERID AND ANTIQUES.ITEM  
= 'Chair';
```

As the column names are different in each table, however, this wasn't necessary.

### ***DISTINCT*** and Eliminating Duplicates

Let's say that you want to list the ID and names of **only** those people who have sold an antique. Obviously, you want a list where each seller is only listed once—you don't want to know how many antiques a person sold, just the fact that this person sold one (for counts, see the Aggregate Function section below). This means that you will need to tell SQL to eliminate duplicate sales rows, and just list each person only once. To do this, use the *DISTINCT* keyword.

First, we will need an equijoin to the AntiqueOwners table to get the detail data of the person's LastName and FirstName. However, keep in mind that since the SellerID column in the Antiques table is a foreign key to the AntiqueOwners table, a seller will only be listed if there is a row in the AntiqueOwners table listing the ID and names. We also want to eliminate multiple occurrences of the SellerID in our listing, so we use ***DISTINCT* on the column where the repeats may occur.**

To throw in one more twist, we will also want the list alphabetized by LastName, then by FirstName (on a LastName tie). Thus, we will use the *ORDER BY* clause:

```
SELECT DISTINCT SELLERID, OWNERLASTNAME, OWNERFIRSTNAME FROM  
ANTIQUES, ANTIQUEOWNERS WHERE SELLERID = OWNERID ORDER BY  
OWNERLASTNAME, OWNERFIRSTNAME;
```

In this example, since everyone has sold an item, we will get a listing of all of the owners, in alphabetical order by last name. For future reference (and in case anyone asks), this type of join is considered to be in the category of *inner joins*. Please note that by no means is this a complete reference!!! It is, however, a guide to the queries you will need to know in order to (hopefully) extract the data you seek. Have fun...

### *The 'Ping of Death'*

Essentially, it is possible to crash, reboot or otherwise kill a large number of systems by sending a ping of a certain size from a remote machine. This is a serious problem, mainly because this can be reproduced very easily, and from a remote machine. The attacker needs to know nothing about the machine other than its IP address. Be afraid.

It's very easy to exploit - basically, some systems don't like being pinged with a packet greater than 65536 bytes (as opposed to the default 64 bytes).

An IP datagram of 65536 bytes is illegal, but possible to create owing to the way the packet is fragmented (broken into chunks for transmission). When the fragments are reassembled at the other end into a complete packet, it overflows the buffer on some systems, causing a reboot, panic

or hang, but sometimes even having no effect at all.

Most implementations of ping won't allow an invalid datagram like this to be sent. Among the exceptions are Windows '95 and NT, although they are certainly not the only ones...

IP packets as per RFC-791 can be up to 65,535 ( $2^{16}-1$ ) octets long, which includes the header length (typically 20 octets if no IP options are specified. An ICMP ECHO request "lives" inside the IP packet, consisting of eight octets of ICMP header information (RFC-792) followed by the number of data octets in the "ping" request. Hence the maximum allowable size of the data area is  $65535 - 20 - 8 = 65507$  octets.

Note that it is possible to send an illegal echo packet with more than 65507 octets of data due to the way the fragmentation is performed. The fragmentation relies on an offset value in each fragment to determine where the individual fragment goes upon reassembly. Thus on the last fragment, it is possible to combine a valid offset with a suitable fragment size such that  $(\text{offset} + \text{size}) > 65535$ . Since typical

machines don't process the packet until they have all fragments and have tried to reassemble it, there is the possibility for overflow of 16 bit internal variables, which can lead to system crashes, reboots, kernel dumps and the like. The problem can be exploited by anything that sends an IP datagram - probably the most fundamental building block of the net. Not only ICMP echo, but TCP, UDP and (apparently) even new style IPX can be used to hit machines where it hurts. This bug is extremely easy to exploit. Users are already trying it out "just to see if it works"!

### Port Numbers and Services

This data is from Internet Assigned Numbers Authority (IANA). IANA maintains the Assigned Numbers RFC. The entries in this file are in the same format as found in a standard Berkeley UNIX /etc/services file. There are also links between the protocol and services names, and their respective RFCs (their standard documentation). This file has two sections:

Well known Port Numbers: port numbers that IANA assigns  
Registered Port Numbers: port numbers that IANA does not assign. This provides a list of which ports are used by which services. There really is more to the net than HTTP alone!

#### WELL KNOWN PORT NUMBERS

The Well Known Ports are controlled and assigned by the IANA and on most systems can only be used by system (or root) processes or by programs executed by privileged users. Ports are used in the TCP [RFC793] to name the ends of logical connections which carry long term conversations. For the purpose of providing services to unknown callers, a service contact port is defined. This list specifies the port used by the server process as its contact port. The contact port is sometimes called the "well-known port".

To the extent possible, these same port assignments are used with the UDP [RFC768].

The assigned ports use a small portion of the possible port numbers. For many years the assigned ports were in the range 0-255. Recently, the range for assigned ports managed by the IANA has been expanded to the range 0-1023.

[Go back to top of file]

Port Assignments:

Keyword	Decimal	Description	References
-----	-----	-----	-----
	0/tcp	Reserved	
	0/udp	Reserved	
#		Jon Postel <postel@isi.edu>	
tcpmux	1/tcp	TCP Port Service Multiplexer	
tcpmux	1/udp	TCP Port Service Multiplexer	
#		Mark Lottor <MKL@nisc.sri.com>	
compressnet	2/tcp	Management Utility	
compressnet	2/udp	Management Utility	
compressnet	3/tcp	Compression Process	
compressnet	3/udp	Compression Process	
#		Bernie Volz <VOLZ@PROCESS.COM>	
#	4/tcp	Unassigned	
#	4/udp	Unassigned	
rje	5/tcp	Remote Job Entry	
rje	5/udp	Remote Job Entry	
#		Jon Postel <postel@isi.edu>	
#	6/tcp	Unassigned	
#	6/udp	Unassigned	
echo			
echo	7/tcp	Echo	
echo	7/udp	Echo	
#		Jon Postel <postel@isi.edu>	
#	8/tcp	Unassigned	
#	8/udp	Unassigned	
discard			
discard	9/tcp	Discard	
discard	9/udp	Discard	
#		Jon Postel <postel@isi.edu>	
#	10/tcp	Unassigned	
#	10/udp	Unassigned	
systat	11/tcp	Active Users	
systat	11/udp	Active Users	
#		Jon Postel <postel@isi.edu>	
#	12/tcp	Unassigned	
#	12/udp	Unassigned	
daytime			
daytime	13/tcp	Daytime	
daytime	13/udp	Daytime	
#		Jon Postel <postel@isi.edu>	
#	14/tcp	Unassigned	
#	14/udp	Unassigned	
#	15/tcp	Unassigned [was netstat]	
#	15/udp	Unassigned	
#	16/tcp	Unassigned	



```

#          16/udp  Unassigned
qotd      17/tcp  Quote of the Day
qotd      17/udp  Quote of the Day
#          Jon Postel <postel@isi.edu>
msp       18/tcp  Message Send Protocol
msp       18/udp  Message Send Protocol
#          Rina Nethaniel <---none--->

chargen   19/tcp  Character Generator
chargen   19/udp  Character Generator

ftp (data and control)
ftp-data  20/tcp  File Transfer [Default Data]
ftp-data  20/udp  File Transfer [Default Data]
ftp       21/tcp  File Transfer [Control]
ftp       21/udp  File Transfer [Control]
#          Jon Postel <postel@isi.edu>
ssh       22/tcp  SSH Remote Login Protocol
ssh       22/udp  SSH Remote Login Protocol
#          Tatu Ylonen <ylo@cs.hut.fi>
telnet    23/tcp  Telnet
telnet    23/udp  Telnet
#          Jon Postel <postel@isi.edu>
          24/tcp  any private mail system
          24/udp  any private mail system
#          Rick Adams <rick@UUNET.UU.NET>
smtp      25/tcp  Simple Mail Transfer
smtp      25/udp  Simple Mail Transfer
#          Jon Postel <postel@isi.edu>
#          26/tcp  Unassigned
#          26/udp  Unassigned
nsw-fe    27/tcp  NSW User System FE
nsw-fe    27/udp  NSW User System FE
#          Robert Thomas <BThomas@F.BBN.COM>
#          28/tcp  Unassigned
#          28/udp  Unassigned
msg-icp   29/tcp  MSG ICP
msg-icp   29/udp  MSG ICP
#          Robert Thomas <BThomas@F.BBN.COM>
#          30/tcp  Unassigned
#          30/udp  Unassigned
msg-auth  31/tcp  MSG Authentication
msg-auth  31/udp  MSG Authentication
#          Robert Thomas <BThomas@F.BBN.COM>
#          32/tcp  Unassigned
#          32/udp  Unassigned
dsp       33/tcp  Display Support Protocol
dsp       33/udp  Display Support Protocol
#          Ed Cain <cain@edn-unix.dca.mil>
#          34/tcp  Unassigned

```

```

#      34/udp  Unassigned
35/tcp  any private printer server
35/udp  any private printer server
#      Jon Postel <postel@isi.edu>
#      36/tcp  Unassigned
#      36/udp  Unassigned
time    37/tcp  Time
time    37/udp  Time
#      Jon Postel <postel@isi.edu>
rap     38/tcp  Route Access Protocol
rap     38/udp  Route Access Protocol
#      Robert Ullmann <ariel@world.std.com>
rlp     39/tcp  Resource Location Protocol
rlp     39/udp  Resource Location Protocol
#      Mike Accetta <MIKE.ACCETTA@CMU-CS-A.EDU>
#      40/tcp  Unassigned
#      40/udp  Unassigned
graphics 41/tcp  Graphics
graphics 41/udp  Graphics
nameserver 42/tcp  Host Name Server
nameserver 42/udp  Host Name Server
nickname 43/tcp  Who Is
nickname 43/udp  Who Is
mpm-flags 44/tcp  MPM FLAGS Protocol
mpm-flags 44/udp  MPM FLAGS Protocol
mpm      45/tcp  Message Processing Module [recv]
mpm      45/udp  Message Processing Module [recv]
mpm-snd  46/tcp  MPM [default send]
mpm-snd  46/udp  MPM [default send]
#      Jon Postel <postel@isi.edu>
ni-ftp   47/tcp  NI FTP
ni-ftp   47/udp  NI FTP
#      Steve Kille <S.Kille@isode.com>
auditd   48/tcp  Digital Audit Daemon
auditd   48/udp  Digital Audit Daemon
#      Larry Scott <scott@zk3.dec.com>
bbn-login 49/tcp  Login Host Protocol (TACACS)
bbn-login 49/udp  Login Host Protocol (TACACS)
#      Pieter Ditmars <pditmars@BBN.COM>
re-mail-ck 50/tcp  Remote Mail Checking Protocol
re-mail-ck 50/udp  Remote Mail Checking Protocol
#      Steve Dorner <s-dorner@UIUC.EDU>
la-maint 51/tcp  IMP Logical Address Maintenance
la-maint 51/udp  IMP Logical Address Maintenance
#      Andy Malis <malis_a@timeplex.com>
xns-time 52/tcp  XNS Time Protocol
xns-time 52/udp  XNS Time Protocol
#      Susie Armstrong <Armstrong.wbst128@XEROX>
domain   53/tcp  Domain Name Server
domain   53/udp  Domain Name Server

```

```

#           Paul Mockapetris <PVM@ISI.EDU>
xns-ch     54/tcp  XNS Clearinghouse
xns-ch     54/udp  XNS Clearinghouse
#           Susie Armstrong <Armstrong.wbst128@XEROX>
isi-gl     55/tcp  ISI Graphics Language
isi-gl     55/udp  ISI Graphics Language
xns-auth   56/tcp  XNS Authentication
xns-auth   56/udp  XNS Authentication
#           Susie Armstrong <Armstrong.wbst128@XEROX>
57/tcp     any private terminal access
57/udp     any private terminal access
#           Jon Postel <postel@isi.edu>
xns-mail   58/tcp  XNS Mail
xns-mail   58/udp  XNS Mail
#           Susie Armstrong <Armstrong.wbst128@XEROX>
59/tcp     any private file service
59/udp     any private file service
#           Jon Postel <postel@isi.edu>
60/tcp     Unassigned
60/udp     Unassigned
ni-mail    61/tcp  NI MAIL
ni-mail    61/udp  NI MAIL
#           Steve Kille <S.Kille@isode.com>
acas       62/tcp  ACA Services
acas       62/udp  ACA Services
#           E. Wald <ewald@via.enet.dec.com>
whois++    63/tcp  whois++
whois++    63/udp  whois++
#           Rickard Schoultz <schoultz@sunet.se>
covia      64/tcp  Communications Integrator (CI)
covia      64/udp  Communications Integrator (CI)
#           "Tundra" Tim Daneliuk
#           <tundraix!tundra@clout.chi.il.us>
tacacs-ds  65/tcp  TACACS-Database Service
tacacs-ds  65/udp  TACACS-Database Service
#           Kathy Huber <khuber@bbn.com>
sql*net    66/tcp  Oracle SQL*NET
sql*net    66/udp  Oracle SQL*NET
#           Jack Haverty <jhaverty@ORACLE.COM>
bootps     67/tcp  Bootstrap Protocol Server
bootps     67/udp  Bootstrap Protocol Server
bootpc     68/tcp  Bootstrap Protocol Client
bootpc     68/udp  Bootstrap Protocol Client
#           Bill Croft <Croft@SUMEX-AIM.STANFORD.EDU>
tftp       69/tcp  Trivial File Transfer
tftp       69/udp  Trivial File Transfer
#           David Clark <ddc@LCS.MIT.EDU>
gopher     70/tcp  Gopher
gopher     70/udp  Gopher
#           Mark McCahill <mpm@boombox.micro.umn.edu>
netrjs-1   71/tcp  Remote Job Service

```

```

netrjs-1    71/udp  Remote Job Service
netrjs-2    72/tcp  Remote Job Service
netrjs-2    72/udp  Remote Job Service
netrjs-3    73/tcp  Remote Job Service
netrjs-3    73/udp  Remote Job Service
netrjs-4    74/tcp  Remote Job Service
netrjs-4    74/udp  Remote Job Service
#           Bob Braden <Braden@ISI.EDU>
75/tcp     any private dial out service
75/udp     any private dial out service
#           Jon Postel <postel@isi.edu>
deos       76/tcp  Distributed External Object Store
deos       76/udp  Distributed External Object Store
#           Robert Ullmann <ariel@world.std.com>
77/tcp     any private RJE service
77/udp     any private RJE service
#           Jon Postel <postel@isi.edu>
vettcp     78/tcp  vettcp
vettcp     78/udp  vettcp
#           Christopher Leong <leong@kolmod.mlo.dec.com>
finger     79/tcp  Finger
finger     79/udp  Finger
#           David Zimmerman <dpz@RUTGERS.EDU>
http       80/tcp  World Wide Web HTTP
http       80/udp  World Wide Web HTTP
www-http   80/tcp  World Wide Web HTTP
www-http   80/udp  World Wide Web HTTP
#           Tim Berners-Lee <timbl@W3.org>
hosts2-ns  81/tcp  HOSTS2 Name Server
hosts2-ns  81/udp  HOSTS2 Name Server
#           Earl Killian <EAK@MORDOR.S1.GOV>
xfer       82/tcp  XFER Utility
xfer       82/udp  XFER Utility
#           Thomas M. Smith <tmsmith@esc.syr.ge.com>
mit-ml-dev 83/tcp  MIT ML Device
mit-ml-dev 83/udp  MIT ML Device
#           David Reed <--none-->
ctf        84/tcp  Common Trace Facility
ctf        84/udp  Common Trace Facility
#           Hugh Thomas <thomas@oils.enet.dec.com>
mit-ml-dev 85/tcp  MIT ML Device
mit-ml-dev 85/udp  MIT ML Device
#           David Reed <--none-->
mfcobol    86/tcp  Micro Focus Cobol
mfcobol    86/udp  Micro Focus Cobol
#           Simon Edwards <--none-->
87/tcp     any private terminal link
87/udp     any private terminal link
#           Jon Postel <postel@isi.edu>
kerberos   88/tcp  Kerberos
kerberos   88/udp  Kerberos

```

