

# Denial of Service

**By:**

Nancy Chan  
Rob Lockwood  
Stephan Freeman  
Pavan Farmah  
Costas Chousiadis  
Fatima Hamid  
Marc Diedert  
Paul Levy  
Keum Hoon  
Richard Lewis  
Richard Sreeharan  
Toshihiko Kamon  
John Mitsianis  
Ari Das-Purkayastha  
Yong Wook Chung

# Index

<b>INTRODUCTION.....</b>	<b>2</b>
<b>PHYSICAL ATTACKS.....</b>	<b>6</b>
<b>STANDALONE HOST ATTACKS</b>	
<b>Destructive and Resource Exhaustion .....</b>	<b>7</b>
<b>Logic Bombs .....</b>	<b>12</b>
<b>Trojan Horses .....</b>	<b>13</b>
<b>Viruses .....</b>	<b>15</b>
<b>NETWORKED HOST ATTACKS</b>	
<b>Transport and Network Layer .....</b>	<b>25</b>
<b>Application Layer .....</b>	<b>29</b>
<b>INFORMATION WARFARE .....</b>	<b>32</b>
<b>CASE STUDIES .....</b>	<b>37</b>
<b>SUMMARY .....</b>	<b>39</b>
<b>BIBLIOGRAPHY .....</b>	<b>41</b>
<b>APPENDIX.....</b>	<b>44</b>

# INTRODUCTION

## What is Denial of Service?

There are many definitions of denial of service (DoS). All allude to some compromise of resource availability. Loosely speaking, DoS is a threat that prevents legitimate users getting access to the information or resource that they need, when they need it. Some definitions are listed in the table below:

Source	Definition
ISO 7498-2	"The prevention of authorised access to resources or the delaying of time critical operations"
NIST	"...actions that prevent a network element from functioning in accordance with its intended purpose. Network elements may be rendered partially or entirely unusable for legitimate users. Denial of service may cause operations which depend on timeliness to be delayed."
NSA Glossary	"Action(s), which prevent any part of an AIS [Automated Information System] from functioning in accordance with its intended purpose."
Edward Amoroso	"...the DOS [Denial of Service] threat will be defined to occur when a service associated with a maximum waiting time (denoted <i>MWT</i> ) is requested by a user at time <i>t</i> and is not provided to that user by the time ( <i>t + MWT</i> )."

The definition of 'resource' in the context of DoS is broad, and ranges from physical hardware and network equipment to application software. Essentially, any hardware and software element of a networked computer system is at risk.

It is important to realise that resources have dependencies, and attacks against the dependencies may often be more damaging than attacking the resource itself. For example, vulnerabilities in routers and other network components might be exploited to overload a subnet or perhaps prevent packets from being delivered to the correct local destinations. This would have the effect of denying legitimate users access to all hosts on the attacked subnet.

## Is Denial of Service a likely threat?

### **Non-intentional Causes**

Causes of DoS can be accidental as well as deliberate. Accidental threats are arguably the most varied and least predictable, and are therefore difficult to specifically protect against. The effects of this type of threat can perhaps be minimised by having good general safeguards in place: for example, making sure key equipment is physically secured, having well defined problem escalation procedures, formulating an IT disaster recovery and business contingency plan, ensuring employees have adequate awareness of security risks and what they should do if a threat materialises.

Examples of accidental threats include:

- Computer equipment damaged or destroyed by fire, flood, smoke, lightning and

- JCB digger inadvertently cutting through an organisation's external leased line, thereby severing all communications to its ISP (Internet Service Provider) and/or business partners;
- First run of bug-ridden program causes it to run in a recursive loop such that disk space and CPU time are eventually exhausted;
- Accidental disconnection of the power supply to business critical equipment. One story is of a company who had recently bought a new enterprise server, and couldn't work out why it was mysteriously re-booting itself each night. They eventually found out that the cleaners, who came in each evening, had been pulling the plug on the server, to free up a power socket for their vacuum cleaner, before replacing the plug afterwards!

## **Deliberate Causes**

The deliberate threat can be further categorised into 'deliberate-authorized' and 'deliberate-unauthorized' incidents. The latter category includes the kinds of attacks most commonly associated with DoS. These two categories are further explained below.

### *Deliberate-Authorised Denial of Service*

In some situations, Denial of Service can be an entirely authorised act. Models of Denial of Service are in existence that allow an entity to define the conditions under which Denial of Service is not allowed. Incidents of DoS that do not satisfy these conditions are construed to be authorised. Two such models are the Mandatory DoS model (Amoroso) and the Resource Allocation model (Millen).

*In order to define the denial of service problem precisely we need to introduce the entities involved and the relationships among them.*

**V.Gligor**

*Some individuals are specifically authorised to take service-denying actions, such as deleting user accounts, and disconnecting the system from networks. By definition, users with high priorities for resource use are authorised to deny service to lower priority users.*

**D.Sterne**

From the above two quotations, the topic of access controls becomes apparent as they both address the level of authorisation an entity can have, which can define whether a denial of service has occurred or not.

Expanding upon Amoroso's definition given in the table, earlier, DoS violation may initially be defined as cases where authorised users are not provided a requested service. Gligor was the first to take time into this notion of service grant or denial. Temporal logic has been used by the real-time systems community for several years, with Gligor now applying it to the concepts of DoS. The Maximum Waiting Time (MWT) defines the period of time for which a requested service is not stale. For example, if an aircraft flight controller requests position information while the aircraft rolls in some direction, then any delay in provision will cause the position information to become stale as the aircraft's position changes. A DOS threat can be defined whenever a service with associated MWT is requested by an authorised user at a

time  $t$  and is not provided to that user on time ( $t + MWT$ ). Therefore, requests to authorised users should not be late.

Amoroso's mandatory DoS model draws parallels with the Biba and Bell LaPadula integrity and confidentiality models by assuming that subjects have different levels of priority or authority. Two rules are defined: the first rule, No Deny Up (NDU), specifies that no subject can deny service to a subject with a greater or equal priority. The second rule allows one to restrict denial of service protection to a specific set of objects only. This second rule significantly reduces the cost and difficulty of implementing a DoS policy.

An example of the NDU rule is the following:

Suppose a company provides Internet access for its employees, and the system administrator is responsible for monitoring usage. If a particular employee subsequently abuses this facility by surfing inappropriate web sites, the system administrator may suspend that employee's Internet access.

From the employee's point of view, he/she is a victim of DoS. However, if the system administrator has higher authority (or in Amoroso's terms, a higher priority) than the suspended employee, then this DoS is deemed to be authorised within Amoroso's mandatory DoS model.

Further details of the DoS models mentioned here can be found in chapter 14 of Amoroso's book 'Fundamentals of Computer Security Technology'.

To summarise, the denial of service threat occurs whenever an authorised user is not granted a requested service within a defined maximum waiting time. A simple temporal logic expression of a DOS requirement can be used to illustrate DOS concepts. Two rules comprise a mandatory DOS model that specifies the avoidance of malicious DOS threats. (Amoroso) The application specific attacks such as exploiting software bugs are often easier to fix than the attacks based on weaknesses in standardised internet protocols.

#### *Deliberate-Unauthorised Denial of Service*

There are no tangible benefits to be gained from successful DoS attacks, although in some cases DoS is the first step in a larger planned attack. For example, applications or programs that are known to fail to an insecure state may be crashed deliberately to allow the attacker to obtain access to superuser privileges (a 'root' shell) on a particular web server. From here, the compromised web server could be a stepping-stone to attacks on a corporation's internal network.

Some examples of situations that may increase the likelihood of an entity's (individual, corporation, organisation) systems becoming the target for such attacks include:

- You're behind on system maintenance and as a result you are running buggy versions of critical networked applications and programs – a target for semi-technical 'script kiddies' looking to test their latest downloaded DoS hacker tool;

- You're a high profile corporation and organisation – a target for hackers looking to gain 'kudos';
- You make contentious statements on your web site – a target for offended entities wishing to prove a point;
- You have competitors with a grudge against you;
- You have (ex-) employees with a grudge against you;

Later in the report we describe cases of DoS attacks that have made the headlines.

### **What types of Denial of Service are there?**

For the purposes of discussion, we have attempted to classify the various DoS attacks into those that are active against standalone machines and hosts, and those that are aimed at network-connected hosts. Attacks in the latter category are the most well known and are of increasing concern, especially to businesses conducting e-commerce on the Internet. For these businesses, any server downtime could be costly as a result of lost business opportunities. For high profile companies, network unavailability as a consequence of DoS attacks may also create bad publicity. In this report, we review a number of cases where this has occurred.

This introduction ends with a summary of each of the sections of the report.

### **Physical Attacks against Resources**

Perpetrators of DoS attacks can be both insiders (individuals with inside knowledge of the company/organisation) and outsiders. To counter the internal threat, a company must make sure adequate physical security measures are in place for servers and network components. This section surveys the types of components that need to be secured and how to secure them.

### **Standalone Host DoS attacks**

Such attacks may result in the exhaustion of system resources such as disk space and CPU time. Attacks may be perpetrated by individuals or by programs. Such programs are known as viruses. Possible protection measures against these attacks are also described.

### **Networked Host DoS attacks**

This is the main section of the report and discussion focuses on attacks at the Application, Transport and Network layers. Several well known attacks are described here, including E-mail bombing, Ping of Death, Teardrop and Land, Smurf and Fraggle, together with a discussion of how to protect against these attacks.

### **Information Warfare**

Developed countries are becoming increasingly reliant on a computer-based information and communication infrastructure. Denial of Service attacks against this infrastructure would have disastrous consequences. Therefore Information Warfare is considered an increasing concern by the major military powers. We review the state of play in this area.

## PHYSICAL ATTACKS AGAINST RESOURCES

Perpetrators of DoS attacks can be both insiders (individuals with inside knowledge of the company/organisation) and outsiders. To counter the internal threat, a company must make sure adequate physical security measures are in place for servers and network components. For example, for insiders with physical access to the network and computer systems, pulling the plug on a critical server is a damaging non-technical way to bring down a network as is physically severing the communications link from a company to its ISP (Internet Service Provider).

Some of the critical components of a networked computer system that need to be secured are:

- Application, email and file servers – servers should be fitted with uninterruptible power supplies to prevent damage in the event of power outages; should be located in a temperature and humidity controlled environment.
- Routers and switches, especially if these form part of the network backbone – redundant fault tolerant configurations should be implemented if possible; where the component offers an access control facility, this should be used to restrict update access to internal configuration information.
- Network wiring closets – the closet should be locked; all wiring within the closets should be clearly labelled to ease network maintenance and troubleshooting.
- Internal and external network cabling – alternative methods of communication should be established for critical connections, for example ISDN links in the event of a failure of the main leased line

In addition to the specific safeguards, pervasive security measures need to be in place to support those safeguards. These include:

- Documented and tested IT disaster recovery plans and business contingency plans
- Clearly defined problem escalation procedures and responsibilities
- Coherent logical (system) and physical access control policies
- Adequate back-up and restore procedures for critical business and system configuration data

# STANDALONE HOST ATTACKS

## Destructive and Resource Exhaustion Attacks

Although the most well known attacks are against network services and connectivity, system resources, such as disk space and CPU cycles, are also at risk of being the subject of a Denial of Service. This section surveys the various attacks and prevention measures, using the UNIX operating system as an illustration.

Denial of Service attacks in Unix can be broadly classified into two categories:

- Attacks that attempt to **damage / destroy** resources so that it cannot be used, for example causing a disk crash resulting in system halt or deleting critical commands like **cc(1)** and **ls(1)**.
- Attacks that **overload** system services, deliberately or accidentally, so that the system cannot be used for service for example filling up a disk partition so that users and system programs cannot create new files.

Many denial of service attacks of the second kind are not deliberate. Runaway programs and user errors may cause it. One common cause is typing errors like `X= = 0` instead of `X! =0`

### Attacks to Damage/Destroy Resources

Attacks intending to damage or destroy computing resources are a major cause of concern. But these kinds of attacks are rather easy to prevent by restricting access to system and application critical accounts and files and protecting them from unauthorised users.

### Possible Destruction Attacks and their Prevention

#### *Reformatting a disk partition*

Prevent accessing machine in single user mode. Protect superuser account. Write-protect disks that are read-only.

#### *Deletion of critical files*

Protect system files and accounts by specifying appropriate modes (755 or 711). Protect superuser account.

*(like all the files in /dev or /etc/passwd)*

#### *Shutting off power to the computer*

Secure computer physically. Physically secure circuit breaker boxes

#### *Cutting network or terminal cables*

Physically secure cables by running through conduits etc.

## Overload Attacks

Overload attacks are typically carried out in shared resource or service environment. The shared resource or service is overloaded with requests to such a point that it is unable to satisfy requests from other users.

### **Process Overload Attacks**

This is one of the simplest denial of service attacks, where the user makes the system unusable for other users who are sharing the computer at the same time. Typically, this is a problem in the shared system environment, as an overload in a single user workstation it is only the user who cannot use the resource. But consider a programmer unintentionally running a faulty program that spawns so many processes that the system cannot support and the system crashes.

*Too many processes*

**Fork()** instructions can lead to too many processes.

```
main( )
{
    while (1)
        fork( );
}
```

This program causes execution of **fork()** instruction that initiates a second process similar to the first one. Both these processes execute **fork()** instruction creating four more new processes. This growth continues until the system can support no more processes. This is a total attack since all the child processes wait for the new process to be executed. The current versions of Unix have a limit to the number of processes that can be run by an UID. This is called **MAXUPROC**. This is usually configured into the kernel when the system is built. Some versions of Unix allow this limit to be set at boot time. As a result a user employing this mode of attack will exhaust his/her quota of processes but no more. The superuser can then use the **ps(1)** command to determine the process numbers of the offending processes and kill them by **kill(1)** command. It is not possible to kill the processes one by one as the remaining processes simply create more. So a better approach is to use kill command to stop each process. In spite of **MAXUPROC** the suppressers in modern versions of Unix can still launch such an attack on the system because there is no limit to the number of processes that superuser can spawn.

It is also possible that there are so many users logged on to the system that the system reaches the total number of allowable processes because so many users are logged on although none of them has reached their individual limit. So when an error message from the shell says "no more processes" then either the user has created too many child processes or there are too many processes running on the system that the system cannot support any more process.

```
% ps -aux
no more processes
```

%

If the process problem does not correct itself then it might require rebooting the computer.

## System Overload Attacks

Users spawning large number of processes that consumes large amount of CPU may cause denial of service. Because most versions of Unix implement round-robin scheduling, these overloads reduce the total CPU time available for other users, for instance

- despatching a large number of **find(1)** commands with **grep(1)** components throughout usenet directories may slow down the system drastically.
- Spawning a dozen large **troff** jobs will also have the same effect.

Most these attacks are unintentional, therefore the best way to counter them is to educate the users about using the shared systems fairly. Using *nice* commands to reduce priority of background tasks and doing these few at a time will help avoid such *DoS*. Deferring execution of long tasks when the system is idle is also a way of avoiding such overload situation. Forceful measures have to be adopted with users who intentionally create such overload of system. When the system is crawling, one can log in as **root** and set his own priority as high as one wants using **renice**<sup>1</sup> commands.

```
# renice -19 $$
```

Using **ps** command one can see what is running and then follow with the **kill** command to remove the processes that are burdening the system. **Renice** commands can also be used to slow down these processes.

## Disk Attacks

A way of overwhelming the system is to launch a disk attack. In this attack, a user may fill up the disk partition making it impossible for others to create files or do other important works.

### *Disk full attacks*

Disks may fill up suddenly when an application program or a user by mistake creates too many files or creates few files that are too large. Another reason for disk filling up can be users slowly increasing their disk usage.

To focus on the clean up effort., **du** (1) command can be used. It searches through the tree of directories recursively and finds out which directory contains what amount of data.. It prints how many blocks are used by each one.

To check the whole **/usr** partition one has to type

```
# du /usr
```

Another approach to prevent disk full attack is to check for larger files. This can be done through using **find** command . **find** command along with **-size** option that is

available in Unix helps to find out files larger than a certain size. This way of finding larger files is as fast as doing a **du** search.

```
Example:    # find /usr -size +1500 -exec ls -l {} \ ;
```

The **quot** (1) command – available on some System V and Berkley systems - summarizes file system usage by user. **quot** with **-f** prints the number of files and the number of blocks used by each user.

```
# quot -f /dev/sd0a
```

### *INODE ATTACKS*

Inodes in Unix filesystem are used to store information about files. So, one way of disk attack would be to use up all the available inodes on a disk. As a result no new files can be created. Each new directory, file, FIFO, pipe, or socket requires an inode on disk to describe it. So, exhausting all inodes by, say, creating thousands of empty files - intentionally or accidentally - will result in the system unable to allocate disk space even if there are free disk spaces.

The **df** command with the **-i** option tells how many free inodes are on the disk.

```
% df -I /usr
```

At the time of booting the disk the number of inodes available to a filesystem is decided. This default created at the time of booting can be changed to increase or decrease the number of inodes.

## Prevention Measures

### Using Partitions

The hard disk can be divided into smaller partitions. Different users' home directories in different partitions. The advantage is even if a user fills up one partition, users in other partition are not affected.

### Using Quotas

A better way of protecting disk full attacks is to use quota system available in some unix systems. With disk quotas each user is allotted a limit to how many inodes and how many disk blocks he can use.

There are two types of quotas:

- **hard quotas** are absolute limits on how many inodes and how many disk space an user can use
- **soft quota** are advisory in nature. A user can exceed his quota for a period of three days.

### Reserved Space

Unix versions implementing the BSD Fast File system reserves about 10 per cent of disk and makes it inaccessible to regular users. This is mainly to do with the performance of the system – the BSD fast File system can perform well if less than 10 percent of the disk is free. Consequently, this prevents the ordinary users to completely fill up the disk. This restriction does apply to the superuser a processes run by the superuser.

### Swap Space Attacks

Unix systems are configured such that a part of the disk space can hold process memory images when they are swapped or paged out of the main memory. If the system is not configured with enough swap space, it might happen that new processes will not run although there is enough disk space. The error message is often "no space". When there is no swap space for a new process to run, then one can increase the space allocated to backing store. Usually for this the computer has to be shutdown and repartitioning of the disk has to be done. In case of a malicious process using up the swap space, the process has to be killed.

### Soft Process Limits

Berkley unix has the capacity to set limit to the amount of memory or CPU time that a particular process can consume. Limits can be set for **cputime, file size, data size, stack size, core dump size** and **memory use**. Also limit command can be used to change a limit. Soft process limits prevent accidental denial of service. At the time of program development it ensures that the other users sharing the system are not denied the resources because of some error in the programming logic.