```

#           B. Clifford Neuman <bcn@isi.edu>
su-mit-tg   89/tcp  SU/MIT Telnet Gateway
su-mit-tg   89/udp  SU/MIT Telnet Gateway
#           Mark Crispin <MRC@PANDA.COM>
dnsix       90/tcp  DNSIX Securit Attribute Token Map
dnsix       90/udp  DNSIX Securit Attribute Token Map
#           Charles Watt <watt@sware.com>
mit-dov     91/tcp  MIT Dover Spooler
mit-dov     91/udp  MIT Dover Spooler
#           Eliot Moss <EBM@XX.LCS.MIT.EDU>
npp         92/tcp  Network Printing Protocol
npp         92/udp  Network Printing Protocol
#           Louis Mamakos <louie@sayshell.umd.edu>
dcp         93/tcp  Device Control Protocol
dcp         93/udp  Device Control Protocol
#           Daniel Tappan <Tappan@BBN.COM>
objcall     94/tcp  Tivoli Object Dispatcher
objcall     94/udp  Tivoli Object Dispatcher
#           Tom Bereiter <--none-->
supdup      95/tcp  SUPDUP
supdup      95/udp  SUPDUP
#           Mark Crispin <MRC@PANDA.COM>
dixie       96/tcp  DIXIE Protocol Specification
dixie       96/udp  DIXIE Protocol Specification
#           Tim Howes <Tim.Howes@terminator.cc.umich.edu>
swift-rvf   97/tcp  Swift Remote Virtual File Protocol
swift-rvf   97/udp  Swift Remote Virtual File Protocol
#           Maurice R. Turcotte
#           <mailrus!uflorida!rm1!dnmrt%rmatl@uunet.UU.NET>

tacnews     98/tcp  TAC News
tacnews     98/udp  TAC News
#           Jon Postel <postel@isi.edu>
metagram    99/tcp  Metagram Relay
metagram    99/udp  Metagram Relay
#           Geoff Goodfellow <Geoff@FERNWOOD.MPK.CA.U>
newacct     100/tcp [unauthorized use]
hostname    101/tcp  NIC Host Name Server
hostname    101/udp  NIC Host Name Server
#           Jon Postel <postel@isi.edu>
iso-tsap    102/tcp  ISO-TSAP Class 0
iso-tsap    102/udp  ISO-TSAP Class 0
#           Marshall Rose <mrose@dbc.mtview.ca.us>
gppitnp     103/tcp  Genesis Point-to-Point Trans Net
gppitnp     103/udp  Genesis Point-to-Point Trans Net
acr-nema    104/tcp  ACR-NEMA Digital Imag. & Comm.
300
acr-nema    104/udp  ACR-NEMA Digital Imag. & Comm.
300
#           Patrick McNamee <--none-->

```

```

csnet-ns    105/tcp  Mailbox Name Nameserver
csnet-ns    105/udp  Mailbox Name Nameserver
#           Marvin Solomon <solomon@CS.WISC.EDU>
3com-tsmux  106/tcp  3COM-TSMUX
3com-tsmux  106/udp  3COM-TSMUX
#           Jeremy Siegel <jzs@NSD.3Com.COM>
rtelnet     107/tcp  Remote Telnet Service
rtelnet     107/udp  Remote Telnet Service
#           Jon Postel <postel@isi.edu>
snagas      108/tcp  SNA Gateway Access Server
snagas      108/udp  SNA Gateway Access Server
#           Kevin Murphy <murphy@sevens.lkg.dec.com>
pop2        109/tcp  Post Office Protocol - Version 2
pop2        109/udp  Post Office Protocol - Version 2
#           Joyce K. Reynolds <jkrey@isi.edu>
pop3        110/tcp  Post Office Protocol - Version 3
pop3        110/udp  Post Office Protocol - Version 3
#           Marshall Rose <mrose@dbc.mtview.ca.us>
sunrpc      111/tcp  SUN Remote Procedure Call
sunrpc      111/udp  SUN Remote Procedure Call
#           Chuck McManis <cmcmanis@sun.com>
mcidas      112/tcp  McIDAS Data Transmission Protocol
mcidas      112/udp  McIDAS Data Transmission Protocol
#           Glenn Davis <davis@unidata.ucar.edu>
auth        113/tcp  Authentication Service
auth        113/udp  Authentication Service
#           Mike St. Johns <stjohns@arpa.mil>
audionews   114/tcp  Audio News Multicast
audionews   114/udp  Audio News Multicast
#           Martin Forssen <maf@dtek.chalmers.se>
sftp        115/tcp  Simple File Transfer Protocol
sftp        115/udp  Simple File Transfer Protocol
#           Mark Lottor <MKL@nisc.sri.com>
ansanotify  116/tcp  ANSA REX Notify
ansanotify  116/udp  ANSA REX Notify
#           Nicola J. Howarth <njh@ansa.co.uk>
uucp-path   117/tcp  UUCP Path Service
uucp-path   117/udp  UUCP Path Service
sqlserv     118/tcp  SQL Services
sqlserv     118/udp  SQL Services
#           Larry Barnes <barnes@broke.enet.dec.com>
nntp        119/tcp  Network News Transfer Protocol
nntp        119/udp  Network News Transfer Protocol
#           Phil Lapsley <phil@UCBARPA.BERKELEY.EDU>
cfdpkt      120/tcp  CFDPKT
cfdpkt      120/udp  CFDPKT
#           John Ioannidis <ji@close.cs.columbia.ed>
erpc        121/tcp  Encore Expedited Remote Pro.Call
erpc        121/udp  Encore Expedited Remote Pro.Call
#           Jack O'Neil <---none--->
smakynet    122/tcp  SMAKYNET

```

```

smakynet    122/udp  SMAKYNET
#           Mike O'Dowd <odowd@ltisun8.epfl.ch>
ntp         123/tcp  Network Time Protocol
ntp         123/udp  Network Time Protocol
#           Dave Mills <Mills@HUEY.UDEL.EDU>
ansatrader  124/tcp  ANSA REX Trader
ansatrader  124/udp  ANSA REX Trader
#           Nicola J. Howarth <njh@ansa.co.uk>
locus-map   125/tcp  Locus PC-Interface Net Map Ser
locus-map   125/udp  Locus PC-Interface Net Map Ser
#           Eric Peterson <lcc.eric@SEAS.UCLA.EDU>
unitary     126/tcp  Unisys Unitary Login
unitary     126/udp  Unisys Unitary Login
#           <feil@kronos.nisd.cam.unisys.com>
locus-con   127/tcp  Locus PC-Interface Conn Server
locus-con   127/udp  Locus PC-Interface Conn Server
#           Eric Peterson <lcc.eric@SEAS.UCLA.EDU>
gss-xlicen  128/tcp  GSS X License Verification
gss-xlicen  128/udp  GSS X License Verification
#           John Light <johnl@gssc.gss.com>
pwdgen      129/tcp  Password Generator Protocol
pwdgen      129/udp  Password Generator Protocol
#           Frank J. Wacho <WANCHO@WSMR-SIMTEL20.ARMY.MIL>
cisco-fna   130/tcp  cisco FNATIVE
cisco-fna   130/udp  cisco FNATIVE
cisco-tna   131/tcp  cisco TNATIVE
cisco-tna   131/udp  cisco TNATIVE
cisco-sys   132/tcp  cisco SYSMANT
cisco-sys   132/udp  cisco SYSMANT
statsrv     133/tcp  Statistics Service
statsrv     133/udp  Statistics Service
#           Dave Mills <Mills@HUEY.UDEL.EDU>
ingres-net  134/tcp  INGRES-NET Service
ingres-net  134/udp  INGRES-NET Service
#           Mike Berrow <---none--->
loc-srv     135/tcp  Location Service
loc-srv     135/udp  Location Service
#           Joe Pato <apollo!pato@EDDIE.MIT.EDU>
profile     136/tcp  PROFILE Naming System
profile     136/udp  PROFILE Naming System
#           Larry Peterson <llp@ARIZONA.EDU>
netbios-ns  137/tcp  NETBIOS Name Service
netbios-ns  137/udp  NETBIOS Name Service
netbios-dgm 138/tcp  NETBIOS Datagram Service
netbios-dgm 138/udp  NETBIOS Datagram Service
netbios-ssn 139/tcp  NETBIOS Session Service
netbios-ssn 139/udp  NETBIOS Session Service
#           Jon Postel <postel@isi.edu>
emfis-data  140/tcp  EMFIS Data Service
emfis-data  140/udp  EMFIS Data Service
emfis-ctrl  141/tcp  EMFIS Control Service

```

```

emfis-cntl  141/udp  EMFIS Control Service
#           Gerd Beling <GBELING@ISI.EDU>
bl-idm      142/tcp  Britton-Lee IDM
bl-idm      142/udp  Britton-Lee IDM
#           Susie Snitzer <---none--->
imap2       143/tcp  Interim Mail Access Protocol v2
imap2       143/udp  Interim Mail Access Protocol v2
#           Mark Crispin <MRC@PANDA.COM>
news        144/tcp  NewS
news        144/udp  NewS
#           James Gosling <JAG@SUN.COM>
uaac        145/tcp  UAAC Protocol
uaac        145/udp  UAAC Protocol
#           David A. Gomberg <gomberg@GATEWAY.MITRE.ORG>
iso-tp0     146/tcp  ISO-IP0
iso-tp0     146/udp  ISO-IP0
iso-ip      147/tcp  ISO-IP
iso-ip      147/udp  ISO-IP
#           Marshall Rose <mrose@dbc.mtview.ca.us>
cronus      148/tcp  CRONUS-SUPPORT
cronus      148/udp  CRONUS-SUPPORT
#           Jeffrey Buffun <jbuffun@APOLLO.COM>
aed-512     149/tcp  AED 512 Emulation Service

aed-512     149/udp  AED 512 Emulation Service

#           Albert G. Broscius <broscius@DSL.CIS.UPENN.EDU>
sql-net     150/tcp  SQL-NET
sql-net     150/udp  SQL-NET
#           Martin Picard <---none--->
hems        151/tcp  HEMS
hems        151/udp  HEMS
#           Christopher Tengi <tengi@Princeton.EDU>
bftp        152/tcp  Background File Transfer Program
bftp        152/udp  Background File Transfer Program
#           Annette DeSchon <DESCHON@ISI.EDU>
sgmp        153/tcp  SGMP
sgmp        153/udp  SGMP
#           Marty Schoffstahl <schoff@NISC.NYSER.NET>
netsc-prod  154/tcp  NETSC
netsc-prod  154/udp  NETSC
netsc-dev   155/tcp  NETSC
netsc-dev   155/udp  NETSC
#           Sergio Heker <heker@JVNCC.CSC.ORG>
sqlsrv      156/tcp  SQL Service
sqlsrv      156/udp  SQL Service
#           Craig Rogers <Rogers@ISI.EDU>
knet-cmp    157/tcp  KNET/VM Command/Message Protocol
knet-cmp    157/udp  KNET/VM Command/Message Protocol
#           Gary S. Malkin <GMALKIN@XYLOGICS.COM>
pcmail-srv  158/tcp  PCMail Server

```



```

pcmail-srv 158/udp PCMail Server
#          Mark L. Lambert <markl@PTT.LCS.MIT.EDU>
nss-routing 159/tcp NSS-Routing
nss-routing 159/udp NSS-Routing
#          Yakov Rekhter <Yakov@IBM.COM>
sgmp-traps 160/tcp SGMP-TRAPS
sgmp-traps 160/udp SGMP-TRAPS
#          Marty Schoffstahl <schoff@NISC.NYSER.NET>
snmp       161/tcp  SNMP
snmp       161/udp  SNMP
snmptrap   162/tcp  SNMPTRAP
snmptrap   162/udp  SNMPTRAP
#          Marshall Rose <mrose@dbc.mtview.ca.us>
cmip-man   163/tcp  CMIP/TCP Manager
cmip-man   163/udp  CMIP/TCP Manager
cmip-agent 164/tcp  CMIP/TCP Agent
smip-agent 164/udp  CMIP/TCP Agent
#          Amatzia Ben-Artzi <---none--->
xns-courier 165/tcp  Xerox
xns-courier 165/udp  Xerox
#          Susie Armstrong <Armstrong.wbst128@XEROX.COM>
s-net      166/tcp  Sirius Systems
s-net      166/udp  Sirius Systems
#          Brian Lloyd <---none--->
namp       167/tcp  NAMP
namp       167/udp  NAMP
#          Marty Schoffstahl <schoff@NISC.NYSER.NET>
rsvd       168/tcp  RSVD
rsvd       168/udp  RSVD
#          Neil Todd <mcvax!list.co.uk!neil@UUNET.UU.NET>
send       169/tcp  SEND
send       169/udp  SEND
#          William D. Wisner <wisner@HAYES.FAI.ALASKA.EDU>
print-srv  170/tcp  Network PostScript
print-srv  170/udp  Network PostScript
#          Brian Reid <reid@DECWRL.DEC.COM>
multiplex  171/tcp  Network Innovations Multiplex
multiplex  171/udp  Network Innovations Multiplex
cl/1       172/tcp  Network Innovations CL/1
cl/1       172/udp  Network Innovations CL/1
#          Kevin DeVault <<---none--->
xyplex-mux 173/tcp  Xyplex
xyplex-mux 173/udp  Xyplex
#          Bob Stewart <STEWART@XYPLEX.COM>
mailq      174/tcp  MAILQ

mailq      174/udp  MAILQ

#          Rayan Zachariassen <rayan@AI.TORONTO.EDU>
vmnet      175/tcp  VMNET

```

```

vmnet      175/udp  VMNET
#          Christopher Tengi <tengi@Princeton.EDU>
genrad-mux 176/tcp  GENRAD-MUX
genrad-mux 176/udp  GENRAD-MUX
#          Ron Thornton <thornton@qm7501.genrad.com>
xdmcp      177/tcp  X Display Manager Control Protocol

xdmcp      177/udp  X Display Manager Control Protocol

#          Robert W. Scheifler <RWS@XX.LCS.MIT.EDU>
nextstep   178/tcp  NextStep Window Server
NextStep   178/udp  NextStep Window Server
#          Leo Hourvitz <leo@NEXT.COM>
bgp        179/tcp  Border Gateway Protocol

bgp        179/udp  Border Gateway Protocol

#          Kirk Lougheed <LOUGHEED@MATHOM.CISCO.COM>
ris        180/tcp  Intergraph

ris        180/udp  Intergraph

#          Dave Buehmann <ingr!daveb@UUNET.UU.NET>
unify      181/tcp  Unify
unify      181/udp  Unify
#          Vinod Singh <--none-->
audit      182/tcp  Unisys Audit SITP