## **Logic Bombs**

"A Logic bomb is that which is only executed when a specific trigger condition is met." [1] Logic bombs usually are embedded within a commonly used piece of software and are usually placed within the program by the software developer.

What triggers a logic bomb can be multiple of actions, the presence or absence of a user, or file, a day and/or time of the week in a particular year, the execution of a application by a certain user. The logic bombs can check which users are logged on before they trigger themselves. When triggered the logic bomb can cause terrible damage by altering or deleting files or crash workstations.

The most common example of a logic bomb is when a bomb is triggered when an employee ID number is missing on the payroll twice because the employee has left the company.

Time-outs are a special kind of logic bomb that forces payment. Time-outs stop a program running unless some special action is taken. (E.g. the SCRIBE text formatting system uses quarterly to require licensees to pay their quarterly license fees)

To protect against logic bombs software has to be tested and read thoroughly, also keep backups to correct any damage caused by the logic bomb.

### **CIH Virus**

The most recent example of a logic bomb is the CIH virus. It infects Windows 95 and 98 systems. It was first detected on June 1998 in Taiwan. The most common version of the virus is CIH 1.2, which triggers on the 26<sup>th</sup> of April 1999. It overwrites the hard disk and the flash BIOS making the computer unusable.

### **Cases of Logic Bombs Attacks**

They have been several cases involving disgruntled employees leaving logic bombs on the company computer systems but have been difficult to prosecute. In London, Ontario, an employee planted a logic bomb that would have destroyed the computer system, but he was not convicted because previous infections had not been prosecuted. In Toronto, a company had a logic bomb triggered the day an employee was dismissed and it did wipe out the computer system. In the UK a James McMahon, a contract systems programmer was accused of planting logic bombs in clients computer systems but was cleared of all charges. When working for Pandair he was accused of planting logic bombs that would lock terminals at the Heaton office and a second that would have wipe the memory of the computers at the Birmingham office. He was found not guilty as the prosecution couldn't authenticate that he had placed the logic bombs. His motive was that he had lost a valuable contract with Pandair.

## **Trojan Horses**

"A Trojan horse is a program with hidden side-effects that are not specified in the program documentation and are not intended by the user executing the program." [1]  
Trojan horses can disguise themselves as anything from spreadsheets, games, an editor (buffer), WWW pages, PostScript files, Perl scripts and shell files. While the program appears to operate correctly it also executes other instructions that are malicious without the user's knowledge. For example while the user may be playing a game the program may be reformatting part of the user's hard disk.

There are different types of Trojan horses, some of the most common are explained below:

### **Remote Access Trojans**

These are the most common types of Trojan horse. When executed on the victim's server the perpetrator is given the victim's IP address and full access controls. Common remote access functions given are keylogging, uploading and downloading files, making a screenshot etc.

Some Trojan horses restart every time Windows is loaded, this is achieved by altering the registry or win.ini file or systems file so that the Trojan is guaranteed to be restarted. Some Trojans create a file in the WINDOWS\SYSTEM directory and look like a normal Windows executable file to the user. To stop a user killing the Trojan horse process the program hides from the Alt+Ctrl+Del menu, they use fake names which makes it hard for the user to tell which process to kill.

The remote access Trojans open a port on the victim's computer letting everyone to connect. Some options available like changing the access port and a password so only the perpetrator can access the infected computer. The change port option is useful, as you don't want the victim to see the same port is open all the time.

### **Password Sending Trojans**

This type of Trojan horse reads any passwords stored in the cache and sends them to the perpetrator via a secret e-mail. Most of these use port 25 to send the e-mail. Some of these Trojans also send other information like ICQ computer number info and so on.

### **Keyloggers**

These Trojans log the keys the user is typing in and then check for passwords in the log file. Most of these Trojans restart every time Windows is loaded. They are able to record on-line or off-line. When on-line the perpetrator knows the victim is on-line and can record everything. When off-line everything the victim types is recorded and saved to the victim's hard drive waiting to be transmitted when the user is on-line.

## **Destructive**

These are a very dangerous kind of Trojan horse that destroy and delete files. They delete mainly all the .dll, .exe or .ini files on your computer. Shar format file was a type of destructive Trojan horse. The shar file unpacked a number of files in the local directory, after a couple of hundred lines the following commands were executed:

```
rm -rf $HOME  
echo Boom!
```

This removes the user's directory. Some users lost most of their file system as shar was executed while the user was in the root directory. To stop this happening, execute the shar file on a second machine or use the chroot() system call in a restricted directory, which prevents the file from accessing files or directories outside the working directory.

Text editors can allow commands to be loaded into their buffers at the start or last few lines of code to allow the editor to initialise itself and execute commands. These few lines can be used cause damage.

## **FTP Trojans**

This Trojan horse opens port 21, which lets FTP clients log on to your computer without a password and allows uploading and downloading files to your computer unchallenged.

Prevent Trojan horses never execute a program until you have read the entire file. Use an editor that displays control codes in a visible manner and never run the program under administrator mode.

# STANDALONE HOST ATTACKS

## Viruses

A virus is a program that copies itself. This can very quickly exhaust a host's system resources. In addition, in the right conditions, a virus can propagate very quickly to all other networked hosts, bringing the whole network to a standstill in a very short space of time. This section discusses the newer macro viruses as well as the traditional operating system viruses.

### What is a Virus?

A virus need do no more than replicate in order to be a virus. Indeed, 95% of viruses do no more than that, plus some trivial extra like beeping the keyboard, or displaying a message. And conversely, if a program does something nasty that you weren't expecting, that doesn't make it a virus, unless it replicates. Such a program is called a 'trojan', after the famous horse of Troy.

A VIRUS is a small, executable program with the ability to replicate itself by adding its code to that of a host program and/or the system area of a hard or floppy disk. The user is generally unaware of the actions of a virus as it replicates and usually only becomes aware of its presence when the virus 'activates', which it does according to a given set of conditions and at which time it is often too late. However, once the user knows what signs to look for, it can be very obvious when viral activity occurs.

### History of Viruses

The term "computer virus" was formally defined by Fred Cohen in 1983, while he performed academic experiments on a Digital Equipment Corporation VAX system. Viruses are classified as being one of two types: research or 'in the wild.' A research virus is one that has been written for research or study purposes and has received almost no distribution to the public. On the other hand, viruses that have been seen with any regularity are termed "in the wild." The first computer viruses were developed in the early 1980s. The first viruses found in the wild were Apple II viruses, such as Elk Cloner, which was reported in 1981 [\[Den90\]](#). Viruses have now been found on the following platforms:

- Apple II
- IBM PC
- Macintosh
- Atari
- Amiga

Note that all viruses found in the wild target personal computers. As of today, the overwhelming number of virus strains are IBM PC viruses. However, as of August 1989, the number of PC, Atari ST, Amiga, and Macintosh viruses were almost

identical (21, 22, 18, and 12 respectively [Den90]). Academic studies have shown that viruses are possible for multi-tasking systems, but they have not yet appeared.

Viruses have ``evolved" over the years due to efforts by their authors to make the code more difficult to detect, disassemble, and eradicate. This evolution has been especially apparent in the IBM PC viruses; since there are more distinct viruses known for the DOS operating system than any other.

The first IBM-PC virus appeared in 1986 [Den90]; this was the *Brain* virus. *Brain* was a boot sector virus and remained resident. In 1987, *Brain* was followed by *Alameda* (Yale), *Cascade*, *Jerusalem*, *Lehigh*, and *Miami* (South African Friday the 13th). These viruses expanded the target executables to include COM and EXE files. *Cascade* was encrypted to deter disassembly and detection. Variable encryption appeared in 1989 with the *1260* virus. Stealth viruses, which employ various techniques to avoid detection, also first appeared in 1989, such as *Zero Bug*, *Dark Avenger* and *Frodo* (4096 or 4K). In 1990, self-modifying viruses, such as *Whale* were introduced. The year 1991 brought the *GPI* virus, which is ``network-sensitive" and attempts to steal Novell NetWare passwords. Since their inception, viruses have become increasingly complex.

Examples from the IBM-PC family of viruses indicate that the most commonly detected viruses vary according to continent, but *Stoned*, *Brain*, *Cascade*, and members of the *Jerusalem* family, have spread widely and continue to appear. This implies that highly survivable viruses tend to be benign, replicate many times before activation, or are somewhat innovative, utilizing some technique never used before in a virus.

Personal computer viruses exploit the lack of effective access controls in these systems. The viruses modify files and even the operating system itself. These are ``legal" actions within the context of the operating system. While more stringent controls are in place on multi-tasking, multi-user operating systems, configuration errors, and security holes (security bugs) make viruses on these systems more than theoretically possible.

This leads to the following initial conclusions:

- Viruses exploit weaknesses in operating system controls and human patterns of system use/misuse.
- Destructive viruses are more likely to be eradicated.
- An innovative virus may have a larger initial window to propagate before it is discovered and the ``average" anti-viral product is modified to detect or eradicate it.

### **Different Kinds of Virus**

We discuss the traditional types of operating system viruses below.

#### **Boot Sector Viruses**

Stoned. These infect the boot sectors of floppy disks, and either the partition sector [Master Boot Record, MBR] or the DOS boot sector [DOS Boot Record, DBR] of hard disks. Here's how a BSV spreads. A floppy disk has just arrived, with some data on it [some word-processed files and a spreadsheet, perhaps]. This is part of a project that you are doing jointly with a colleague. What your colleague doesn't know is that his computer is infected with a BSV, and therefore so is the disk he sent you. You put the disk in drive A and start using these files. So far, the virus hasn't done anything. But when you finish for the day, you switch off the computer and go home. Next day, you come in and switch on. The floppy disk is still in drive A, so the computer tries to boot up from this disk. It loads the first sector into memory and executes it [normally, this is a little program written by Microsoft to load DOS; or if it can't find DOS on the disk, to tell you so - 'Non-System disk, or disk error. Replace and press any key when ready']. Everyone has seen this message numerous times, and so you open the drive door and press a key. But this disk is infected with Stoned, so what executed was not just the program by Microsoft, but the Stoned virus, written in 1987 in New Zealand [and so sometimes called the New Zealand virus]. The virus installs itself on the hard disk, replacing the MBR, and copying the original MBR to a place a little further down the disk.

When you start up from the hard disk, the MBR runs, but this is Stoned virus. Stoned virus goes memory resident, capturing the disk read/write interrupt 13h, and then it loads the original MBR, and the boot-up process continues as normal. But, since the disk read/write interrupt is captured, every time any write or read access [you think you're making a read, but the virus decides to write anyway] is made to drive A, the floppy is examined, and if it is not already infected, Stoned virus is installed on the boot sector. Thus, your computer is now infecting every disk put in drive A, and sooner or later one of these will be sent to a colleague, and the cycle continues.

BSVs infect PCs. They don't care what operating system is running, or what security software is installed, because at the time the BSV installs itself the operating system or security program is not running yet. However, with some non-DOS operating systems for example, Windows NT, or OS/2], although the PC is infected the virus cannot copy itself on to subsequent disks and cannot spread. It can, however, still do damage, as was discovered by one surprised UNIX user when Michelangelo triggered on 6 March. To most people, the fact that viruses can infect in this way comes as a big surprise, which partly accounts for BSVs being so common.

### **TSR File Viruses**

TSR file viruses are no longer common. As the name suggests, these infect files. These are usually COM and EXE, but there are some device driver viruses, and some viruses infect overlay files; executable programs don't always have the extension COM or EXE, although over 99% do.

For a TSR virus to spread, someone has to run an infected program. The virus goes memory resident, and typically looks at each program run thereafter and infects it if it is not already infected. Some viruses are called 'fast infectors', and they infect if you just open the file [for example, a backup might open every file on the drive]. Dark Avenger was the first 'fast infector'. In the case of Green Caterpillar, the infection trigger is anything that determines what files are present [such as DIR].

Other triggers have been used, but the commonest is to infect each program that you are about to run.

### **Non TSR File Viruses**

It is much easier to write a non-TSR virus, and so many of the budding virus authors do so. But it is quite rare for such a virus to be encountered 'in the wild'; less than 1% of reported outbreaks are a non-TSR virus. With such a virus, running an infected program runs the virus, which at that time looks for another file to infect, and infects it. Vienna is the commonest non-TSR virus; Vienna was the first file virus 'in the wild', but now has the status of 'rare'.

There are a lot of viruses based on Vienna, because a disassembly [which is almost equivalent to source code] was published in a book in 1987.

### **Companion Viruses**

If you have a COM file and an EXE file with the same filename and type that name, DOS runs the COM file in preference to the EXE file. Companion viruses use this feature of DOS. Each EXE file that you have acquires a companion COM file with the same name. Then, when you try to run your EXE program, actually the COM program runs, and that is the virus. When the virus has finished doing what it wants to do [such as creating another companion for another file], it then runs the EXE program, so that everything seems to work normally.

There have been a few successful companion viruses, but not many. The main advantage to the virus author is that because the EXE file does not change, some change-detection software might not realise that a virus is spreading.

Another type of companion is the 'path companion'. This sort of virus puts a program in a directory that is earlier in the DOS PATH than is the victim. When you run a program that is not in your current sub-directory, DOS searches for the program in various sub-directories, as specified by the PATH command in your AUTOEXEC.BAT file. Path companions are harder to write than ordinary companions, so there aren't many of them.

### **Overwriting Viruses**

An overwriting virus simply overwrites each file it infects with itself, and the program no longer functions. Because this is so glaringly obvious, overwriting viruses are never successful in spreading.

### **Multipartite Viruses**

Some viruses, such as Tequila, infect multiple objects. When you run a Tequila-infected EXE, Tequila installs itself on the MBR. When you boot up the computer, Tequila runs from the MBR, and goes memory resident. While Tequila is memory resident, it infects EXE files. Other viruses, such as some of the versions of Antidote

infect COM, EXE and MBRs interchangeably. Some viruses infect COM, EXE, MBRs and device drivers.

### **Miscellaneous Objects of Infection**

There is a virus that infects OBJ files. There is a virus [Starship] that infects by creating a new DBR, leaving the old one intact, leaving the code on the MBR intact, and changing the pointer in the MBR so that the Starship DBR is executed before the original DBR. There are other viruses [DIR II and Dir.Byway] that infect file systems by changing the FAT and directories so that files on the hard disk are all cross linked to the virus. There are all sorts of ways of skinning this particular cat.

### **Macro Viruses**

Many applications provide the functionality to create macros. A macro is a series of commands to perform some application-specific task. Macros are designed to make life easier; for example, to perform some everyday tasks like text-formatting or spreadsheet calculations.

Macros can be saved as a series of keystrokes (the application records what keys you press); or they can be written in special macro languages. Modern applications combine both approaches; and their advanced macro languages are as complex as general purpose programming languages. When the macro language allows files to be modified, it becomes possible to create macros which copy themselves from one file to another. Such self-replicating macros are called macro viruses.

Many software packages have a macro language. Perhaps the very first well known and widespread was Lotus 123. It was proved long ago that for Lotus 123 it is possible to write a self-replicating macro (a virus macro), which would be capable of spreading from one file to another. However, viruses have never been a problem for Lotus 123. You would literally see the infection process right on your screen.

Macro viruses have become a major threat to PC users, for several reasons.

- Macros which infect Microsoft Word for Windows or Microsoft Excel for Windows are written in WordBasic. This is easily accessible to many PC users; and macros are much easier to write than executable file viruses
- They are the first viruses to infect data files, rather than executables. Data files, to which macros are attached, provide viruses with a more effective replication method than executable files. Data files are exchanged far more frequently than executable files. The development of macro viruses has taken place parallel with the increased use of e-mail [and the ability to attach files to e-mail], and mass access to the Internet. This makes macro viruses a much greater threat to computer users than executable file viruses and boot sector viruses.
- Macro viruses are not platform-specific. There are versions of Microsoft Word for Windows 3.x, Windows 95/98, Windows NT and Macintosh. This makes all of these operating systems susceptible to macro viruses.

Macro viruses have already had a marked effect. And while most cause no damage to data, we have already seen the first steps towards macro viruses which threaten data. If instructions within a macro make calls to a specific operating system, they will be restricted to that particular operating system. However, the WM.MDMA virus gets round this restriction by including variable payloads for different operating systems.

Perhaps the most well known of the macro viruses is Melissa.

## **The Melissa Macro Virus**

Dateline: 04/02/99

It all started when the poster "Sky Rocket" introduced a file containing a list of 80 pornographic sites in the news group "alt.sex" on March 26, 1999. It had the list of sites and the virus - Melissa.

### **What does it do?**

- Infects Word Documents (Word 97 & above)
- Emails itself using MS Outlook (\*not\* Outlook Express)
- Changes settings to ease infection
- Changes settings to avoid detection

### **How does it work?**

- It infects Word Documents in Word 97 and above versions of Word. It works like this:
  - When the document is opened in Word, it copies itself on NORMAL.DOT file.
  - All other documents use NORMAL.DOT and are automatically infected.
  - The virus code runs, every time you open or close a document as it adds the registry key.  
HKEY\_Current\_User/Software/Microsoft/Office/Melissa?
- It changes settings of Word to disable macro security features.

It launches MS Outlook and sends an email with following message to 50 users from the AddressBook:

### **Who gained?**

Melissa is spreading like a chain reaction all across the Internet. It seems more than likely that it is the result of a well-planned effort for some material benefit. The major gainers due to Melissa are: The 80 pornographic sites included in the list. Their sites have been promoted all throughout the world, for free? There is a parallel advertising in such illegal sites which pays, most times, more than "legal" sites. AntiVirus software manufacturers in terms of [bumper sales and increased stock value](#). Media sites such as [CNET](#), [ZDnet](#), [CNBC](#), etc. in terms of increased hits. [Coastline.com](#) for quickly pouncing upon the opportunity and registering [MelissaVirus.com](#) on March 29, 1999 (3 days after introduction). Virus writers in terms of free "publicity" and feeling of being a "crusader" *against* peace and ease of computing.

## **How Viruses are Spread**

It seems to be a common belief that viruses are spread by games, by shareware or by BBSes. The truth is more complex. First, remember how the most common sort of virus, boot sector viruses, work. A physical floppy disk has to be involved, and there doesn't need to be any software on it. You cannot get a BSV by using a BBS.

The most likely routes by which a virus gets into an organisation are engineers and parents.

1. Hardware engineers visit a large number of computers, and like the busy little bee, could pick up some pollen here, and deposit it there. Hardware engineers should have all their software disks permanently write-protected, but don't. Hardware engineers should frequently check any write-enabled disks for viruses, but don't. Of course, the majority of hardware engineers are clean and well-behaved, but there are a few that need re-education.
2. Parents have children, and if there is a PC at home, and the children are young teens, then they quite possibly swap software at school. The disks that they bring home might well be infected, and if the parent is taking disks to and from work, they could easily take a virus into work with them.
3. A boot sector virus could arrive on a data disk from a colleague.
4. The newer macro viruses can propagate automatically via Internet messaging systems.