audit      182/udp  Unisys Audit SITP

#          Gil Greenbaum <gcole@nisd.cam.unisys.com>
ocbinder   183/tcp  OCBinder
ocbinder   183/udp  OCBinder
ocserver   184/tcp  OCServer
ocserver   184/udp  OCServer
#          Jerrilynn Okamura <--none-->
remote-kis 185/tcp  Remote-KIS
remote-kis 185/udp  Remote-KIS
kis        186/tcp  KIS Protocol
kis        186/udp  KIS Protocol
#          Ralph Droms <rdroms@NRI.RESTON.VA.US>
aci        187/tcp  Application Communication Interface
aci        187/udp  Application Communication Interface
#          Rick Carlos <rick.ticipa.csc.ti.com>
mumps     188/tcp  Plus Five's MUMPS
mumps     188/udp  Plus Five's MUMPS
#          Hokey Stenn <hokey@PLUS5.COM>
qft       189/tcp  Queued File Transport
qft       189/udp  Queued File Transport
#          Wayne Schroeder <schroeder@SDS.SDSC.EDU>
gacp      190/tcp  Gateway Access Control Protocol

```

cacp 190/udp Gateway Access Control Protocol  
# C. Philip Wood <cpw@LANL.GOV>  
prospero 191/tcp Prospero Directory Service  
prospero 191/udp Prospero Directory Service  
# B. Clifford Neuman <bcn@isi.edu>  
osu-nms 192/tcp OSU Network Monitoring System  
  
osu-nms 192/udp OSU Network Monitoring System  
  
# Doug Karl <KARL-D@OSU-20.IRCC.OHIO-STATE.EDU>  
srmp 193/tcp Spider Remote Monitoring Protocol  
srmp 193/udp Spider Remote Monitoring Protocol  
# Ted J. Socolofsky <Teds@SPIDER.CO.UK>  
irc 194/tcp Internet Relay Chat Protocol  
  
irc 194/udp Internet Relay Chat Protocol  
  
# Jarkko Oikarinen <jto@TOLSUN.OULU.FI>  
dn6-nlm-aud 195/tcp DNSIX Network Level Module Audit  
  
dn6-nlm-aud 195/udp DNSIX Network Level Module Audit  
  
dn6-smm-red 196/tcp DNSIX Session Mgt Module Audit Redir  
dn6-smm-red 196/udp DNSIX Session Mgt Module Audit Redir  
# Lawrence Lebahn <DIA3@PAXRV-NES.NAVY.MIL>  
dls 197/tcp Directory Location Service  
dls 197/udp Directory Location Service  
dls-mon 198/tcp Directory Location Service Monitor  
dls-mon 198/udp Directory Location Service Monitor  
# Scott Bellew <smb@cs.purdue.edu>  
smux 199/tcp SMUX  
smux 199/udp SMUX  
# Marshall Rose <mrose@dbc.mtview.ca.us>  
src 200/tcp IBM System Resource Controller  
src 200/udp IBM System Resource Controller  
# Gerald McBrearty <---none--->  
at-rtmp 201/tcp AppleTalk Routing Maintenance  
  
at-rtmp 201/udp AppleTalk Routing Maintenance  
  
at-nbp 202/tcp AppleTalk Name Binding  
  
at-nbp 202/udp AppleTalk Name Binding  
  
at-3 203/tcp AppleTalk Unused  
  
at-3 203/udp AppleTalk Unused  
  
at-echo 204/tcp AppleTalk Echo  
  
at-echo 204/udp AppleTalk Echo

at-5 205/tcp AppleTalk Unused  
at-5 205/udp AppleTalk Unused  
at-zis 206/tcp AppleTalk Zone Information  
at-zis 206/udp AppleTalk Zone Information  
at-7 207/tcp AppleTalk Unused  
at-7 207/udp AppleTalk Unused  
at-8 208/tcp AppleTalk Unused  
at-8 208/udp AppleTalk Unused  
# Rob Chandhok <chandhok@gnome.cs.cmu.edu>  
tam 209/tcp Trivial Authenticated Mail Protocol  
tam 209/udp Trivial Authenticated Mail Protocol  
# Dan Bernstein <djb@silvertan.berkeley.edu>  
z39.50 210/tcp ANSI Z39.50  
z39.50 210/udp ANSI Z39.50  
# Mark Needleman  
# <mhnur%uccmvs.a.bitnet@cornell.cit.cornell.edu>  
914c/g 211/tcp Texas Instruments 914C/G Terminal  
914c/g 211/udp Texas Instruments 914C/G Terminal  
# Bill Harrell <---none--->  
anet 212/tcp ATEXSSTR  
anet 212/udp ATEXSSTR  
# Jim Taylor <taylor@heart.epps.kodak.com>  
ipx 213/tcp IPX  
ipx 213/udp IPX  
# Don Provan <donp@xlnvax.novell.com>  
vmpwscs 214/tcp VM PWSCS  
vmpwscs 214/udp VM PWSCS  
# Dan Shia <dset!shia@uunet.UU.NET>  
softpc 215/tcp Insignia Solutions  
softpc 215/udp Insignia Solutions  
# Martyn Thomas <---none--->  
atls 216/tcp Access Technology License Server  
atls 216/udp Access Technology License Server  
# Larry DeLuca <henrik@EDDIE.MIT.EDU>  
dbase 217/tcp dBASE Unix  
dbase 217/udp dBASE Unix  
# Don Gibson

```

# <sequent!aero!twinsun!ashtate.A-T.COM!dong@uunet.UU.NET>

mpp      218/tcp  Netix Message Posting Protocol
mpp      218/udp  Netix Message Posting Protocol
#        Shannon Yeh <yeh@netix.com>
uarps    219/tcp  Unisys ARPs
uarps    219/udp  Unisys ARPs
#        Ashok Marwaha <---none--->
imap3    220/tcp  Interactive Mail Access Protocol v3
imap3    220/udp  Interactive Mail Access Protocol v3
#        James Rice <RICE@SUMEX-AIM.STANFORD.EDU>
fln-spx  221/tcp  Berkeley rlogind with SPX auth
fln-spx  221/udp  Berkeley rlogind with SPX auth
rsh-spx  222/tcp  Berkeley rshd with SPX auth
rsh-spx  222/udp  Berkeley rshd with SPX auth
cdc      223/tcp  Certificate Distribution Center
cdc      223/udp  Certificate Distribution Center
#        Kannan Alagappan <kannan@sejour.enet.dec.com>
#        224-241  Reserved
#        Jon Postel <postel@isi.edu>
#        242/tcp  Unassigned
#        242/udp  Unassigned
sur-meas 243/tcp  Survey Measurement
sur-meas 243/udp  Survey Measurement
#        Dave Clark <ddc@LCS.MIT.EDU>
#        244/tcp  Unassigned
#        244/udp  Unassigned
link     245/tcp  LINK
link     245/udp  LINK
dsp3270  246/tcp  Display Systems Protocol
dsp3270  246/udp  Display Systems Protocol
#        Weldon J. Showalter <Gamma@MINTAKA.DCA.MIL>
#        247-255  Reserved
#        Jon Postel <postel@isi.edu>
#        256-343  Unassigned
pdap     344/tcp  Prospero Data Access Protocol
pdap     344/udp  Prospero Data Access Protocol
#        B. Clifford Neuman <bcn@isi.edu>
pawserv  345/tcp  Perf Analysis Workbench
pawserv  345/udp  Perf Analysis Workbench
zserv    346/tcp  Zebra server
zserv    346/udp  Zebra server
fatserv  347/tcp  Fatmen Server
fatserv  347/udp  Fatmen Server
csi-sgwp 348/tcp  Cabletron Management Protocol
csi-sgwp 348/udp  Cabletron Management Protocol
#        349-370  Unassigned
clearcase 371/tcp  Clearcase
clearcase 371/udp  Clearcase
#        Dave LeBlang <leklang@atria.com>
ulistserv 372/tcp  Unix Listserv

```

```

ulisterv  372/udp  Unix Listserv
#          Anastasios Kotsikonas <tasos@cs.bu.edu>
legent-1  373/tcp  Legent Corporation
legent-1  373/udp  Legent Corporation
legent-2  374/tcp  Legent Corporation
legent-2  374/udp  Legent Corporation
#          Keith Boyce <---none--->
hassle    375/tcp  Hassle
hassle    375/udp  Hassle
#          Reinhard Doelz <doelz@comp.bioz.unibas.ch>
nip       376/tcp  Amiga Envoy Network Inquiry Proto

nip       376/udp  Amiga Envoy Network Inquiry Proto
#          Heinz Wrobel <heinz@iam.com>
#          Dale L. Larson <dale@iam.com>
tnETOS    377/tcp  NEC Corporation
tnETOS    377/udp  NEC Corporation
dsETOS    378/tcp  NEC Corporation
dsETOS    378/udp  NEC Corporation
#          Tomoo Fujita <tf@arc.bs1.fc.nec.co.jp>
is99c     379/tcp  TIA/EIA/IS-99 modem client
is99c     379/udp  TIA/EIA/IS-99 modem client
is99s     380/tcp  TIA/EIA/IS-99 modem server
is99s     380/udp  TIA/EIA/IS-99 modem server
#          Frank Quick <fquick@qualcomm.com>
hp-collector 381/tcp  hp performance data collector
hp-collector 381/udp  hp performance data collector
hp-managed-node 382/tcp  hp performance data managed node
hp-managed-node 382/udp  hp performance data managed node
hp-alarm-mgr 383/tcp  hp performance data alarm manager
hp-alarm-mgr 383/udp  hp performance data alarm manager
#          Frank Blakely <frankb@hpptc16.rose.hp.com>
arns      384/tcp  A Remote Network Server System
arns      384/udp  A Remote Network Server System
#          David Hornsby <djh@munnari.OZ.AU>
ibm-app   385/tcp  IBM Application
ibm-app   385/tcp  IBM Application
#          Lisa Tomita <---none--->
asa       386/tcp  ASA Message Router Object Def.
asa       386/udp  ASA Message Router Object Def.
#          Steve Laitinen <laitinen@brutus.aa.ab.com>
aurp      387/tcp  Appletalk Update-Based Routing Pro.
aurp      387/udp  Appletalk Update-Based Routing Pro.
#          Chris Ranch <cranch@novell.com>
unidata-ldm 388/tcp  Unidata LDM Version 4
unidata-ldm 388/udp  Unidata LDM Version 4
#          Glenn Davis <davis@unidata.ucar.edu>
ldap      389/tcp  Lightweight Directory Access Protocol
ldap      389/udp  Lightweight Directory Access Protocol
#          Tim Howes <Tim.Howes@terminator.cc.umich.edu>
uis       390/tcp  UIS

```

```

uis      390/udp  UIS
#        Ed Barron <---none--->
synotics-relay 391/tcp  SynOptics SNMP Relay Port
synotics-relay 391/udp  SynOptics SNMP Relay Port
synotics-broker 392/tcp  SynOptics Port Broker Port
synotics-broker 392/udp  SynOptics Port Broker Port
#        Illan Raab <iraab@synoptics.com>
dis      393/tcp  Data Interpretation System
dis      393/udp  Data Interpretation System
#        Paul Stevens <pstevens@chinacat.Metaphor.COM>
embl-ndt 394/tcp  EMBL Nucleic Data Transfer
embl-ndt 394/udp  EMBL Nucleic Data Transfer
#        Peter Gad <peter@bmc.uu.se>
netcp    395/tcp  NETscout Control Protocol
netcp    395/udp  NETscout Control Protocol
#        Anil Singhal <---none--->
netware-ip 396/tcp  Novell Netware over IP
netware-ip 396/udp  Novell Netware over IP
mptn     397/tcp  Multi Protocol Trans. Net.
mptn     397/udp  Multi Protocol Trans. Net.
#        Soumitra Sarkar <sarkar@vnet.ibm.com>
kryptolan 398/tcp  Kryptolan
kryptolan 398/udp  Kryptolan
#        Peter de Laval <pdl@sectra.se>
iso-tsap-c2 399/tcp  ISO-TSAP Class 2
iso-tsap-c2 399/udp  ISO-TSAP Class 2
#        Yanivk Pouffary <pouffary@yaec.enet.dec.com>
work-sol 400/tcp  Workstation Solutions
work-sol 400/udp  Workstation Solutions
#        Jim Ward <jimw@worksta.com>
ups       401/tcp  Uninterruptible Power Supply
ups       401/udp  Uninterruptible Power Supply
#        Guenther Seybold <gs@hrz.th-darmstadt.de>
genie     402/tcp  Genie Protocol
genie     402/udp  Genie Protocol
#        Mark Hankin <---none--->
decap     403/tcp  decap
decap     403/udp  decap
nced      404/tcp  nced
nced      404/udp  nced
nclld     405/tcp  nclld
nclld     405/udp  nclld
#        Richard Jones <---none--->
imsp      406/tcp  Interactive Mail Support Protocol
imsp      406/udp  Interactive Mail Support Protocol
#        John Myers <jgm+@cmu.edu>
timbuktu 407/tcp  Timbuktu
timbuktu 407/udp  Timbuktu
#        Marc Epard <marc@waygate.farallon.com>
prm-sm    408/tcp  Prospero Resource Manager Sys. Man.
prm-sm    408/udp  Prospero Resource Manager Sys. Man.

```

```

prm-nm      409/tcp  Prospero Resource Manager Node Man.
prm-nm      409/udp  Prospero Resource Manager Node Man.
#           B. Clifford Neuman <bcn@isi.edu>
decladebug  410/tcp  DECLadebug Remote Debug Protocol
decladebug  410/udp  DECLadebug Remote Debug Protocol
#           Anthony Berent <berent@rdgeng.enet.dec.com>
rmt         411/tcp  Remote MT Protocol
rmt         411/udp  Remote MT Protocol
#           Peter Eriksson <pen@lysator.liu.se>
synoptics-trap 412/tcp  Trap Convention Port
synoptics-trap 412/udp  Trap Convention Port
#           Illan Raab <iraab@synoptics.com>
smsp        413/tcp  SMSP
smsp        413/udp  SMSP
infoseek    414/tcp  InfoSeek
infoseek    414/udp  InfoSeek
#           Steve Kirsch <stk@frame.com>
bnet        415/tcp  BNet
bnet        415/udp  BNet
#           Jim Mertz <JMertz+RV09@rvdc.unisys.com>
silverplatter 416/tcp  Silverplatter
silverplatter 416/udp  Silverplatter
#           Peter Ciuffetti <petec@silverplatter.com>
onmux       417/tcp  Onmux
onmux       417/udp  Onmux
#           Stephen Hanna <hanna@world.std.com>
hyper-g     418/tcp  Hyper-G
hyper-g     418/udp  Hyper-G
#           Frank Kappe <fkappe@iicm.tu-graz.ac.at>
ariel1      419/tcp  Ariel
ariel1      419/udp  Ariel
#           Jonathan Lavigne <BL.JPL@RLG.Stanford.EDU>
smpte       420/tcp  SMPTE
smpte       420/udp  SMPTE
#           Si Becker <71362.22@CompuServe.COM>
ariel2      421/tcp  Ariel
ariel2      421/udp  Ariel
ariel3      422/tcp  Ariel
ariel3      422/udp  Ariel
#           Jonathan Lavigne <BL.JPL@RLG.Stanford.EDU>
opc-job-start 423/tcp  IBM Operations Planning and Control
Start
opc-job-start 423/udp  IBM Operations Planning and Control
Start
opc-job-track 424/tcp  IBM Operations Planning and Control
Track
opc-job-track 424/udp  IBM Operations Planning and Control
Track
#           Conny Larsson <cocke@VNET.IBM.COM>
icad-el     425/tcp  ICAD
icad-el     425/udp  ICAD

```