You can catch a virus by executing an infected program, whether you realize the program was run or not. This includes overlay files, system drivers, EXE and COM files, etc. You can catch a virus by ATTEMPTING to boot from an infected floppy disk or hard disk, without regard as to whether that attempt was successful. A cold boot will remove a virus from memory, a warm boot won't necessarily do it. So press the button on your computer instead of using CTRL-ALT-DEL. You CAN'T get a virus just from looking at an infected disk or file. You CAN'T get a virus from a data file, unless it is actually an executable and some other program renames it. So in order to keep yourself in the clear, always check any new program for viruses before running it, and never leave a disk in the floppy drive when you boot up?

## **Virus Prevention**

A good set of rules might be as follows.

1. Any incoming floppy disk must be virus-checked.
2. If your anti-virus software finds a virus, tell PC Support.

Notice that the rules are very simple. That way, people are more likely to remember and follow them. The next thing you need is procedures. The procedure tells the users how to obey the rules. The procedure for checking disks should be written down in detail ['Put the floppy disk in the drive, and type . . .']. If you have a 'sheep-dip' computer, put the procedure up on the wall near to it.

Education is also important. You can't just tell grown-ups to do something and expect that they'll obey without question. You have to explain the reason to them. You can do this with talks, or by getting the Dr Solomon's 'Virus Video' and letting them watch it.

You also have to provide tools. You can't detect a virus with your bare hands. Any sensible anti-virus strategy must take account of the fact that even 'well-educated' users are fallible. The foundation of any comprehensive anti-virus strategy, therefore, must be anti-virus tools which will effectively detect, remove and prevent virus infection . . . even when the rules and procedures have not been followed.

Below we describe the main types of anti-virus tools.

### **Scanners**

A scanner is a program that knows how to find a particular repertoire of viruses. Scanners are updated, quarterly or monthly. For many users, quarterly upgrades are sufficient, but every now and then, a new virus comes out and spreads very fast. In that case, you could be unable to detect this 'in the wild' virus for several weeks, depending on where you are in the update cycle. So, many people subscribe to monthly upgrades to avoid this situation.

Scanners can be either on-demand, or on-access. FindVirus is an on-demand scanner, and must be run by the user [although this could be done automatically, at start-up, from the AUTOEXEC.BAT; or using a scheduler].

WinGuard [Windows 3.x, Windows 95/98 and Windows NT] are on-access scanners, and work continuously. As soon as any disk is accessed, it is checked for boot sector viruses; and as soon as any file is used, it is checked for file viruses. Both programs may be [optionally] configured to check files as they are written to the hard disk [useful if files are being downloaded from a remote site, such as a BBS, or the Internet]. WinGuard, which is a Windows-specific program uses zero conventional memory. Any additional time-overhead involved in checking the disk or file is unlikely to be noticeable in most cases. WinGuard, which does not have the constraints of a DOS TSR program, has the same detection capability as FindVirus.

### **Checksummers**

A checksummer is a change detector. Executable files should not change, except for a good reason, such as updating of software. A checksummer aims to detect changes. The advantage of checksummers is that they do not detect a repertoire of viruses, so do not need updating. The downside of checksummers, is that they are more hassle than scanners [files change on your computer more often than you might have thought, for good and valid reasons], and they do not detect all viruses. For example, checksummers do not detect 'slow infectors'; they do not detect all boot sector viruses [if the hard disk code is left unchanged]; and they have problems with stealth viruses. Checksummers can be on-demand [like ViVerify], or on-access.

## **Viruses and Networks**

A network is a group of computers connected together to make it easier to share data. This provides interesting opportunities for viruses, and for dealing with viruses.

There is a common perception that once a virus gets on to a network, somehow it flashes round the whole network very quickly. The truth, of course, is more complex. Firstly, Boot Sector Viruses cannot travel across networks. If several machines on a network are infected, that's because the virus spread via floppy disks in the usual way. Here's how a file virus spreads across a network.

1. User 1 gets his/her computer infected, perhaps by a salesman's demo. disk. The virus goes TSR.
2. User 1 runs other programs on his/her hard disk. They get infected.
3. User 1 runs some programs on the network. They get infected. A network emulates a DOS device; reading and writing to files on the server is done in exactly the same way as locally. The virus doesn't have to behave any differently to infect files on the server.
4. User 2 logs on to the server, and runs an infected file. The virus is now TSR in user 2's machine.
5. User 2 runs several other programs, on the local hard disk, and on the server. Each file becomes infected.
6. User 3, user 4 and user 5 log on and run infected files.

## **Virus Prevention for Networks**

70% of networks use Novell NetWare, so we'll use that as an example, but you can adapt the same principles for other network operating systems.

1. You can make directories read-only. If you make files on the local hard disk read-only, you're wasting your time, because just about every file virus will make them read/write, infect them, and make them read-only again. This is because the user has the privilege to make files read/write on his/her local hard disk. But on a file server, you don't have to give that privilege to the user, and the virus has the same privilege as the user. Indeed, the virus is the user, and can do no more than the user can. There is no magic about viruses; they are subject to the same constraints as any other programs. Unfortunately, some packages can't be run from read-only sub-directories, because they want to write to configuration, or temporary files, in the same directory.
2. You can make programs execute-only. This means that although the directory is read/write, the executables cannot be written to, or even read. They can only be run. Be warned, though, that on Novell NetWare this is a one-way street. Once you've made a file execute-only, you can't go back. All you can do is delete it, even if you are Supervisor. So, make a copy first. Some programs won't run if they are execute-only, because they have overlays that

are concatenated on the end of the EXE file. So if the EXE file can't be read, the overlays can't be loaded.

3. You can make individual files read-only using the DOS attribute, and then deny the user the modify attribute privilege in that directory.

Using a combination of the three techniques above, you should be able to make a large percentage of the files on the server uninfected [indeed, unchangeable without Supervisor intervention]. This stops viruses infecting most of the executables on the server.

Unfortunately, the important files on the server are the data, and you haven't protected those. The user has read/write access to the data [he/she needs it to do their job], and so the virus also has read/write access to the data. Deleting files is the least of our worries. Consider that some virus damage routines consist of altering files. So how do we protect the data on the server? The only answer is to keep viruses off the workstation as well. The next thing you can do, if your server is running Novell NetWare, is run an anti-virus NLM [NetWare Loadable Module] on the server. This can be scheduled to check the files on the server. The use of a server-based on access scanner provides a multi-layered defence against virus infection, checking files as they are passed to and from the workstations. In addition, users can be denied access to the server unless their workstation is protected.

Dr Solomon's Anti-Virus Toolkit for Windows NT offers comprehensive protection for workstations and servers. FindVirus provides on-demand scanning; and a Scheduler, to check the system at pre-defined times. Winguard for Windows NT provides constant background protection, checking files and disks before they are accessed. If your server is LAN Manager or LAN Server, you can run an OS/2 version of the anti-virus program on the server.

# **NETWORKED HOST ATTACKS**

## **Transport and Network Layer**

Flooding attack incidents have been constantly increasing during the last years. This is due to the attackers' growing accessibility to networks, and the growing number of organizations connected to networks. Many companies are vulnerable, because most systems have not implemented spoof prevention filters (due to issues such as equipment age or capability, access control list (ACL) management capabilities, etc.) and therefore there is very little protection currently implemented against attacks.

An attacker relies on anonymity when attacking hosts/networks so that he/she can do it without being identified. Attacks like "Smurf" and "Fraggle" (described later) work only when IP source-address spoofing is possible, because of the reflexive nature of the attack. Without spoofing, they would just be flooding themselves. Other attacks, such as the fragmentation attacks mentioned, simply use spoofing as a way to avoid being identified. Large lists of super-user accounts, as well as user-level accounts, are passed around in order to help deter the identification process when an attack occurs and it can be traced.

Hopping from account to account increases the chance the attacker will not be found due to uncooperative administrators. Internet Relay Chat (IRC) is a tool used by many to pass around exploit information. It's also used by many attackers to "show off" their attacks to their peers. Unfortunately, the wide use of IRC by attackers makes IRC servers, operators, and users a target when the attacker wants revenge. The first targets of the "smurf" program were IRC servers. Providers who do terminate accounts due to abuse are usually targets as well, just like IRC servers who ban users from using the servers.

### **Goal of the attacks**

The goal of a flooding attack is typically to prevent other users from using a network connection, by disabling a host or service. The reasons for this goal vary: Usually the prevention of network, host, or service usability is a result of revenge, or for the "fun" of wrecking a system, or there might be financial motives behind the attack (e.g. disabling a competitor's systems).

### **Types of Attack**

#### **Smurf & Fraggle**

The "Smurf" and "Fraggle" attacks are two of the most severe Denial of Service flooding attacks found today, because they allow a user with relatively low bandwidth to generate a very large amount of bogus traffic towards a remote network. They utilize IP directed broadcasts in combination with echo protocols and spoofed packets in order to generate multiplied traffic streams. There are two victims: The intended victim, who receives a large amount of traffic from intermediate sites, and the intermediate sites, or "bounce sites" used to multiply the traffic streams.

The attacks are similar in nature to traditional ping and UDP flooding, except that ping and UDP flooding require that the perpetrator have more bandwidth than the target he/she is attacking. Smurf and Fraggle allow the multiplication of traffic through the broadcast mechanism and therefore only require that the sites used to multiply the traffic have enough hosts to increase the factor by which the traffic streams are multiplied. The single stream from the perpetrator to the broadcast LAN represents the flow of information from the perpetrator to the broadcast LAN, usually several packets per second of ICMP echo ("Smurf") or UDP echo ("Fraggle") traffic spoofed to look like it is coming from the victim's system.

If the router at the edge of the LAN forwards the broadcast ping to the LAN, each device on the LAN will respond with an echo-reply (ICMP) or will bounce the traffic (UDP), creating a multiplication of the original traffic flow. The traffic is then directed to the victim. There are usually several bounce sites involved, used to increase the factor by which traffic is multiplied. This attack is characterized by many ICMP echo reply packets at the victim's site or many UDP packets involving the diagnostic "echo" port.

Attackers are still using "Smurf" and "Fraggle" widely, due to the powerful nature of the attack. "Fraggle" was released in March 1998, billed as "udpsmurf" as well.

However, the results of the attack have been reduced greatly; what was once an average of 80 Mbps of traffic from a "Smurf" attack is now less than 4 Mbps, still overpowering a T1 or dial-up connection. This reduction is a result of several methods of education: White paper on smurf/fraggle at <http://www.quadrunner.com/~chuegen/smurf.txt> . Attacked NOCs using network flow information to mail the contacts for networks used in the attack (the "bounce sites" as described in the white paper), security advisory from CERT, passed on by NASIRC, CIAC, FedCIRC, mailing lists

## **Land**

The "Land" attack disables many IP stacks or host operating systems by sending a spoofed TCP packet with identical source address/port and destination address/port parameters, where the address is the device's own IP address. This causes many stacks to get very confused, by using the same connection control block for both ends of the connection, crashing many stacks. Again, it requires the ability to send source spoofed packets from the perpetrator's network.

The "Land" attack, as well as vendor information regarding vulnerability, is discussed in CERT advisory CA-97.28, available at [http://www.cert.org/pub/advisories/CA-97.28.Teardrop\\_Land.html](http://www.cert.org/pub/advisories/CA-97.28.Teardrop_Land.html)

## **Teardrop**

The "Teardrop" attack affects mostly Linux and Win95/NT hosts (among others). It sends a 2-fragment IP packet, with one fragment too small. This causes IP stacks to overwrite a large amount of memory and crash. The "Teardrop" attack, as well as vendor information regarding vulnerability, is discussed in CERT advisory number CA-97.28, available at [http://www.cert.org/pub/advisories/CA-97.28.Teardrop\\_Land.html](http://www.cert.org/pub/advisories/CA-97.28.Teardrop_Land.html)

## **Bonk & Boink**

The "Bonk" and "Boink" attacks reverse the "Teardrop" attack in that they set a fragment offset larger than the packet size. These exploits affect Windows machines. "Bonk" attacks only port 53 on these machines, which isn't always open. "Boink" was released in order to send the attack packets to a range of ports, in order to make the attack more usable.

## **NewTear**

The "NewTear" attack affects Windows machines as well. It is simply a modified version of "Teardrop" which changes padding length and increases the UDP header length field to twice the size of the packet.

## **Ping of Death**

The "Ping of Death" attack affects many IP stacks, sending a fragmented packet which, when reassembled, is larger than 65536 bytes. This causes an IP stack not protecting against the attack to overwrite the buffer used to reassemble the packet. Attacked Windows machines normally experience the "blue screen" with error messages in kernel or network drivers. Affected UNIX systems generally experience a kernel panic or no response to IP traffic.

The "Ping of Death" attack, as well as vendor information regarding vulnerability, is discussed in CERT advisory number CA-96.26, available at <http://www.cert.org/pub/advisories/CA-96.26.ping.html>

## **SYN**

When two computers begin to communicate using TCP protocol, they exchange some packets to establish a connection in advance of exchanging service-specific data. This is called the three-way handshake. When host 1 requires a TCP-based service on host 2, the procedure is as follows:

1. Host 1 sends host 2 a SYN packet.
2. Host 2 replies with an ACK/SYN packet
3. Host 1 issues an ACK packet

When host 2 sends an ACK/SYN packet, it stores in its memory this connection request as a pending connection. This is called half-open connection. When host 2 receives an ACK packet from host 1, host 2 deletes the connection related data from the half-open connections data structure. If host 2 does not receive an ACK packet, it deletes the connection related data after a certain time. However, the size of the buffer keeping the half-open connections is limited. If it fills up, then host 2 cannot accept any new connections. This is what we call the TCP SYN flooding, and it works by exploiting a design flaw in some BSD-based TCP/IP stacks.

Here it must be noted that TCP SYN packets are sent with spoofed source addresses. Since the attack involves spoofed addresses, the connections are bogus. Many times, the spoofed source addresses do not represent actual hosts, and the server waits for a lengthy amount of time (usually 3-6 minutes) for the connection to be confirmed.

Every computer providing TCP-based service potentially becomes a victim of this attack. Because this attack prevent the computer from accepting TCP connections, all TCP-based services such as email and/or web cannot be serviced. There is no complete solution for this problem. However there are some ways to reduce this problem.

In order to reduce IP-spoofed packets, the router should be configured as follows:

1. If an inbound packet has a source address of the internal network, discard the packet
2. If an outbound packet does not have a source address of the internal network, discard the packet

Another way is to allocate more memory for the half-open connections data structure. Nonetheless the half-open connections data structure (buffer) is still limited. Furthermore, shortening the timeout of the half-open connections can be a countermeasure. However, this may reject legitimate connection requests from distant computers.

## **Application Layer**

A Denial Of Service attack, usually referred to as DoS is an activity started by someone with the intent to render a chosen system unavailable. There are many ways of doing this, the more common ones are: flooding, making a machine perform a huge amount of calculations, causing the server application or the entire server to crash.

Denial of Service attacks can be performed on all layers of the TCP/IP reference model. This document shall discuss those relating mainly to the application layer.

The DoS threat is not a new one, it had been known for many years. Some examples are viruses and mail bombs. Those might have taken longer to spread in the pre-Internet era but consisted a major threat. For example a common attack on the Fido-net (amateur net to send e-mails around the world, which worked mainly off-line) was to send a mail to a gateway that consisted of a large zip-file. The gateway would take the zip-file and unpack it to check it for viruses. However if the compressed file inside the zip-file were too big, the gateway's system would not be able to unpack the entire file. Often the decompression program crashed without removing the data it had already unpacked and hence rendered the gateway unavailable for receiving any new data.

With the growth of the Internet the amount of DoS attacks has increased. There are few main reasons:

- The number of potential victims
  - The amount of applications available
  - The complexity of protocols used
  - The increase of functionality in programs (the bigger a program, the more potential bugs you will have)
- The DoS tools are readily available

- You do not need to know what you are doing, just use a tool
- Most people are easy victims (unawareness of what's going on in their computer)

The motivation for starting DoS attacks may differ, but the result is usually the same. Legitimate users cannot make use of a service on the attacked system. Teenagers and internet newbies probably start most DoS attacks. They get hold of an exploit and want to show their friends what they are able to do. A good example of this would be teardrop and winnuke, which was mainly used by people who did not even know why they could crash other people's computers.

The effects of DoS attacks may vary, from just being annoying to threaten your existence. Suppose you are surfing the web and someone manages to crash your computer remotely, usually all you have to do is reboot your system and dial-up again. Probably you will get a new IP address and the attacker will not bother crashing your system again. However if you are running an Internet-Bank and your system get crashed every time you get it up again for a couple of days, you might loose a lot of business or even all your customers. If you are providing "useful" information on the web and someone hacks into your site and changes the content, you loose integrity and availability on the information you usually provide. If an army looses availability, be-it complete loss of data or huge delays, of their communications people's lives might be at stake.

Web applications are probably under the most often mentioned targets for DoS attacks, immediately after operating systems if you talk about DoS attacks over the Internet. One of the biggest problems is the fact that DoS vulnerability may exist on the client side as well as on the server side. Especially since the introduction of Java, Java-script and ActiveX, the amount of news about DoS vulnerabilities in Web-Browsers has increased drastically. In a browser Java-applets are supposed to run inside a sandbox, which should prevent them from accessing any vital resources on your system. So many people think Java in browsers is secure, however an applet which start a huge amount of new browsers, will render your system pretty unusable and might use up all your memory and other system resources until your system crashes. ActiveX can be even worse, some people say executing ActiveX applications is like playing at Russian roulette. ActiveX does not know any kind of system security; the system only distinguishes between trusted and un-trusted sources. This means once an ActiveX control has been authenticated it got complete control of your system. The authentication will not reveal anything about the dangers of the application. Sometimes even simple HTML can crash Web browsers, for example:

```
<html>
<body>
<form method="POST">
<table>
<tr>
<td width="20%">
<input type="text" name="State" size="99999999"
maxlength="99999999" value="">
</td>
</tr>
</table>
</form>
</body>
```

</html>

will crash Internet Explorer (last tried on version 5.0).

On the server side, Web servers are very vulnerable as well to DoS attacks. For example there is no week passing without new bugs being found in Microsoft's IIS server. For example, a bug that has been known for quite some time now is one involving a problem with SSL, but there is no real fix. For example if you request webpages from an IIS server and instead of using 'http' you use 'https' for requests, the server will encrypt all content. If you then request massive documents the CPU resources might go down to a point where the server gets unusable for some time or even crashes. Unless you are using two different servers, there is no fix for this problem.