```

#      Larry Stone <lcs@icad.com>
smartsdp  426/tcp  smartsdp
smartsdp  426/udp  smartsdp
#      Alexander Dupuy <dupuy@smarts.com>
svrloc    427/tcp  Server Location
svrloc    427/udp  Server Location
#      <veizades@ftp.com>
ocs_cmu   428/tcp  OCS_CMU
ocs_cmu   428/udp  OCS_CMU
ocs_amu   429/tcp  OCS_AMU
ocs_amu   429/udp  OCS_AMU
#      Florence Wyman <wyman@peabody.plk.af.mil>
utmpsd    430/tcp  UTMPSD
utmpsd    430/udp  UTMPSD
utmpcd    431/tcp  UTMPCD
utmpcd    431/udp  UTMPCD
iasd      432/tcp  IASD
iasd      432/udp  IASD
#      Nir Baroz <nbaroz@encore.com>
nnsdp     433/tcp  NNSP
nnsdp     433/udp  NNSP
#      Rob Robertson <rob@gangrene.berkeley.edu>
mobileip-agent 434/tcp  MobileIP-Agent
mobileip-agent 434/udp  MobileIP-Agent
mobilip-mn 435/tcp  MobilIP-MN
mobilip-mn 435/udp  MobilIP-MN
#      Kannan Alagappan <kannan@sejour.lkg.dec.com>
dna-cml   436/tcp  DNA-CML
dna-cml   436/udp  DNA-CML
#      Dan Flowers <flowers@smaug.lkg.dec.com>
comscm    437/tcp  comscm
comscm    437/udp  comscm
#      Jim Teague <teague@zso.dec.com>
dsfgw     438/tcp  dsfgw
dsfgw     438/udp  dsfgw
#      Andy McKeen <mckeen@osf.org>
dasp      439/tcp  dasp  Thomas Obermair
dasp      439/udp  dasp  tommy@inlab.m.eunet.de
#      Thomas Obermair <tommy@inlab.m.eunet.de>
sgcp      440/tcp  sgcp
sgcp      440/udp  sgcp
#      Marshall Rose <mrose@dbc.mtview.ca.us>
decvms-sysmgt 441/tcp  decvms-sysmgt
decvms-sysmgt 441/udp  decvms-sysmgt
#      Lee Barton <barton@star.enet.dec.com>
cvc_hostd 442/tcp  cvc_hostd
cvc_hostd 442/udp  cvc_hostd
#      Bill Davidson <billd@equalizer.cray.com>
https     443/tcp  https MCom
https     443/udp  https MCom
#      Kipp E.B. Hickman <kipp@mcom.com>

```

```

snpp      444/tcp  Simple Network Paging Protocol
snpp      444/udp  Simple Network Paging Protocol
#         [RFC1568]
microsoft-ds 445/tcp  Microsoft-DS
microsoft-ds 445/udp  Microsoft-DS
#         Arnold Miller <arnoldm@microsoft.com>
ddm-rdb    446/tcp  DDM-RDB
ddm-rdb    446/udp  DDM-RDB
ddm-dfm    447/tcp  DDM-RFM
ddm-dfm    447/udp  DDM-RFM
ddm-byte   448/tcp  DDM-BYTE
ddm-byte   448/udp  DDM-BYTE
#         Jan David Fisher <jdfisher@VNET.IBM.COM>
as-servermap 449/tcp  AS Server Mapper
as-servermap 449/udp  AS Server Mapper
#         Barbara Foss <BGFOSS@rchvmv.vnet.ibm.com>
tserver    450/tcp  TServer
tserver    450/udp  TServer
#         Harvey S. Schultz <hss@mtgzfs3.mt.att.com>
sfs-smp-net 451/tcp  Cray Network Semaphore server
sfs-smp-net 451/udp  Cray Network Semaphore server
sfs-config 452/tcp  Cray SFS config server
sfs-config 452/udp  Cray SFS config server
#         Walter Poxon <wdp@ironwood.cray.com>
creativeserver 453/tcp  CreativeServer
creativeserver 453/udp  CreativeServer
contentserver 454/tcp  ContentServer
contentserver 454/udp  ContentServer
creativepartnr 455/tcp  CreativePartnr
creativepartnr 455/udp  CreativePartnr
#         Jesus Ortiz <jesus_ortiz@emotion.com>
macon-tcp  456/tcp  macon-tcp
macon-udp  456/udp  macon-udp
#         Yoshinobu Inoue
#         <shin@hodaka.mfd.cs.fujitsu.co.jp>
scohelp    457/tcp  scohelp
scohelp    457/udp  scohelp
#         Faith Zack <faithz@sco.com>
appleqtcp  458/tcp  apple quick time
appleqtcp  458/udp  apple quick time
#         Murali Ranganathan <murali_ranganathan@quickmail.apple.com>
ampr-rcmd  459/tcp  ampr-rcmd
ampr-rcmd  459/udp  ampr-rcmd
#         Rob Janssen <rob@sys3.pe1chl.ampr.org>
skronk     460/tcp  skronk
skronk     460/udp  skronk
#         Henry Strickland <strick@yak.net>
datasurfsrv 461/tcp  DataSurfSrv
datasurfsrv 461/udp  DataSurfSrv
datasurfsrvsec 462/tcp  DataSurfSrvSec
datasurfsrvsec 462/udp  DataSurfSrvSec

```

```

#           Larry Barnes <Larryb@larryb.MV.COM>

alpes      463/tcp  alpes
alpes      463/udp  alpes
#           Alain Durand <Alain.Durand@imag.fr>
kpasswd    464/tcp  kpasswd
kpasswd    464/udp  kpasswd
#           Theodore Ts'o <tytso@MIT.EDU>
ssmtp      465/tcp  ssmtp
ssmtp      465/udp  ssmtp
#           John Hemming <JohnHemming@Mkn.co.uk>
digital-vrc 466/tcp  digital-vrc
digital-vrc 466/udp  digital-vrc
#           Dave Forster <forster@marvin.enet.dec.com>
mylex-mapd 467/tcp  mylex-mapd
mylex-mapd 467/udp  mylex-mapd
#           Gary Lewis <GaryL@hq.mylex.com>
photuris   468/tcp  proturis
photuris   468/udp  proturis
#           Bill Simpson <Bill.Simpson@um.cc.umich.edu>
rcp        469/tcp  Radio Control Protocol
rcp        469/udp  Radio Control Protocol
#           Jim Jennings +1-708-538-7241
scx-proxy  470/tcp  scx-proxy
scx-proxy  470/udp  scx-proxy
#           Walter Poxon <wdp@ironwood-fddi.cray.com>

mondex     471/tcp  Mondex
mondex     471/udp  Mondex
#           Bill Reding <redingb@nwdt.natwest.co.uk>
ljk-login  472/tcp  ljk-login
ljk-login  472/udp  ljk-login
#           LJK Software, Cambridge, Massachusetts
#           <support@ljk.com>
hybrid-pop 473/tcp  hybrid-pop
hybrid-pop 473/udp  hybrid-pop
#           Rami Rubin <rami@hybrid.com>
tn-tl-w1   474/tcp  tn-tl-w1
tn-tl-w2   474/udp  tn-tl-w2
#           Ed Kress <eskress@thinknet.com>
tcpnethasprv 475/tcp  tcpnethasprv
tcpnethasprv 475/tcp  tcpnethasprv
#           Charlie Hava <charlie@aladdin.co.il>
#           476-511 Unassigned
exec       512/tcp  remote process execution;
#           authentication performed using
#           passwords and UNIX loppin names
biff       512/udp  used by mail system to notify users
#           of new mail received; currently
#           receives messages only from
#           processes on the same machine

```

```

login      513/tcp  remote login a la telnet;
#          automatic authentication performed
#          based on privileged port numbers
#          and distributed data bases which
#          identify "authentication domains"
who        513/udp  maintains data bases showing who's
#          logged in to machines on a local
#          net and the load average of the
#          machine
cmd        514/tcp  like exec, but automatic
#          authentication is performed as for
#          login server
syslog     514/udp
printer    515/tcp  spooler
printer    515/udp  spooler
#          516/tcp  Unassigned
#          516/udp  Unassigned
talk       517/tcp  like tenex link, but across
#          machine - unfortunately, doesn't
#          use link protocol (this is actually
#          just a rendezvous port from which a
#          tcp connection is established)
talk       517/udp  like tenex link, but across
#          machine - unfortunately, doesn't
#          use link protocol (this is actually
#          just a rendezvous port from which a
#          tcp connection is established)
ntalk      518/tcp
ntalk      518/udp
utime      519/tcp  unixtime
utime      519/udp  unixtime
efs        520/tcp  extended file name server
router     520/udp  local routing process (on site);
#          uses variant of Xerox NS routing
#          information protocol
#          521-524  Unassigned
timed      525/tcp  timeserver
timed      525/udp  timeserver
tempo      526/tcp  newdate
tempo      526/udp  newdate
#          527-529  Unassigned
courier    530/tcp  rpc
courier    530/udp  rpc
conference 531/tcp  chat
conference 531/udp  chat
netnews    532/tcp  readnews
netnews    532/udp  readnews
netwall    533/tcp  for emergency broadcasts
netwall    533/udp  for emergency broadcasts
#          534-538  Unassigned
apertus-ldp 539/tcp  Apertus Technologies Load Determination

```

```

apertus-ldp 539/udp Apertus Technologies Load Determination
uucp 540/tcp uucpd
uucp 540/udp uucpd
uucp-rlogin 541/tcp uucp-rlogin
uucp-rlogin 541/udp uucp-rlogin
# Stuart Lynne <sl@wimsey.com>
# 542/tcp Unassigned
# 542/udp Unassigned
klogin 543/tcp
klogin 543/udp
kshell 544/tcp krcmd
kshell 544/udp krcmd
appleqtcsrvr 545/tcp appleqtcsrvr
appleqtcsrvr 545/udp appleqtcsrvr
# Murali Ranganathan <Murali_Ranganathan@quickmail.apple.com>
dhcp-client 546/tcp DHCP Client
dhcp-client 546/udp DHCP Client
dhcp-server 547/tcp DHCP Server
dhcp-server 547/udp DHCP Server
# Jim Bound <bound@zk3.dec.com>
# 548/tcp Unassigned
# 548/udp Unassigned
# 549/tcp Unassigned
# 549/udp Unassigned
new-rwho 550/tcp new-who
new-rwho 550/udp new-who
cybercash 551/tcp cybercash
cybercash 551/udp cybercash
# Donald E. Eastlake 3rd <dee@cybercash.com>
deviceshare 552/tcp deviceshare
deviceshare 552/udp deviceshare
# Brian Schenkenberger <brians@advsyscon.com>
pirp 553/tcp pirp
pirp 553/udp pirp
# D. J. Bernstein <djb@silvertan.berkeley.edu>
# 554/tcp Unassigned
# 554/udp Unassigned
dsf 555/tcp
dsf 555/udp
remotefs 556/tcp rfs server
remotefs 556/udp rfs server
openvms-sysipc 557/tcp openvms-sysipc
openvms-sysipc 557/udp openvms-sysipc
# Alan Potter <potter@movies.enet.dec.com>
sdnskmp 558/tcp SDNSKMP
sdnskmp 558/udp SDNSKMP
teedtap 559/tcp TEEDTAP
teedtap 559/udp TEEDTAP
# Mort Hoffman <hoffman@mail.ndhm.gtegs.com>
rmonitor 560/tcp rmonitord
rmonitor 560/udp rmonitord

```

```

monitor    561/tcp
monitor    561/udp
chshell    562/tcp  chcmd
chshell    562/udp  chcmd
snews      563/tcp  snews
snews      563/udp  snews
#          Kipp E.B. Hickman <kipp@netscape.com>
9pfs       564/tcp  plan 9 file service
9pfs       564/udp  plan 9 file service
whoami     565/tcp  whoami
whoami     565/udp  whoami
streetalk  566/tcp  streetalk
streetalk  566/udp  streetalk
banyan-rpc 567/tcp  banyan-rpc
banyan-rpc 567/udp  banyan-rpc
#          Tom Lemaire <toml@banyan.com>
ms-shuttle 568/tcp  microsoft shuttle
ms-shuttle 568/udp  microsoft shuttle
#          Rudolph Balaz <rudolphb@microsoft.com>
ms-rome    569/tcp  microsoft rome
ms-rome    569/udp  microsoft rome
#          Rudolph Balaz <rudolphb@microsoft.com>
meter      570/tcp  demon
meter      570/udp  demon
meter      571/tcp  udemon
meter      571/udp  udemon
sonar      572/tcp  sonar
sonar      572/udp  sonar
#          Keith Moore <moore@cs.utk.edu>
banyan-vip 573/tcp  banyan-vip
banyan-vip 573/udp  banyan-vip
#          Denis Leclerc <DLeclerc@banyan.com>
#          574-599 Unassigned
ipcserver  600/tcp  Sun IPC server
ipcserver  600/udp  Sun IPC server
nqs        607/tcp  nqs
nqs        607/udp  nqs
urm        606/tcp  Cray Unified Resource Manager
urm        606/udp  Cray Unified Resource Manager
#          Bill Schiefelbein <schief@aspen.cray.com>
sift-uft   608/tcp  Sender-Initiated/Unsolicited File Transfer

sift-uft   608/udp  Sender-Initiated/Unsolicited File Transfer
#          Rick Troth <troth@rice.edu>
npmp-trap  609/tcp  npmp-trap
npmp-trap  609/udp  npmp-trap
npmp-local 610/tcp  npmp-local
npmp-local 610/udp  npmp-local
npmp-gui   611/tcp  npmp-gui
npmp-gui   611/udp  npmp-gui
#          John Barnes <jbarnes@crl.com>

```

```

ginad      634/tcp  ginad
ginad      634/udp  ginad
#          Mark Crother <mark@eis.calstate.edu>
mdqs      666/tcp
mdqs      666/udp
doom      666/tcp  doom Id Software
doom      666/udp  doom Id Software
#          <ddt@idcube.idsoftware.com>
elcsd     704/tcp  errlog copy/server daemon
elcsd     704/udp  errlog copy/server daemon
entrustmanager 709/tcp  EntrustManager
entrustmanager 709/udp  EntrustManager
#          Peter Whittaker <pww@bnr.ca>
netviewdm1 729/tcp  IBM NetView DM/6000 Server/Client
netviewdm1 729/udp  IBM NetView DM/6000 Server/Client
netviewdm2 730/tcp  IBM NetView DM/6000 send/tcp
netviewdm2 730/udp  IBM NetView DM/6000 send/tcp
netviewdm3 731/tcp  IBM NetView DM/6000 receive/tcp
netviewdm3 731/udp  IBM NetView DM/6000 receive/tcp
#          Philippe Binet (phbinet@vnet.IBM.COM)
netgw     741/tcp  netGW
netgw     741/udp  netGW
netrcs   742/tcp  Network based Rev. Cont. Sys.
netrcs   742/udp  Network based Rev. Cont. Sys.
#          Gordon C. Galligher <gorpong@ping.chi.il.us>
flexlm    744/tcp  Flexible License Manager
flexlm    744/udp  Flexible License Manager
#          Matt Christiano
#          <globes@matt@oliveb.atc.olivetti.com>

fujitsu-dev 747/tcp  Fujitsu Device Control
fujitsu-dev 747/udp  Fujitsu Device Control
ris-cm     748/tcp  Russell Info Sci Calendar Manager
ris-cm     748/udp  Russell Info Sci Calendar Manager
kerberos-adm 749/tcp  kerberos administration
kerberos-adm 749/udp  kerberos administration
rfile     750/tcp
loadav    750/udp
pump      751/tcp
pump      751/udp
qrh       752/tcp
qrh       752/udp
rrh       753/tcp
rrh       753/udp
tell      754/tcp  send
tell      754/udp  send
nlogin    758/tcp
nlogin    758/udp
con       759/tcp
con       759/udp
ns        760/tcp

```