Another example of a bug in a Web server is [Microsoft FrontPage Sever Extensions for FrontPage PWS 3.0.2926](#). If you type GET with a URL which is more than 166 characters, the program will cause a buffer overflow and the system will crash. This bug had been published on Security Focus on August 08, 1999 and updated August 29, 1999, but until today the problem has not been fixed.

Another application that is often the target of attacks is the mail-server on a system. The size and complexity of some mail-servers has long passed those of the operating system's kernel. Some mail-servers need as much tuning the operating system itself. A few months ago everybody was talking about security holes in sendmail (a mail-server daemon found on unix and unix-like operating systems). It now seems that most of the holes have been closed. However sendmail is not the only vulnerable mail-server. For example the Internet Anywhere Mail Server (version 2.3.1, 3.1) contains two major bugs that will make it crash. If a remote pop3 user issues some commands with more than 200 character arguments or if the user types characters where the server expects numbers then the system will crash.

Email Bombs are one of the oldest types of Denial of Service attack.

An individual can intentionally or inadvertently overwhelm your mail server by flooding it with messages. This type of act is called a denial of service attack. If a denial of service attack is perpetrated against your mail server, either there may be a substantial impact to the throughput of your mail server or your mail server will become overloaded and non-functional.

Tools are available on the market with features enabling you to minimise the possibility of a denial of service attack and well as help you control spam (unwanted or unsolicited email), for example:

- Email access restrictions - Enables you to specify which incoming messages are accepted or denied based on recipient or originating domain, client IP address, server IP address, or originating email address.
- Message size limits - Enables you impose a limit at which a message is deemed too large and rejected by a channel.

Yet another standard server-service that has recently been in the news due to DoS vulnerabilities are the FTP servers. All of the bugs found in the FTP servers were

buffer-overruns. Example of this was the QVC's QVT FTP as well as Gene6's G6 FTP Server. If your username and password are more than 2000 characters the system will log you off, and on a re-login attempt the system will crash. Another problem of this kind was found in the WU-FTP daemon and its derivatives, which runs on many Unix and Unix-like systems. If a user types in a path, which was 1 longer than the maximum number of characters allowed for a path, it would make the service crash.

Not only typical internet applications are vulnerable to DoS attacks, any computer program that is running on the network can be the victim of a DoS attack. For example many Web sites are using SQL servers for storing databases. These are a common target for DoS attacks, for example the current version of Microsoft SQL server 7.0 can be crashed by sending it a TCP packet with three or more NULL bytes in it.

This is only an extremely small selection of DoS attacks possible on the application layer. As I mentioned before the number of potential victims is increasing day by day. Each time a new user subscribes to an ISP a new potential victim is added to the list. The majority of people joining the internet only now are people who have not yet had a lot of experience with computers. This leads to two major problems, these people want to know as little as possible about what is going on in their computer while they are connected, as they would not understand it, but on the other hand they want easy to use and multifunctional programs and often programs which automate everything so that the human interaction shrinks to a minimum. These programs however have the problem of being very large. The problem with large programs is that they can contain a lot of bugs that may be exploited to start a DoS attack. The best example for this is web-browsers, before there was Java, Java-script, ActiveX one heard very little about DoS attacks on browsers. But since their introduction reports about security problems with browsers do not seem to stop anymore. A lot of people writing programs for the internet nowadays are coming from a background where they had little or nothing to do with networks. This means they often have very little knowledge about the protocols they are using and so might introduce vulnerabilities to the applications.

## INFORMATION WARFARE

Information warfare is a specific type of attack, usually resulting in some form of denial of service. The threat from this scene has increased dramatically in the late 1990s, to such an extent that major military powers are having to investigate the potential threat. Since the world has been globalised, it has been easier for hackers to violate secure computer systems. As was shown in Desert Storm, the traditional war is dead; where large numbers of ground based troops battle it out on a huge field. These days, most developed countries rely on a computer based information infrastructure. If it is possible to disable this system, the country would descend into anarchy.

There are several definitions for information warfare. The sources of these definitions can be found in the references at the end of this section.

From NATO<sup>1</sup>:

6. The United States Joint Chiefs of Staff have defined information warfare as:

*"Actions taken to achieve information superiority by affecting adversary information, information-based processes, information systems and computer-based networks while defending one's own information, information-based processes, information systems and computer-based networks."*

A report by the German Bundestag describes it as:

*"The comprehensive use of information and communication technology as well as technologies for the disturbance and destruction of hostile information and communications systems (IaC systems) in crisis and conflicts, in order to gain strategic and tactical superiority."*

And the *Economist* has described information warfare in more accessible terms by saying that :

*"Information warfare could mean disabling an enemy by wrecking his computing, financial, telecommunication or traffic control systems. The relevant weapons might be computer viruses, electromagnetic impulses, microwave beams, well-placed bombs or anything that can smash a satellite."*

From Reliable Software<sup>2</sup>:

According to [DoD96], *information warfare* is defined as "actions taken to achieve information superiority by affecting adversary information, information-based processes, information systems, and computer-based networks while defending one's own information, information-based processes, information systems, and computer-based networks."

So who performs these attacks? There are a number of identified sources of this type of threat. NATO has done a study that shows that the anticipated threats to national security to the US won't be realised for several years:

Examples:

Potential threats:

From the Office of International Criminal Justice<sup>3</sup>:

- A CyberTerrorist will remotely access the processing control systems of a cereal manufacturer, change the levels of iron supplement, and sicken and kill the children of a nation enjoying their food. That CyberTerrorist will then perform similar remote alterations at a processor of infant formula. The key: the CyberTerrorist does not have to be at the factory to execute these acts.
- A CyberTerrorist will place a number of computerized bombs around a city, all simultaneously transmitting unique numeric patterns, each bomb receiving each other's pattern. If bomb one stops transmitting, all the bombs detonate simultaneously. The keys: 1) the CyberTerrorist does not have to be strapped to any of these bombs; 2) no large truck is required; 3) the number of bombs and urban dispersion are extensive; 4) the encrypted patterns cannot be predicted and matched through alternate transmission; and 5) the number of bombs prevents disarming them all simultaneously. The bombs *will* detonate.
- A CyberTerrorist will disrupt the banks, the international financial transactions, the stock exchanges. The key: the people of a country will lose all confidence in the economic system. Would a CyberTerrorist attempt to gain entry to the Federal Reserve building or equivalent? Unlikely, since arrest would be immediate. Furthermore, a large truck pulling along side the building would be noticed. However, in the case of the CyberTerrorist, the perpetrator is sitting on another continent while a nation's economic systems grind to a halt. Destabilization *will* be achieved.
- A CyberTerrorist will attack the next generation of air traffic control systems, and collide two large civilian aircraft. This is a realistic scenario, since the CyberTerrorist will also crack the aircraft's in-cockpit sensors. Much of the same can be done to the rail lines.
- A CyberTerrorist will remotely alter the formulas of medication at pharmaceutical manufacturers. The potential loss of life is unfathomable.
- The CyberTerrorist may then decide to remotely change the pressure in the gas lines, causing a valve failure, and a block of a sleepy suburb detonates and burns. Likewise, the electrical grid is becoming steadily more vulnerable.

Actual Events:

From C-Net<sup>4</sup>:

### **Satellite seizure, blackmail reported**

By [Reuters](#)

Special to CNET News.com

February 28, 1999, 7:35 a.m. PT

### **LONDON--Hackers have reportedly seized control of one of Britain's military communication satellites and issued blackmail threats.**

The *Sunday Business* newspaper, quoting security sources, reported

satellites used by defense planners and military forces around the world.

The sources said the satellite's course was changed just over two weeks ago. The hackers then issued a blackmail threat, demanding money to stop interfering with the satellite, according to the report.

"This is a nightmare scenario," one intelligence source said. Military strategists said that, if Britain were to come under nuclear attack, an aggressor would first interfere with military communications systems.

"This is not just a case of computer nerds mucking about. This is very, very serious, and the blackmail threat has made it even more serious," a security source said.

Police said they would not comment because the investigation was at too sensitive a stage. The Ministry of Defense made no comment.

And another from C-Net<sup>5</sup>:

### **Mexican hackers speak out**

By [Reuters](#)

Special to CNET News.com

August 5, 1998, 11:15 a.m. PT

MEXICO CITY--They have [plastered](#) the face of revolutionary hero Emiliano Zapata on the Finance Ministry's Web site and claim to have monitored visits by senators to X-rated Internet salons.

They also have [vowed](#) to mine official data bases for incriminating numbers and publicize government bank accounts, cellular phone conversations, and email addresses.

A small group of computer hackers has declared electronic war on the Mexican state. So far the cyberpirates appear to be more a nuisance than a serious threat, but they are serving as a wake-up call for computer security in Mexico, experts said.

"We protest with the weapons we have and those weapons are computers," said one of the hackers, who calls himself as LoTek, in an online interview with Reuters.

The hackers, who say they are a trio of Mexicans, surfaced in February when visitors to the Finance Ministry's official Web site were surprised to find Zapata staring back at them.

Initially the prank was thought to be the work of the Zapatista National Liberation Army, named after the leader of Mexico's 1910 revolution, which led an armed uprising in 1994 in the southern state of Chiapas. The Zapatistas run their own Web site and are known to be Net-savvy.

But the hackers, calling themselves "X-Ploit," said they have no links to the Zapatistas, they just share their disdain for the government of President Ernesto Zedillo.

"We're only looking to show that we don't agree [with the government]," LoTek said. "In other places these protests are not heard, but a hacked Web site is read by millions."