```

ns      760/udp
rxex    761/tcp
rxex    761/udp
quotad  762/tcp
quotad  762/udp
cycleserv 763/tcp
cycleserv 763/udp
omserv  764/tcp
omserv  764/udp
webster 765/tcp
webster 765/udp
phonebook 767/tcp phone
phonebook 767/udp phone
vid     769/tcp
vid     769/udp
cadlock 770/tcp
cadlock 770/udp
rtip    771/tcp
rtip    771/udp
cycleserv2 772/tcp
cycleserv2 772/udp
submit  773/tcp
notify  773/udp
rpasswd 774/tcp
acmaint_dbd 774/udp
entomb  775/tcp
acmaint_transd 775/udp
wpages 776/tcp
wpages 776/udp
wpgs 780/tcp
wpgs 780/udp
concert 786/tcp Concert
concert 786/udp Concert
#       Josyula R. Rao <jrrao@watson.ibm.com>
mdbs_daemon 800/tcp
mdbs_daemon 800/udp
device 801/tcp
device 801/udp
accessbuilder 888/tcp AccessBuilder
accessbuilder 888/udp AccessBuilder
#       Steve Sweeney <Steven_Sweeney@3mail.3com.com>
vsinet 996/tcp vsinet
vsinet 996/udp vsinet
#       Rob Juergens <robj@vsi.com>
maitrd 997/tcp
maitrd 997/udp
busboy 998/tcp
puparp 998/udp
garcon 999/tcp
applix 999/udp Applix ac
puprouter 999/tcp

```



```

puprouter 999/udp
cadlock 1000/tcp
ock 1000/udp
        1023/tcp   Reserved
        1024/udp   Reserved
#          IANA <iana@isi.edu>

```

## REGISTERED PORT NUMBERS

The Registered Ports are not controlled by the IANA and on most systems can be used by ordinary user processes or programs executed by ordinary users. Ports are used in the TCP [RFC793] to name the ends of logical connections which carry long term conversations. For the purpose of providing services to unknown callers, a service contact port is defined. This list specifies the port used by the server process as its contact port. While the IANA can not control uses of these ports it does register or list uses of these ports as a convenience to the community. To the extent possible, these same port assignments are used with the UDP [RFC768].

The Registered Ports are in the range 1024-65535.

[Go back to top of file]

Port Assignments:

Keyword	Decimal	Description	References
	1024/tcp	Reserved	
	1024/udp	Reserved	
#		IANA <iana@isi.edu>	
blackjack	1025/tcp	network blackjack	
blackjack	1025/udp	network blackjack	
iad1	1030/tcp	BBN IAD	
iad1	1030/udp	BBN IAD	
iad2	1031/tcp	BBN IAD	
iad2	1031/udp	BBN IAD	
iad3	1032/tcp	BBN IAD	
iad3	1032/udp	BBN IAD	
#		Andy Malis <malis_a@timeplex.com>	
nim	1058/tcp	nim	
nim	1058/udp	nim	
nimreg	1059/tcp	nimreg	
nimreg	1059/udp	nimreg	
#		Robert Gordon <rbg@austin.ibm.com>	
instl_boots	1067/tcp	Installation Bootstrap Proto. Serv.	
instl_boots	1067/udp	Installation Bootstrap Proto. Serv.	
instl_bootc	1068/tcp	Installation Bootstrap Proto. Cli.	
instl_bootc	1068/udp	Installation Bootstrap Proto. Cli.	

```

#           David Arko <<darko@hpfern.fc.hp.com>
socks      1080/tcp  Socks
socks      1080/udp  Socks
#           Ying-Da Lee <ylee@syl.dl.nec.com>
ansoft-lm-1 1083/tcp  Anasoft License Manager
ansoft-lm-1 1083/udp  Anasoft License Manager
ansoft-lm-2 1084/tcp  Anasoft License Manager
ansoft-lm-2 1084/udp  Anasoft License Manager
nfsd-status 1110/tcp  Cluster status info
nfsd-keepalive 1110/udp  Client status info
#           Edgar Circenis <ec@hpfcj.fc.hp.com>
nfa        1155/tcp  Network File Access

nfa        1155/udp  Network File Access

#           James Powell <james@mailhost.unidata.com>
lupa       1212/tcp  lupa
lupa       1212/udp  lupa
#           Barney Wolff <barney@databus.com>
nerv 1222/tcp  SNI R&D network
nerv 1222/udp  SNI R&D network
#           Martin Freiss <freiss.pad@sni.de>
hermes 1248/tcp
hermes 1248/udp
alta-ana-lm 1346/tcp  Alta Analytics License Manager
alta-ana-lm 1346/udp  Alta Analytics License Manager
bbn-mmc 1347/tcp  multi media conferencing
bbn-mmc 1347/udp  multi media conferencing
bbn-mmx 1348/tcp  multi media conferencing
bbn-mmx 1348/udp  multi media conferencing
sbook     1349/tcp  Registration Network Protocol

sbook     1349/udp  Registration Network Protocol

editbench 1350/tcp  Registration Network Protocol

editbench 1350/udp  Registration Network Protocol

#           Simson L. Garfinkel <simsong@next.cambridge.ma.us>
equationbuilder 1351/tcp  Digital Tool Works (MIT)

equationbuilder 1351/udp  Digital Tool Works (MIT)

#           Terrence J. Talbot <lexcube!tjt@bu.edu>
lotusnote 1352/tcp  Lotus Note

lotusnote 1352/udp  Lotus Note

#           Greg Pflaum <iris.com!Greg_Pflaum@uunet.uu.net>
relief    1353/tcp  Relief Consulting

```

```

relief      1353/udp  Relief Consulting

#          John Feiler <relief!jjfeiler@uu2.psi.com>
rightbrain 1354/tcp  RightBrain Software
rightbrain 1354/udp  RightBrain Software
#          Glenn Reid <glann@rightbrain.com>
intuitive edge 1355/tcp  Intuitive Edge
intuitive edge 1355/udp  Intuitive Edge
#          Montgomery Zukowski
#          <monty@nextnorth.acs.ohio-state.edu>

cuillamartin 1356/tcp  CuillaMartin Company
cuillamartin 1356/udp  CuillaMartin Company
pegboard    1357/tcp  Electronic PegBoard
pegboard    1357/udp  Electronic PegBoard
#          Chris Cuilla
#          <balr!vpnet!cuilla!chris@clout.chi.il.us>

connlcli    1358/tcp  CONNLCLI

connlcli    1358/udp  CONNLCLI

ftsrv       1359/tcp  FTSRV

ftsrv       1359/udp  FTSRV

#          Ines Homem de Melo <sidinf@brfapesp.bitnet>
mimer       1360/tcp  MIMER

mimer       1360/udp  MIMER

#          Per Schroeder <Per.Schroeder@mimer.se>
linx        1361/tcp  LinX
linx        1361/udp  LinX
#          Steffen Schilke <---none--->
timeflies   1362/tcp  TimeFlies

timeflies   1362/udp  TimeFlies

#          Doug Kent <mouthers@slugg@nwnexus.wa.com>
ndm-requester 1363/tcp  Network DataMover Requester
ndm-requester 1363/udp  Network DataMover Requester
ndm-server    1364/tcp  Network DataMover Server
ndm-server    1364/udp  Network DataMover Server
#          Toshio Watanabe
#          <watanabe@godzilla.rsc.spdd.ricoh.co.jp>

adapt-sna    1365/tcp  Network Software Associates
adapt-sna    1365/udp  Network Software Associates
#          Jeffery Chiao <714-768-401>
netware-csp  1366/tcp  Novell NetWare Comm Service Platform

```

```

netware-csp  1366/udp  Novell NetWare Comm Service Platform
#            Laurie Lindsey <llindsey@novell.com>
dcs          1367/tcp  DCS
dcs          1367/udp  DCS
#            Stefan Siebert <ssiebert@dcs.de>
screencast   1368/tcp  ScreenCast

screencast   1368/udp  ScreenCast

#            Bill Tschumy <other!bill@uunet.UU.NET>
gv-us        1369/tcp  GlobalView to Unix Shell

gv-us        1369/udp  GlobalView to Unix Shell

us-gv        1370/tcp  Unix Shell to GlobalView

us-gv        1370/udp  Unix Shell to GlobalView

#            Makoto Mita <mita@ssdev.ksp.fujixerox.co.jp>
fc-cli       1371/tcp  Fujitsu Config Protocol

fc-cli       1371/udp  Fujitsu Config Protocol

fc-ser       1372/tcp  Fujitsu Config Protocol

fc-ser       1372/udp  Fujitsu Config Protocol

#            Ryuichi Horie <horie@spad.sysrap.cs.fujitsu.co.jp>
chromagrafx  1373/tcp  Chromagrafx

chromagrafx  1373/udp  Chromagrafx

#            Mike Barthelemy <msb@chromagrafx.com>
molly        1374/tcp  EPI Software Systems
molly        1374/udp  EPI Software Systems
#            Jim Vlcek <vlcek@epimbe.com>
bytex        1375/tcp  Bytex
bytex        1375/udp  Bytex
#            Mary Ann Burt <bytex!ws054!maryann@uunet.UU.NET>
ibm-pps      1376/tcp  IBM Person to Person Software
ibm-pps      1376/udp  IBM Person to Person Software
#            Simon Phipps <sphipps@vnet.ibm.com>
cichlid      1377/tcp  Cichlid License Manager
cichlid      1377/udp  Cichlid License Manager
#            Andy Burgess <aab@cichlid.com>
elan         1378/tcp  Elan License Manager
elan         1378/udp  Elan License Manager
#            Ken Greer <kg@elan.com>
dbreporter   1379/tcp  Integrity Solutions

dbreporter   1379/udp  Integrity Solutions

```

```

#           Tim Dawson <tdawson%mspboss@uunet.UU.NET>
telesis-licman 1380/tcp Telesis Network License Manager

telesis-licman 1380/udp Telesis Network License Manager

#           Karl Schendel, Jr. <wiz@telesis.com>
apple-licman 1381/tcp Apple Network License Manager
apple-licman 1381/udp Apple Network License Manager
#           Earl Wallace <earlw@apple.com>
udt_os      1382/tcp
udt_os      1382/udp
gwaha      1383/tcp GW Hannaway Network License Manager
gwaha      1383/udp GW Hannaway Network License Manager
#           J. Gabriel Foster <fop@gwaha.com>
os-licman   1384/tcp Objective Solutions License Manager

os-licman   1384/udp Objective Solutions License Manager

#           Donald Cornwell <don.cornwell@objective.com>
atex_elmd   1385/tcp Atex Publishing License Manager
atex_elmd   1385/udp Atex Publishing License Manager
#           Brett Sorenson <bcs@atex.com>
checksum    1386/tcp CheckSum License Manager

checksum    1386/udp CheckSum License Manager

#           Andreas Glocker <glocker@sirius.com>
cads-lm     1387/tcp Computer Aided Design Software Inc
LM
cads-lm     1387/udp Computer Aided Design Software Inc
LM
#           Sulistio Muljadi
objective-dbc 1388/tcp Objective Solutions DataBase Cache
objective-dbc 1388/udp Objective Solutions DataBase Cache
#           Donald Cornwell
iclpv-dm    1389/tcp Document Manager

iclpv-dm    1389/udp Document Manager

iclpv-sc    1390/tcp Storage Controller

iclpv-sc    1390/udp Storage Controller

iclpv-sas   1391/tcp Storage Access Server

iclpv-sas   1391/udp Storage Access Server

iclpv-pm    1392/tcp Print Manager

iclpv-pm    1392/udp Print Manager

```

iclqv-nls 1393/tcp Network Log Server  
 iclqv-nls 1393/udp Network Log Server  
 iclqv-nlc 1394/tcp Network Log Client  
 iclqv-nlc 1394/udp Network Log Client  
 iclqv-wsm 1395/tcp PC Workstation Manager software  
 iclqv-wsm 1395/udp PC Workstation Manager software  
 # A.P. Hobson <A.P.Hobson@bra0112.wins.icl.co.uk>  
 dvl-activemail 1396/tcp DVL Active Mail  
 dvl-activemail 1396/udp DVL Active Mail  
 audio-activmail 1397/tcp Audio Active Mail  
 audio-activmail 1397/udp Audio Active Mail  
 video-activmail 1398/tcp Video Active Mail  
 video-activmail 1398/udp Video Active Mail  
 # Ehud Shapiro <udi@wisdon.weizmann.ac.il>  
 cadkey-licman 1399/tcp Cadkey License Manager  
 cadkey-licman 1399/udp Cadkey License Manager  
 cadkey-tablet 1400/tcp Cadkey Tablet Daemon  
 cadkey-tablet 1400/udp Cadkey Tablet Daemon  
 # Joe McCollough <joe@cadkey.com>  
 goldleaf-licman 1401/tcp Goldleaf License Manager  
 goldleaf-licman 1401/udp Goldleaf License Manager  
 # John Fox <---none--->  
 prm-sm-np 1402/tcp Prospero Resource Manager  
 prm-sm-np 1402/udp Prospero Resource Manager  
 prm-nm-np 1403/tcp Prospero Resource Manager  
 prm-nm-np 1403/udp Prospero Resource Manager  
 # B. Clifford Neuman <bcn@isi.edu>  
 igi-lm 1404/tcp Infinite Graphics License Manager  
 igi-lm 1404/udp Infinite Graphics License Manager  
 ibm-res 1405/tcp IBM Remote Execution Starter  
 ibm-res 1405/udp IBM Remote Execution Starter  
 netlabs-lm 1406/tcp NetLabs License Manager  
 netlabs-lm 1406/udp NetLabs License Manager  
 dbsa-lm 1407/tcp DBSA License Manager  
 dbsa-lm 1407/udp DBSA License Manager  
 # Scott Shattuck <ss@dbsa.com>  
 sophia-lm 1408/tcp Sophia License Manager

sophia-lm 1408/udp Sophia License Manager

# Eric Brown <sst!emerald!eric@uunet.UU.net>

here-lm 1409/tcp Here License Manager

here-lm 1409/udp Here License Manager

# David Ison <here@dialup.oar.net>

hiq 1410/tcp HiQ License Manager

hiq 1410/udp HiQ License Manager

# Rick Pugh <rick@bilmillennium.com>

af 1411/tcp AudioFile

af 1411/udp AudioFile

# Jim Gettys <jg@crl.dec.com>

innosys 1412/tcp InnoSys

innosys 1412/udp InnoSys

innosys-acl 1413/tcp Innosys-ACL

innosys-acl 1413/udp Innosys-ACL

# Eric Welch <--none-->

ibm-mqseries 1414/tcp IBM MQSeries

ibm-mqseries 1414/udp IBM MQSeries

# Roger Meli <rmmeli%winvmd@vnet.ibm.com>

dbstar 1415/tcp DBStar

dbstar 1415/udp DBStar

# Jeffrey Millman <jcm@dbstar.com>

novell-lu6.2 1416/tcp Novell LU6.2

novell-lu6.2 1416/udp Novell LU6.2

# Peter Liu <--none-->

timbuktu-srv1 1417/tcp Timbuktu Service 1 Port

timbuktu-srv1 1417/udp Timbuktu Service 1 Port

timbuktu-srv2 1418/tcp Timbuktu Service 2 Port

timbuktu-srv2 1418/udp Timbuktu Service 2 Port

timbuktu-srv3 1419/tcp Timbuktu Service 3 Port

timbuktu-srv3 1419/udp Timbuktu Service 3 Port

timbuktu-srv4 1420/tcp Timbuktu Service 4 Port

timbuktu-srv4 1420/udp Timbuktu Service 4 Port

# Marc Epard <marc@waygate.farallon.com>

gandalf-lm 1421/tcp Gandalf License Manager

gandalf-lm 1421/udp Gandalf License Manager

# gilmer@gandalf.ca

autodesk-lm 1422/tcp Autodesk License Manager

autodesk-lm 1422/udp Autodesk License Manager

# David Ko <dko@autodesk.com>

```

essbase      1423/tcp  Essbase Arbor Software
essbase      1423/udp  Essbase Arbor Software
hybrid       1424/tcp  Hybrid Encryption Protocol
hybrid       1424/udp  Hybrid Encryption Protocol
#           Howard Hart <hch@hybrid.com>
zion-lm      1425/tcp  Zion Software License Manager
zion-lm      1425/udp  Zion Software License Manager
#           David Ferrero <david@zion.com>
sas-1        1426/tcp  Satellite-data Acquisition System 1
sas-1        1426/udp  Satellite-data Acquisition System 1
#           Bill Taylor <sais@ssec.wisc.edu>
mloadd       1427/tcp  mloadd monitoring tool
mloadd       1427/udp  mloadd monitoring tool
#           Bob Braden <braden@isi.edu>
informatik-lm 1428/tcp  Informatik License Manager
informatik-lm 1428/udp  Informatik License Manager
#           Harald Schlangmann
#           <schlangm@informatik.uni-muenchen.de>

nms          1429/tcp  Hypercom NMS
nms          1429/udp  Hypercom NMS
tpdu         1430/tcp  Hypercom TPDU
tpdu         1430/udp  Hypercom TPDU
#           Noor Chowdhury <noor@hypercom.com>
rgtp         1431/tcp  Reverse Gossip Transport
rgtp         1431/udp  Reverse Gossip Transport
#           Ian Jackson <iwj@cam-ork.co.uk>
blueberry-lm 1432/tcp  Blueberry Software License Manager

blueberry-lm 1432/udp  Blueberry Software License Manager

#           Steve Beigel <ublue!steve@uunet.uu.net>
ms-sql-s     1433/tcp  Microsoft-SQL-Server
ms-sql-s     1433/udp  Microsoft-SQL-Server
ms-sql-m     1434/tcp  Microsoft-SQL-Monitor
ms-sql-m     1434/udp  Microsoft-SQL-Monitor

#           Peter Hussey <peterhus@microsoft.com>
ibm-cics     1435/tcp  IBM CISC
ibm-cics     1435/udp  IBM CISC
#           Geoff Meacock <gbibmswl@ibmmail.COM>
sas-2        1436/tcp  Satellite-data Acquisition System 2
sas-2        1436/udp  Satellite-data Acquisition System 2
#           Bill Taylor <sais@ssec.wisc.edu>
tabula       1437/tcp  Tabula
tabula       1437/udp  Tabula
#           Marcelo Einhorn
#           <KGUNE%HUJIVM1.bitnet@taunivm.tau.ac.il>

eicon-server 1438/tcp  Eicon Security Agent/Server

```



eicon-server 1438/udp Eicon Security Agent/Server  
eicon-x25 1439/tcp Eicon X25/SNA Gateway  
eicon-x25 1439/udp Eicon X25/SNA Gateway  
eicon-slp 1440/tcp Eicon Service Location Protocol  
eicon-slp 1440/udp Eicon Service Location Protocol  
# Pat Calhoun <CALHOUN@admin.eicon.qc.ca>  
cadis-1 1441/tcp Cadis License Management  
cadis-1 1441/udp Cadis License Management  
cadis-2 1442/tcp Cadis License Management  
cadis-2 1442/udp Cadis License Management  
# Todd Wichers <twichers@csn.org>  
ies-lm 1443/tcp Integrated Engineering Software  
ies-lm 1443/udp Integrated Engineering Software  
# David Tong <David\_Tong@integrated.mb.ca>  
marcam-lm 1444/tcp Marcam License Management  
marcam-lm 1444/udp Marcam License Management  
# Therese Hunt <hunt@marcam.com>  
proxima-lm 1445/tcp Proxima License Manager  
proxima-lm 1445/udp Proxima License Manager  
ora-lm 1446/tcp Optical Research Associates License  
Manager  
ora-lm 1446/udp Optical Research Associates License  
Manager  
apri-lm 1447/tcp Applied Parallel Research LM  
apri-lm 1447/udp Applied Parallel Research LM  
# Jim Dillon <jed@apri.com>  
oc-lm 1448/tcp OpenConnect License Manager  
oc-lm 1448/udp OpenConnect License Manager  
# Sue Barnhill <snb@oc.com>  
peport 1449/tcp PEport  
peport 1449/udp PEport  
# Quentin Neill <quentin@ColumbiaSC.NCR.COM>  
dwf 1450/tcp Tandem Distributed Workbench Facility  
dwf 1450/udp Tandem Distributed Workbench Facility  
# Mike Bert <BERG\_MIKE@tandem.com>  
infoman 1451/tcp IBM Information Management  
infoman 1451/udp IBM Information Management  
# Karen Burns <---none--->  
gtegsc-lm 1452/tcp GTE Government Systems License Man

gtepsc-lm 1452/udp GTE Government Systems License Man  
 # Mike Gregory <Gregory\_Mike@msmail.iipo.gtepsc.com>  
 genie-lm 1453/tcp Genie License Manager  
 genie-lm 1453/udp Genie License Manager  
 # Paul Applegate <p.applegate2@genie.geis.com>  
 interhdl\_elmd 1454/tcp interHDL License Manager  
 interhdl\_elmd 1454/tcp interHDL License Manager  
 # Eli Sternheim eli@interhdl.com  
 esl-lm 1455/tcp ESL License Manager  
 esl-lm 1455/udp ESL License Manager  
 # Abel Chou <abel@willy.esl.com>  
 dca 1456/tcp DCA  
 dca 1456/udp DCA  
 # Jeff Garbers <jgarbers@netcom.com>  
 valisys-lm 1457/tcp Valisys License Manager  
 valisys-lm 1457/udp Valisys License Manager  
 # Leslie Lincoln <leslie\_lincoln@valisys.com>  
 nrcabq-lm 1458/tcp Nichols Research Corp.  
 nrcabq-lm 1458/udp Nichols Research Corp.  
 # Howard Cole <hcole@tumbleweed.nrcabq.com>  
 proshare1 1459/tcp Proshare Notebook Application  
 proshare1 1459/udp Proshare Notebook Application  
 proshare2 1460/tcp Proshare Notebook Application  
 proshare2 1460/udp Proshare Notebook Application  
 # Robin Kar <Robin\_Kar@ccm.hf.intel.com>  
 ibm\_wrless\_lan 1461/tcp IBM Wireless LAN  
 ibm\_wrless\_lan 1461/udp IBM Wireless LAN  
 # <flanne@vnet.IBM.COM>  
 world-lm 1462/tcp World License Manager  
 world-lm 1462/udp World License Manager  
 # Michael S Amirault <ambi@world.std.com>  
 nucleus 1463/tcp Nucleus  
 nucleus 1463/udp Nucleus  
 # Venky Nagar <venky@fafner.Stanford.EDU>  
 msl\_lmd 1464/tcp MSL License Manager  
 msl\_lmd 1464/udp MSL License Manager  
 # Matt Timmermans  
 pipes 1465/tcp Pipes Platform  
 pipes 1465/udp Pipes Platform mfarlin@peerlogic.com  
 # Mark Farlin <mfarlin@peerlogic.com>  
 oceansoft-lm 1466/tcp Ocean Software License Manager  
 oceansoft-lm 1466/udp Ocean Software License Manager  
 # Randy Leonard <randy@oceansoft.com>  
 csdmbase 1467/tcp CSDMBASE  
 csdmbase 1467/udp CSDMBASE  
 csdm 1468/tcp CSDM  
 csdm 1468/udp CSDM  
 # Robert Stabl <stabl@informatik.uni-muenchen.de>

aal-lm 1469/tcp Active Analysis Limited License Manager  
aal-lm 1469/udp Active Analysis Limited License Manager  
# David Snocken +44 (71)437-7009  
uaiact 1470/tcp Universal Analytics  
uaiact 1470/udp Universal Analytics  
# Mark R. Ludwig <Mark-Ludwig@uai.com>  
csdmbase 1471/tcp csdmbase  
csdmbase 1471/udp csdmbase  
csdm 1472/tcp csdm  
csdm 1472/udp csdm  
# Robert Stabl <stabl@informatik.uni-muenchen.de>  
openmath 1473/tcp OpenMath  
openmath 1473/udp OpenMath  
# Garth Mayville <mayville@maplesoft.on.ca>  
telefinder 1474/tcp Telefinder  
telefinder 1474/udp Telefinder  
# Jim White <Jim\_White@spiderisland.com>  
taligent-lm 1475/tcp Taligent License Manager  
taligent-lm 1475/udp Taligent License Manager  
# Mark Sapsford <Mark\_Sapsford@@taligent.com>  
clvm-cfg 1476/tcp clvm-cfg  
clvm-cfg 1476/udp clvm-cfg  
# Eric Soderberg <seric@cup.hp.com>  
ms-sna-server 1477/tcp ms-sna-server  
ms-sna-server 1477/udp ms-sna-server  
ms-sna-base 1478/tcp ms-sna-base  
ms-sna-base 1478/udp ms-sna-base  
# Gordon Mangione <gordm@microsoft.com>  
dberegister 1479/tcp dberegister  
dberegister 1479/udp dberegister  
# Brian Griswold <brian@dancingbear.com>  
pacerforum 1480/tcp PacerForum  
pacerforum 1480/udp PacerForum  
# Peter Caswell <pfc@pacvax.pacersoft.com>  
airs 1481/tcp AIRS  
airs 1481/udp AIRS  
# Bruce Wilson, 905-771-6161  
miteksys-lm 1482/tcp Miteksys License Manager  
miteksys-lm 1482/udp Miteksys License Manager  
# Shane McRoberts <mroberts@miteksys.com>  
afs 1483/tcp AFS License Manager  
afs 1483/udp AFS License Manager  
# Michael R. Pizolato <michael@afs.com>  
confluent 1484/tcp Confluent License Manager  
confluent 1484/udp Confluent License Manager  
# James Greenfiel <jim@pa.confluent.com>  
lansource 1485/tcp LANSource  
lansource 1485/udp LANSource  
# Doug Scott <lansourc@hookup.net>  
nms\_topo\_serv 1486/tcp nms\_topo\_serv  
nms\_topo\_serv 1486/udp nms\_topo\_serv

```

#           Sylvia Siu <Sylvia_Siu@Novell.CO>
localinfosrvr 1487/tcp  LocalInfoSrvr
localinfosrvr 1487/udp  LocalInfoSrvr
#           Brian Matthews <brian_matthews@ibist.ibis.com>
docstor       1488/tcp  DocStor
docstor       1488/udp  DocStor
#           Brian Spears <bspears@salix.com>
dmdocbroker   1489/tcp  dmdocbroker
dmdocbroker   1489/udp  dmdocbroker
#           Razmik Abnous <abnous@documentum.com>
insitu-conf   1490/tcp  insitu-conf
insitu-conf   1490/udp  insitu-conf
#           Paul Blacknell <paul@insitu.com>
anynetgateway 1491/tcp  anynetgateway
anynetgateway 1491/udp  anynetgateway
#           Dan Poirier <poirier@VNET.IBM.COM>
stone-design-1 1492/tcp  stone-design-1
stone-design-1 1492/udp  stone-design-1
#           Andrew Stone <andrew@stone.com>
netmap_lm     1493/tcp  netmap_lm
netmap_lm     1493/udp  netmap_lm
#           Phillip Magson <philm@extro.ucc.su.OZ.AU>
ica           1494/tcp  ica
ica           1494/udp  ica
#           John Richardson, Citrix Systems
cvc           1495/tcp  cvc
cvc           1495/udp  cvc
#           Bill Davidson <billd@equalizer.cray.com>
liberty-lm    1496/tcp  liberty-lm
liberty-lm    1496/udp  liberty-lm
#           Jim Rogers <traneljimbo@pacbell.com>
rfx-lm        1497/tcp  rfx-lm
rfx-lm        1497/udp  rfx-lm
#           Bill Bishop <bil@rfx.rfx.com>
watcom-sql    1498/tcp  Watcom-SQL
watcom-sql    1498/udp  Watcom-SQL
#           Rog Skubowius <rws kubow@ccnga.uwaterloo.ca>
fhc           1499/tcp  Federico Heinz Consultora
fhc           1499/udp  Federico Heinz Consultora
#           Federico Heinz <federico@heinz.com>
vlsi-lm       1500/tcp  VLSI License Manager
vlsi-lm       1500/udp  VLSI License Manager
#           Shue-Lin Kuo <shuelin@mdk.sanjose.vlsi.com>
sas-3         1501/tcp  Satellite-data Acquisition System
3
sas-3         1501/udp  Satellite-data Acquisition System
3
#           Bill Taylor <sais@ssec.wisc.edu>
shivadiscovery 1502/tcp  Shiva
shivadiscovery 1502/udp  Shiva
#           Jonathan Wenocur <jhw@Shiva.COM>

```

```

imtc-mcs      1503/tcp  Databeam
imtc-mcs      1503/udp  Databeam
#             Jim Johnstone <jjohnstone@databeam.com>
evb-elm       1504/tcp  EVB Software Engineering License Manager
evb-elm       1504/udp  EVB Software Engineering License Manager
#             B.G. Mahesh <mahesh@sett.com>
funkproxy     1505/tcp  Funk Software, Inc.
funkproxy     1505/udp  Funk Software, Inc.
#             Robert D. Vincent <bert@willowpond.com>
utcd          1506/tcp  Universal Time daemon (utcd)
utcd          1506/udp  Universal Time daemon (utcd)
#             Walter Poxon <wdp@ironwood.cray.com>
symplex       1507/tcp  symplex
symplex       1507/udp  symplex
#             Mike Turley <turley@symplex.com>
diagmond      1508/tcp  diagmond
diagmond      1508/udp  diagmond
#             Pete Moscatelli <moscat@hprdstl0.rose.hp.com>
robcad-lm     1509/tcp  Robcad, Ltd. License Manager
robcad-lm     1509/udp  Robcad, Ltd. License Manager
#             Hindin Joseph <hindin%robcad@uunet.uu.net>
mvx-lm        1510/tcp  Midland Valley Exploration Ltd. Lic.
Man.
mvx-lm        1510/udp  Midland Valley Exploration Ltd. Lic.
Man.
#             Charles X. Chen <charles@mvel.demon.co.uk>
3l-11        1511/tcp  3l-11
3l-11        1511/udp  3l-11
#             Ian A. Young <iay@threel.co.uk>
wins          1512/tcp  Microsoft's Windows Internet Name
Service
wins          1512/udp  Microsoft's Windows Internet Name
Service
#             Pradeep Bahl <pradeepb@microsoft.com>
fujitsu-dtc   1513/tcp  Fujitsu Systems Business of America,
Inc
fujitsu-dtc   1513/udp  Fujitsu Systems Business of America,
Inc
fujitsu-dtens 1514/tcp  Fujitsu Systems Business of America,
Inc
fujitsu-dtens 1514/udp  Fujitsu Systems Business of America,
Inc
#             Charles A. Higgins
#             <75730.2257@compuserve.com>
ifor-protocol 1515/tcp  ifor-protocol
ifor-protocol 1515/udp  ifor-protocol
#             Dr. R.P. Alston <robin@gradient.com>
vpad          1516/tcp  Virtual Places Audio data
vpad          1516/udp  Virtual Places Audio data

```

vpac 1517/tcp Virtual Places Audio control  
 vpac 1517/udp Virtual Places Audio control  
 vpvdc 1518/tcp Virtual Places Video data  
 vpvdc 1518/udp Virtual Places Video data  
 vpvcc 1519/tcp Virtual Places Video control  
 vpvcc 1519/udp Virtual Places Video control  
 # Ehud Shapiro <udi@ubique.co.il>  
 atm-zip-office 1520/tcp atm zip office  
 atm-zip-office 1520/udp atm zip office  
 # Wilson Kwan <wilsonk%toronto@zip.atm.com>  
 ncube-lm 1521/tcp nCube License Manager  
 ncube-lm 1521/udp nCube License Manager  
 # Maxine Yuen <maxine@hq.ncube.com>  
 rna-lm 1522/tcp Ricardo North America License Manager  
 rna-lm 1522/udp Ricardo North America License Manager  
 # MFlemming@aol.com  
 cichild-lm 1523/tcp cichild  
 cichild-lm 1523/udp cichild  
 # Andy Burgess <aab@cichlid.com>  
 ingreslock 1524/tcp ingres  
 ingreslock 1524/udp ingres  
 orasrv 1525/tcp oracle  
 orasrv 1525/udp oracle  
 prospero-np 1525/tcp Prospero Directory Service non-priv  
 prospero-np 1525/udp Prospero Directory Service non-priv  
 pdap-np 1526/tcp Prospero Data Access Prot non-priv  
  
 pdap-np 1526/udp Prospero Data Access Prot non-priv  
  
 # B. Clifford Neuman <bcn@isi.edu>  
 tlisrv 1527/tcp oracle  
 tlisrv 1527/udp oracle  
 mciautoreg 1528/tcp micautoreg  
 mciautoreg 1528/udp micautoreg  
 # John Klensin <klensin@MAIL1.RESTON.MCI.NET>  
 coauthor 1529/tcp oracle  
 coauthor 1529/udp oracle  
 rap-service 1530/tcp rap-service  
 rap-service 1530/udp rap-service  
 rap-listen 1531/tcp rap-listen  
 rap-listen 1531/udp rap-listen  
 # Phil Servita <meister@ftp.com>  
 miroconnect 1532/tcp miroconnect  
 miroconnect 1532/udp miroconnect  
 # Michael Fischer +49 531 21 13 0  
 virtual-places 1533/tcp Virtual Places Software  
 virtual-places 1533/udp Virtual Places Software  
 # Ehud Shapiro <udi@ubique.co.il>  
  
 micromuse-lm 1534/tcp micromuse-lm  
 micromuse-lm 1534/udp micromuse-lm

```

# Adam Kerrison <adam@micromuse.co.uk>
ampr-info 1535/tcp ampr-info
ampr-info 1535/udp ampr-info
ampr-inter 1536/tcp ampr-inter
ampr-inter 1536/udp ampr-inter
# Rob Janssen <rob@sys3.pe1chl.ampr.org>
sdsc-lm 1537/tcp isi-lm
sdsc-lm 1537/udp isi-lm
# Len Wanger <lrw@sdsc.edu>
3ds-lm 1538/tcp 3ds-lm
3ds-lm 1538/udp 3ds-lm
# Keith Trummel <ktrummel@autodesk.com>
intellistor-lm 1539/tcp Intellistor License Manager
intellistor-lm 1539/udp Intellistor License Manager
# Ron Vaughn <rv@intellistor.com>
rds 1540/tcp rds
rds 1540/udp rds
rds2 1541/tcp rds2
rds2 1541/udp rds2
# Sudhakar Rajamannar <mobius1@cerfnet.com>
gridgen-elmd 1542/tcp gridgen-elmd
gridgen-elmd 1542/udp gridgen-elmd
# John R. Chawner +1 817 354-1004
simba-cs 1543/tcp simba-cs
simba-cs 1543/udp simba-cs
# Betsy Alexander +1 604-681-4549
aspeclmd 1544/tcp aspeclmd
aspeclmd 1544/udp aspeclmd
# V. Balaji <balaji@aspec.com>
vistium-share 1545/tcp vistium-share
vistium-share 1545/udp vistium-share
# Allison Carleton <acarleto@naper1.napervilleil.ncr.com>
abbaccuray 1546/tcp abbaccuray
abbaccuray 1546/udp abbaccuray
# John Wendt 614-261-2000
laplink 1547/tcp laplink
laplink 1547/udp laplink
# Michael Crawford <MichaelC@dev.travsoft.com>
axon-lm 1548/tcp Axon License Manager
axon-lm 1548/udp Axon License Manager
# Mark Pearce <<Mark_A..Pearce/AXON_Networks_Inc..@notes.axon.com>
shivahose 1549/tcp Shiva Hose
shivasound 1549/udp Shiva Sound
# Kin Chan <kchan@shiva.com>
3m-image-lm 1550/tcp Image Storage license manager 3M Company
3m-image-lm 1550/udp Image Storage license manager 3M Company
# J. C. Canessa <jccanessa@mmm.com>
hecmtl-db 1551/tcp HECMTL-DB
hecmtl-db 1551/udp HECMTL-DB
# Maxime Belanger <R173@hec.ca>
pciarray 1552/tcp pciarray

```

```

pciarray 1552/udp pciarray
# Ron Folk <rfolkes@avl.com>
sna-cs 1553/tcp sna-cs
sna-cs 1553/udp sna-cs
# Tony Sowter <ts@datcon.co.uk>
caci-lm 1554/tcp CACI Products Company License Manager
caci-lm 1554/udp CACI Products Company License Manager
# Erik Blume <erikb@caciasl.com>
livelan 1555/tcp livelan
livelan 1555/udp livelan
# khedayat@roadrunner.pictel.com <Kaynam
Hedayat>
ashwin 1556/tcp AshWin CI Technologies
ashwin 1556/udp AshWin CI Technologies
# Dave Neal <daven@ashwin.com>
arbortext-lm 1557/tcp ArborText License Manager
arbortext-lm 1557/udp ArborText License Manager
# David J. Wilson <djwt@arbortext.com>
xingmpeg 1558/tcp xingmpeg
xingmpeg 1558/udp xingmpeg
# Howard Gordon <hgordon@system.xingtech.com>
web2host 1559/tcp web2host
web2host 1559/udp web2host
# Stephen Johnson <sjohnson@mindspring.com>
ascii-val 1560/tcp ascii-val
ascii-val 1560/udp ascii-val
# Brian Schenkenberger <brians@advsyscon.com>
facilityview 1561/tcp facilityview
facilityview 1561/udp facilityview
# Ed Green <egreen@pmeasuring.com>
pconnectmgr 1562/tcp pconnectmgr
pconnectmgr 1562/udp pconnectmgr
# Bob Kaiser <BKaiser@palindrome.com>
cadabra-lm 1563/tcp Cadabra License Manager
cadabra-lm 1563/udp Cadabra License Manager
# Arthur Castonguay <arthurc@doe.carleton.ca>
pay-per-view 1564/tcp Pay-Per-View
pay-per-view 1564/udp Pay-Per-View
# Brian Tung <brian@isi.edu>
winddlb 1565/tcp WinDD
winddlb 1565/udp WinDD
# Kelly Sims <kellys@garnet.wv.tek.com>
corelvideo 1566/tcp CORELVIDEO
corelvideo 1566/udp CORELVIDEO
# Ming Poon <mingp@corel.ca>
jlicelmd 1567/tcp jlicelmd
jlicelmd 1567/udp jlicelmd
# Christian Schormann <100410.3063@compuserve.com>
tsspmap 1568/tcp tsspmap
tsspmap 1568/udp tsspmap
# Paul W. Nelson <nelson@thursby.com>

```



```

ets      1569/tcp  ets
ets      1569/udp  ets
#        Carstein Seeberg <case@boole.no>

orbixd   1570/tcp  orbixd
orbixd   1570/udp  orbixd
#        Bridget Walsh <bwalsh@iona.ie>
rdb-dbs-disp 1571/tcp  Oracle Remote Data Base
rdb-dbs-disp 1571/udp  Oracle Remote Data Base
#        <mackin@us.oracle.com>
chip-lm  1572/tcp  Chipcom License Manager
chip-lm  1572/udp  Chipcom License Manager
#        Jerry Natowitz <Jerry Natowitz>
itscomm-ns 1573/tcp  itscomm-ns
itscomm-ns 1573/udp  itscomm-ns
#        Rich Thompson <richt@watson.ibm.com>

mvel-lm  1574/tcp  mvel-lm
mvel-lm  1574/udp  mvel-lm
#        David Bisset <dbisset@mvel.demon.co.uk>
oraclenames 1575/tcp  oraclenames
oraclenames 1575/udp  oraclenames
#        P.V.Shivkumar <PSHIVKUM@us.oracle.com>
moldflow-lm 1576/tcp  moldflow-lm
moldflow-lm 1576/udp  moldflow-lm
#        Paul Browne <browne@moldflow.com.au>
hypercube-lm 1577/tcp  hypercube-lm
hypercube-lm 1577/udp  hypercube-lm
#        Michael Moller <moller@hyper.hyper.com>
jacobus-lm 1578/tcp  Jacobus License Manager
jacobus-lm 1578/udp  Jacobus License Manager
#        Tony Cleveland <tony.cleveland@jacobus.com>
ioc-sea-lm 1579/tcp  ioc-sea-lm
ioc-sea-lm 1579/udp  ioc-sea-lm
#        Paul Nelson <paul@ioc-sea.com>
tn-tl-r1  1580/tcp  tn-tl-r1
tn-tl-r2  1580/udp  tn-tl-r2
#        Ed Kress <eskress@thinknet.com>
vmf-msg-port 1581/tcp  vmf-msg-port
vmf-msg-port 1581/udp  vmf-msg-port
#        Eric Whitehill <eawhiteh@itt.com>
tams-lm    1582/tcp  Toshiba America Medical Systems

tams-lm    1582/udp  Toshiba America Medical Systems

#        Philip Scott<pks@smtp.orasis.com>
simbaexpress 1583/tcp  simbaexpress
simbaexpress 1583/udp  simbaexpress
#        Betsy Alexander +1 604-681-4549
#        1584-1599 Unassigned
issd 1600/tcp

```

```

issd 1600/udp
#      1601-1641 Unassigned
isis-am  1642/tcp  isis-am
isis-am  1642/udp  isis-am
isis-ambc 1643/tcp  isis-ambc
isis-ambc 1643/udp  isis-ambc
#      Ken Chapman <kchapman@isis.com>
#      1644-1649 Unassigned
nkd 1650/tcp
nkd 1650/udp
shiva_confsrvr 1651/tcp  shiva_confsrvr
shiva_confsrvr 1651/udp  shiva_confsrvr
#      Mike Horowitz <mah@Shiva.COM>
xntp  1652/tcp  xntp
xntp  1652/udp  xntp
#      Ali Saleh <scomm@cerf.net>
#      1653-1660 Unassigned
netview-aix-1 1661/tcp  netview-aix-1
netview-aix-1 1661/udp  netview-aix-1
netview-aix-2 1662/tcp  netview-aix-2
netview-aix-2 1662/udp  netview-aix-2
netview-aix-3 1663/tcp  netview-aix-3
netview-aix-3 1663/udp  netview-aix-3
netview-aix-4 1664/tcp  netview-aix-4
netview-aix-4 1664/udp  netview-aix-4
netview-aix-5 1665/tcp  netview-aix-5
netview-aix-5 1665/udp  netview-aix-5
netview-aix-6 1666/tcp  netview-aix-6
netview-aix-6 1666/udp  netview-aix-6
netview-aix-7 1667/tcp  netview-aix-7
netview-aix-7 1667/udp  netview-aix-7
netview-aix-8 1668/tcp  netview-aix-8
netview-aix-8 1668/udp  netview-aix-8
netview-aix-9 1669/tcp  netview-aix-9
netview-aix-9 1669/udp  netview-aix-9
netview-aix-10 1670/tcp  netview-aix-10
netview-aix-10 1670/udp  netview-aix-10
netview-aix-11 1671/tcp  netview-aix-11
netview-aix-11 1671/udp  netview-aix-11
netview-aix-12 1672/tcp  netview-aix-12
netview-aix-12 1672/udp  netview-aix-12
#      Martha Crisson <CRISSON@ralvm12.vnet.ibm.com>
#      1673-1987 Unassigned
licensedaemon 1986/tcp  cisco license management
licensedaemon 1986/udp  cisco license management
tr-rsrb-p1  1987/tcp  cisco RSRB Priority 1 port
tr-rsrb-p1  1987/udp  cisco RSRB Priority 1 port
tr-rsrb-p2  1988/tcp  cisco RSRB Priority 2 port
tr-rsrb-p2  1988/udp  cisco RSRB Priority 2 port
tr-rsrb-p3  1989/tcp  cisco RSRB Priority 3 port
tr-rsrb-p3  1989/udp  cisco RSRB Priority 3 port

```

#PROBLEMS!=====

mshnet 1989/tcp MHSnet system  
mshnet 1989/udp MHSnet system  
# Bob Kummerfeld <bob@sarad.cs.su.oz.au>

#PROBLEMS!=====

stun-p1 1990/tcp cisco STUN Priority 1 port  
stun-p1 1990/udp cisco STUN Priority 1 port  
stun-p2 1991/tcp cisco STUN Priority 2 port  
stun-p2 1991/udp cisco STUN Priority 2 port  
stun-p3 1992/tcp cisco STUN Priority 3 port  
stun-p3 1992/udp cisco STUN Priority 3 port

#PROBLEMS!=====

ipsendmsg 1992/tcp IPsendmsg  
ipsendmsg 1992/udp IPsendmsg  
# Bob Kummerfeld <bob@sarad.cs.su.oz.au>

#PROBLEMS!=====

snmp-tcp-port 1993/tcp cisco SNMP TCP port  
snmp-tcp-port 1993/udp cisco SNMP TCP port  
stun-port 1994/tcp cisco serial tunnel port  
stun-port 1994/udp cisco serial tunnel port  
perf-port 1995/tcp cisco perf port  
perf-port 1995/udp cisco perf port  
tr-rsrb-port 1996/tcp cisco Remote SRB port  
tr-rsrb-port 1996/udp cisco Remote SRB port  
gdp-port 1997/tcp cisco Gateway Discovery Protocol  
gdp-port 1997/udp cisco Gateway Discovery Protocol  
x25-svc-port 1998/tcp cisco X.25 service (XOT)  
x25-svc-port 1998/udp cisco X.25 service (XOT)  
tcp-id-port 1999/tcp cisco identification port  
tcp-id-port 1999/udp cisco identification port  
callbook 2000/tcp  
callbook 2000/udp  
dc 2001/tcp  
wizard 2001/udp curry  
globe 2002/tcp  
globe 2002/udp  
mailbox 2004/tcp  
emce 2004/udp CCWS mm conf  
berknet 2005/tcp  
oracle 2005/udp  
invokator 2006/tcp  
raid-cc 2006/udp raid  
dectalk 2007/tcp  
raid-am 2007/udp  
conf 2008/tcp  
terminaldb 2008/udp  
news 2009n/tcp  
whosockami 2009/udp  
search 2010/tcp

pipe\_server 2010/udp  
raid-cc 2011/tcp raid  
servserv 2011/udp  
ttyinfo 2012/tcp  
raid-ac 2012/udp  
raid-am 2013/tcp  
raid-cd 2013/udp  
troff 2014/tcp  
raid-sf 2014/udp  
cypress 2015/tcp  
raid-cs 2015/udp  
bootserver 2016/tcp  
bootserver 2016/udp  
cypress-stat 2017/tcp  
bootclient 2017/udp  
terminaldb 2018/tcp  
rellpack 2018/udp  
whosockami 2019/tcp  
about 2019/udp  
xinupageserver 2020/tcp  
xinupageserver 2020/udp  
servexec 2021/tcp  
xinuexpansion1 2021/udp  
down 2022/tcp  
xinuexpansion2 2022/udp  
xinuexpansion3 2023/tcp  
xinuexpansion3 2023/udp  
xinuexpansion4 2024/tcp  
xinuexpansion4 2024/udp  
ellpack 2025/tcp  
xribs 2025/udp  
scrabble 2026/tcp  
scrabble 2026/udp  
shadowserver 2027/tcp  
shadowserver 2027/udp  
submitserver 2028/tcp  
submitserver 2028/udp  
device2 2030/tcp  
device2 2030/udp  
blackboard 2032/tcp  
blackboard 2032/udp  
glogger 2033/tcp  
glogger 2033/udp  
scoremgr 2034/tcp  
scoremgr 2034/udp  
imslodoc 2035/tcp  
imslodoc 2035/udp  
objectmanager 2038/tcp  
objectmanager 2038/udp  
lam 2040/tcp  
lam 2040/udp

```

interbase 2041/tcp
interbase 2041/udp
isis 2042/tcp isis
isis 2042/udp isis
isis-bcast 2043/tcp isis-bcast
isis-bcast 2043/udp isis-bcast
#           Ken Chapman <kchapman@isis.com>
ivs-video  2232/udp IVS Video default
rimsl 2044/tcp
rimsl 2044/udp
cdfunc 2045/tcp
cdfunc 2045/udp
sdfunc 2046/tcp
sdfunc 2046/udp
dls 2047/tcp
dls 2047/udp
dls-monitor 2048/tcp
dls-monitor 2048/udp
shilp 2049/tcp
shilp 2049/udp
dlsrcpn    2065/tcp Data Link Switch Read Port Number
dlsrcpn    2065/udp Data Link Switch Read Port Number
dlswpn     2067/tcp Data Link Switch Write Port Number
dlswpn     2067/udp Data Link Switch Write Port Number
ats        2201/tcp Advanced Training System Program
ats        2201/udp Advanced Training System Program
#
ivs-video  2232/tcp IVS Video default
ivs-video  2232/udp IVS Video default
#   Thierry Turletti <Thierry.Turletti@sophia.inria.fr>
ivsd       2241/tcp IVS Daemon
ivsd       2241/udp IVS Daemon
#   Thierry Turletti <Thierry.Turletti@sophia.inria.fr>
pehelp    2307/tcp pehelp
pehelp    2307/udp pehelp
#           Jens Kilian <jensk@hpbeo82.bbn.hp.com>
#
rtsserv   2500/tcp Resource Tracking system server
rtsserv   2500/udp Resource Tracking system server
rtsclient 2501/tcp Resource Tracking system client
rtsclient 2501/udp Resource Tracking system client
#           Aubrey Turner
#           <S95525ta%etsuacad.bitnet@ETSUADMN.ETSU.EDU>
hp-3000-telnet 2564/tcp HP 3000 NS/VT block mode telnet
www-dev    2784/tcp world wide web - development
www-dev    2784/udp world wide web - development
NSWS 3049/tcp
NSWS 3049/udp
vmodem    3141/tcp VMODEM
vmodem    3141/udp VMODEM
#           Ray Gwinn <p00321@psilink.com>

```

```

ccmail      3264/tcp  cc:mail/lotus
ccmail      3264/udp  cc:mail/lotus
dec-notes   3333/tcp  DEC Notes
dec-notes   3333/udp  DEC Notes
#           Kim Moraros <moraros@via.enet.dec.com>
mapper-nodemgr 3984/tcp  MAPPER network node manager
mapper-nodemgr 3984/udp  MAPPER network node manager
mapper-mapethd 3985/tcp  MAPPER TCP/IP server
mapper-mapethd 3985/udp  MAPPER TCP/IP server
mapper-ws_ethd 3986/tcp  MAPPER workstation server
mapper-ws_ethd 3986/udp  MAPPER workstation server
#           John C. Horton <jch@unirsvl.rsvl.unisys.com>
bmap        3421/tcp  Bull Apprise portmapper
bmap        3421/udp  Bull Apprise portmapper
#           Jeremy Gilbert <J.Gilbert@ma30.bull.com>
#
prsvp       3455/tcp  RSVP Port
prsvp       3455/udp  RSVP Port
#           Bob Braden <Braden@isi.edu>
vat         3456/tcp  VAT default data
vat         3456/udp  VAT default data
#           Van Jacobson <van@ee.lbl.gov>
vat-control 3457/tcp  VAT default control
vat-control 3457/udp  VAT default control
#           Van Jacobson <van@ee.lbl.gov>
#
udt_os      3900/tcp  Unidata UDT OS
udt_os      3900/udp  Unidata UDT OS
#           James Powell <james@mailhost.unidata.com>
netcheque   4008/tcp  NetCheque accounting
netcheque   4008/udp  NetCheque accounting
#           B. Clifford Neuman <bcn@isi.edu>
nuts_dem    4132/tcp  NUTS Daemon
nuts_dem    4132/udp  NUTS Daemon
nuts_bootp  4133/tcp  NUTS Bootp Server
nuts_bootp  4133/udp  NUTS Bootp Server
#           Martin Freiss <freiss.pad@sni.>
rwhois      4321/tcp  Remote Who Is
rwhois      4321/udp  Remote Who Is
#           Mark Kusters <markk@internic.net>
unicall     4343/tcp  UNICALL
unicall     4343/udp  UNICALL
#           James Powell <james@enghp.unidata.comp>
krb524      4444/tcp  KRB524
krb524      4444/udp  KRB524
#           B. Clifford Neuman <bcn@isi.edu>
# PROBLEM krb524 assigned the port,
# PROBLEM nv used it without an assignment
nv-video    4444/tcp  NV Video default
nv-video    4444/udp  NV Video default
#           Ron Frederick <frederick@parc.xerox.com>

```

```

#
sae-urn      4500/tcp sae-urn
sae-urn      4500/udp sae-urn
urn-x-cdchoice 4501/tcp urn-x-cdchoice
urn-x-cdchoice 4501/udp urn-x-cdchoice
#           Paul Hoffman <phoffman@proper.com>
rfa         4672/tcp  remote file access server
rfa         4672/udp  remote file access server
complex-main 5000/tcp
complex-main 5000/udp
complex-link 5001/tcp
complex-link 5001/udp
rfe         5002/tcp  radio free ethernet
rfe         5002/udp  radio free ethernet
claris-fmpro 5003/tcp  Claris FileMaker Pro
claris-fmpro 5003/udp  Claris FileMaker Pro
#           Jon Thatcher <jon_thatcher@qm.claris.com>
telepathstart 5010/tcp  TelepathStart
telepathstart 5010/udp  TelepathStart
telepathattack 5011/tcp  TelepathAttack
telepathattack 5011/udp  TelepathAttack
#           Helmut Breitenfellner <hbreitenf@vnet.imb.com>
mmcc        5050/tcp  multimedia conference control tool
mmcc        5050/udp  multimedia conference control tool
#           Steve Casner <Casner@isi.edu>
rmonitor_secure 5145/tcp
rmonitor_secure 5145/udp
aol         5190/tcp  America-Online
aol         5190/udp  America-Online
#           Marty Lyons <marty@aol.com>
aol-1      5191/tcp  AmericaOnline1
aol-1      5191/udp  AmericaOnline1
aol-2      5192/tcp  AmericaOnline2
aol-2      5192/udp  AmericaOnline2
aol-3      5193/tcp  AmericaOnline3
aol-3      5193/udp  AmericaOnline3
#           Bruce Mackey <BAMackey@aol.com>
padl2sim 5236/tcp
padl2sim 5236/udp
hacl-hb 5300/tcp # HA cluster heartbeat
hacl-hb 5300/udp # HA cluster heartbeat
hacl-gs 5301/tcp # HA cluster general services
hacl-gs 5301/udp # HA cluster general services
hacl-cfg 5302/tcp # HA cluster configuration
hacl-cfg 5302/udp # HA cluster configuration
hacl-probe 5303/tcp # HA cluster probing
hacl-probe 5303/udp # HA cluster probing
hacl-local 5304/tcp
hacl-local 5304/udp
hacl-test 5305/tcp
hacl-test 5305/udp

```

```

#                Eric Soderberg <seric@hposl102.cup.hp>
proshareaudio 5713/tcp  proshare conf audio
proshareaudio 5713/udp  proshare conf audio
prosharevideo 5714/tcp  proshare conf video
prosharevideo 5714/udp  proshare conf video
prosharedata  5715/tcp  proshare conf data
prosharedata  5715/udp  proshare conf data
prosharerequest 5716/tcp  proshare conf request
prosharerequest 5716/udp  proshare conf request
prosharenotify 5717/tcp  proshare conf notify
prosharenotify 5717/udp  proshare conf notify
#                <gunner@ibeam.intel.com>
x11           6000-6063/tcp  X Window System
x11           6000-6063/udp  X Window System
#                Stephen Gildea <gildea@expo.lcs.mit.edu>
softcm        6110/tcp  HP SoftBench CM
softcm        6110/udp  HP SoftBench CM
spc           6111/tcp  HP SoftBench Sub-Process Control
spc           6111/udp  HP SoftBench Sub-Process Control
#                Scott A. Kramer <sk@tleilaxu.sde.hp.com>
dtspcd        6112/tcp  dtspcd
dtspcd        6112/udp  dtspcd
#                Doug Royer <Doug.Royer@eng.sun.com>
meta-corp     6141/tcp  Meta Corporation License Manager
meta-corp     6141/udp  Meta Corporation License Manager
#                Osamu Masuda <---none--->
aspentec-lm   6142/tcp  Aspen Technology License Manager
aspentec-lm   6142/udp  Aspen Technology License Manager
#                Kevin Massey <massey@aspentec.com>
watershed-lm  6143/tcp  Watershed License Manager
watershed-lm  6143/udp  Watershed License Manager
#                David Ferrero <david@zion.com>
statsci1-lm   6144/tcp  StatSci License Manager - 1
statsci1-lm   6144/udp  StatSci License Manager - 1
statsci2-lm   6145/tcp  StatSci License Manager - 2
statsci2-lm   6145/udp  StatSci License Manager - 2
#                Scott Blachowicz <scott@statsci.com>
lonewolf-lm   6146/tcp  Lone Wolf Systems License Manager
lonewolf-lm   6146/udp  Lone Wolf Systems License Manager
#                Dan Klein <dvk@lonewolf.com>
montage-lm    6147/tcp  Montage License Manager
montage-lm    6147/udp  Montage License Manager
#                Michael Ubell <michael@montage.com>
ricardo-lm    6148/tcp  Ricardo North America License Manager
ricardo-lm    6148/udp  Ricardo North America License Manager
#                M Flemming <mflemming@aol.com>
xdsxdm        6558/tcp
xdsxdm        6558/udp
acmsoda       6969/tcp  acmsoda
acmsoda       6969/udp  acmsoda
#                Daniel Simms <dsimms@acm.uiuc.edu>

```



```

afs3-fileserver 7000/tcp  file server itself
afs3-fileserver 7000/udp  file server itself
afs3-callback 7001/tcp  callbacks to cache managers
afs3-callback 7001/udp  callbacks to cache managers
afs3-prserver 7002/tcp  users & groups database
afs3-prserver 7002/udp  users & groups database
afs3-vlserver 7003/tcp  volume location database
afs3-vlserver 7003/udp  volume location database
afs3-kaserver 7004/tcp  AFS/Kerberos authentication service
afs3-kaserver 7004/udp  AFS/Kerberos authentication service
afs3-volser 7005/tcp  volume management server
afs3-volser 7005/udp  volume management server
afs3-errors 7006/tcp  error interpretation service
afs3-errors 7006/udp  error interpretation service
afs3-bos 7007/tcp  basic overseer process
afs3-bos 7007/udp  basic overseer process
afs3-update 7008/tcp  server-to-server updater
afs3-update 7008/udp  server-to-server updater
afs3-rmtsys 7009/tcp  remote cache manager service
afs3-rmtsys 7009/udp  remote cache manager service
ups-onlinet 7010/tcp  onlinet uninterruptable power supplies
ups-onlinet 7010/udp  onlinet uninterruptable power supplies
#      Brian Hammill <hamill@dolphin.exide.com>
font-service 7100/tcp  X Font Service
font-service 7100/udp  X Font Service
#      Stephen Gildea <gildea@expo.lcs.mit.edu>
fodms 7200/tcp  FODMS FLIP
fodms 7200/udp  FODMS FLIP
#      David Anthony <anthony@power.amasd.anatcp.rockwell.com>
dlip 7201/tcp  DLIP
dlip 7201/udp  DLIP
#      Albert Manfredi <manfredi@enr05.comsys.rockwell.com>
nmp 8450/tcp  nmp
nmp 8450/udp  nmp
#      Ian Chard <ian@tanagra.demon.co.uk>
man 9535/tcp
man 9535/udp
sd 9876/tcp  Session Director
sd 9876/udp  Session Director
#      Van Jacobson <van@ee.lbl.gov>
distinct 9999/tcp  distinct
distinct 9999/udp  distinct
#      Anoop Tewari <anoop@next.distinct.com>
isode-dua 17007/tcp
isode-dua 17007/udp
biimenu 18000/tcp Beckman Instruments, Inc.
biimenu 18000/udp Beckman Instruments, Inc.
R. L. Meyering <RLMEYERING@BIIVAX.DP.BECKMAN.COM>
icl-twobase1 25000/tcp icl-twobase1
icl-twobase1 25000/udp icl-twobase1
icl-twobase2 25001/tcp icl-twobase2

```

icl-twobase2 25001/udp icl-twobase2  
icl-twobase3 25002/tcp icl-twobase3  
icl-twobase3 25002/udp icl-twobase3  
icl-twobase4 25003/tcp icl-twobase4  
icl-twobase4 25003/udp icl-twobase4  
icl-twobase5 25004/tcp icl-twobase5  
icl-twobase5 25004/udp icl-twobase5  
icl-twobase6 25005/tcp icl-twobase6  
icl-twobase6 25005/udp icl-twobase6  
icl-twobase7 25006/tcp icl-twobase7  
icl-twobase7 25006/udp icl-twobase7  
icl-twobase8 25007/tcp icl-twobase8  
icl-twobase8 25007/udp icl-twobase8  
icl-twobase9 25008/tcp icl-twobase9  
icl-twobase9 25008/udp icl-twobase9  
icl-twobase10 25009/tcp icl-twobase10  
icl-twobase10 25009/udp icl-twobase10  
# J. A. (Tony) Sever <J.A.Sever@bra0119.wins.icl.co.uk>  
dbbrowse 47557/tcp Databeam Corporation  
dbbrowse 47557/udp Databeam Corporation  
# Cindy Martin <cmartin@databeam.com>

#### REFERENCES

[RFC768] Postel, J., "User Datagram Protocol", STD 6,  
RFC 768,  
USC/Information Sciences Institute, August 1980.

[RFC793] Postel, J., ed., "Transmission Control Protocol

- DARPA  
Internet Program Protocol Specification", STD 7,

RFC 793,

USC/Information Sciences Institute, September 1981.