The group, which also slapped its smiley-face logo on the Health Ministry's Web site, said it was monitoring senators' online activities and email. "We've been able to capture some of the [Web] visiting habits of some senators, including triple-X pages," LoTek said.

While they have gained a certain online notoriety, federal attorney general's office and Interior Ministry spokeswomen said they knew of no official investigation into the group.

The hackers appear more interested in propaganda than sabotage or espionage, but they do represent a potential threat to government and corporate computer systems, industry executives said.

"The danger is if they can do that, of course they can hack and get into very important data and erase it or put it to their own use," said Luis Loranca, Mexico country manager for [Network Associates](#), a Santa Clara, California-based maker of antivirus and computer network security products.

But he said that with relatively few hackers in Mexico, the most common threat comes from in-house.

Few companies here have adopted internal computer network security measures common in the United States to limit access to sensitive databases and files, Loranca said. This has made corporate espionage relatively easy, although few companies have publicly acknowledged security breaches.

"Espionage is very, very large in Mexico," Loranca said.

Javier Matuk, chief executive of Mexican online service provider [Spin Internet](#), said his firm had not been hacked in its eight years of operation but it constantly monitors its network for intruders. He said he was not too concerned about "X-ploit" because they seemed most interested in being a thorn in the side of the government.

"If what they claim is true, then definitely the government has a problem. But just the government. The rest of us don't have a problem," Matuk said. But he said he thought it was a bit "infantile" to expect that sensitive government information could be gleaned by monitoring senators' email.

Nevertheless, the Mexican hackers said they had their "sniffers" in place at several ministries. "We'll let you know when we've trapped something good," LoTek said.

From ConSeal<sup>6</sup>:

### **Get ready for world war three**

says Tim Phillips

'World war three has already begun,' Robert Steele is telling InfoWarCon, a three-day seminar attended in equal parts by business people, academics and the military, and held to discuss the threat of high-tech warfare. Steele, a former US Marine and CIA employee, is now considered to be an authority on the use of computers to attack individuals or organisations. 'In Europe the understanding of the public is zero, and the understanding of the government is zero.' Steele is describing the new battlefield (computer networks) and the new soldiers (mercenary computer experts). If anyone wants to exploit the lack of security, he says, they don't have to try very hard. Often, they won't even break the law. 'Consistently in Europe the law is 50 years behind the technology,' he warns. 'World war three is a war between governments and gangs.' So-called 'secure' information systems are vulnerable to attack from terrorists and hostile regimes, says the experts. The attacks, they contend, have already begun: the only reason we haven't heard much about them is because either the victims don't know they have been targeted yet, or they would rather keep it a secret.

## CASE STUDIES

In 1996 Zetnet was one of the first to experience a denial of service attack in the UK. The internet service provider was hit by a denial of service attack that halted internet access to its subscribers for over seven hours. It appeared that an argument between one of Zetnets customers and the user of another system on IRC led to an unidentified hacker bringing down the whole of Zetnet in revenge. The outage was caused by continuous streams of very large data packets (up to three times Zetnets bandwidth) being sent to one of the ISPs dial up connections, which resulted in 'normal' traffic being bullied out of the way.

This is not the only example over the years there have been a horrendous number of denial of service attacks. However usually companies do not wish these to become common knowledge due to the repercussions associated with them. Moreover a number of sites are attacked again and again a common example is that of government sites. In an article on Technology News early last year it was reported that:

*"a widespread attack on servers connected to the internet including several US government and university servers, was launched by unknown and virtually untraceable computer criminals. The attacks were launched using software tools that automate denial of service attack, the identity of the actual tool used is unknown but appears to be modified version of tool similar to other tools called new tear, bonk and boink."*

Vnunet also in March of last year reported that:

*"system administrators were alarmed at the ease with which sites at nine of NASA's 10 major field offices and MIT were brought down."*

A number of security intrusions have also occurred this year when Russian and other Serbian sympathisers "ping" attacked NATO Web servers and web-sites in NATO countries including the US, using virus-infected e-mail.

Outlined below are further examples of denial of service attacks.

Melissa Virus:

*Lucent Technologies a telecommunications company, in the middle of March this year had no e-mail communication with the outside world for a couple of days as a result of the Melissa virus attack, forcing the company to shutdown e-mail systems. Michael Vatis director of the FBI's national infrastructure protection centre was quoted as saying that: "damages in the first half of 1999 from viruses alone topped \$7 billion."* Technology News.

Smurf Attack:

In early February a Canadian teenager unleashed a ping flood attack that brought down an ISP. Administrators of Sympatico, the ISP in Nova Scotia realised that they were under a "Smurf" attack when they experienced a series of slowdowns, each lasting for up to a week. The teen targeted the Sympatico service after it caught him in some of his hacking activities and tried to shut him down. The teenager retaliated with the Smurf flood, which took the whole system down for one day at one point.

#### Spamming:

Netcom subscribers in 1997 lost the ability to receive email from people with Hotmail accounts due to a Spam attack. Netcom had to temporarily block mail coming from Hotmail accounts in response to a 500 copy spam with a forged header that hit Netcom servers.

#### Employee Grudge:

A former employee of Intel, KenHamadi repeatedly spammed Intel employees by sending them email promoting his anti-Intel site.

More recently however, towards the end of September US networks came under assault by a fundamentally new style of computer attack known as distributed co-ordinated attacks. These attacks use hundreds of or thousands of servers co-opted by a malicious programmer to tag-team a single server. Because so many servers are being used each attack can be camouflaged as a legitimate connection attempt, making it difficult for the victims intrusion software to identify that it is under attack and impossible to identify who is attacking.

It is interesting to note that in 1997 the annual survey of 520 companies, government agencies and universities found that 25% were subject to denial of service attacks. According to information weeks global information security survey conducted with PWC which is based on responses from 2,700 executives, security professionals and technology managers from 49 countries. The survey found that globally 64% of companies were hit by at least one virus in the past 12 months, up from 53% from the year before, and that denial of service attacks had declined from 13% to 11%.

## SUMMARY

Denial of Service can be the result of accidental threats. These are difficult to specifically protect against; the best that can be done is to establish a good baseline level of IT security.

Denial of Service can be authorised in some circumstances, for example, a system administrator may be charged with the responsibility of revoking the access of any user who has breached company security policy. Models of Denial of Service exist and these can formalise the conditions in which authorised Denials of Service can occur.

The most well known aspect of Denial of Service is the deliberate-unauthorised type - the Denial of Service attack. Attacks can be aimed at system resources, such as disk space and CPU time, as well as network-based resources. Agents of attacks range from human users to mobile code to virus programs. Network based attacks can target any layer of the OSI seven layer model, but are most commonly aimed at the Application, Transport, Network and Physical layers.

No protective technique covers all types of attacks. To understand the safeguards that are required, we must first understand the vulnerabilities that attacks exploit. Most attacks take advantage of one of three things:

- Incorrectly configured programs and application software
- Bugs in programs and application software
- Weaknesses in the network protocols

The first two types of vulnerability are application-specific. Once identified, such a vulnerability is relatively easier to eliminate, either by changing configuration parameters to 'safe' values or installing the required bug patches.

The third type of vulnerability is the most difficult to protect against. Various prevention measures are suggested by CERT and these include implementing router filters to reject IP packets with certain attributes, hosting a minimal set of network services and reviewing physical security over critical computer equipment. More details can be found at [http://www.cert.org/tech\\_tips/denial\\_of\\_service.html](http://www.cert.org/tech_tips/denial_of_service.html).

Although there are many ways in which Denial of Service can occur, the result is the same: legitimate users are prevented from using a resource when it is needed. The growth of the Internet and e-commerce has led to an increase in Denial of Service attacks and the stakes are now raised even higher. Web server downtime may result in potential lost custom. Web site attacks may threaten the trust and good reputation attributed to high profile companies. Resource availability may even result in loss of life.

Looking to the future, an emerging threat is Information Warfare. Information Warfare uses electronic means to disable an 'enemy', usually resulting in some form of Denial of Service. The major military powers consider this threat to be an increasing concern in respect of national security, since these days, most countries are highly reliant on a computer-based information and communications infrastructure.

## **BIBLIOGRAPHY**

### INTRODUCTION

Amoroso, E., 1994, Fundamentals of Computer Security Technology

Beckman, M., 1998, "Prevent Network Denial of Service Attacks"  
<http://204.56.55.10/issues/Jul98/beckman/Beckman.htm>

Chowdhry, P., 1999, "Attacked and hacked!"  
<http://www.zdnet.co.za/pcweek/stories/news/0,4153,2350743,00.html>

Downey, J., 1998, "Denial of Service Attacks"  
<http://www.zdnet.co.za/pcmag/pctech/content/17/08/nt1708.001.html>

Gollman, D., 1999, Computer Security

Hautio, J., Weckstrom, T., 1999, "Denial of Service Attacks"  
[http://www.hut.fi/u/tweckstr/hakkeri/DoS\\_paper.html](http://www.hut.fi/u/tweckstr/hakkeri/DoS_paper.html)

Spafford, E., 1997, Practical Unix and Internet Security

### STANDALONE HOST ATTACKS – GENERAL

Gollmann, D., 1999, Computer Security

Spafford, E., 1997, Practical Unix and Internet Security

<http://newdata.box.sk/manic/different.txt>

[www.datafellows.com/cih](http://www.datafellows.com/cih)

The Risks Digest Volume 5 Issue 63.htm

The Risks Digest Volume 6 Issue 8.htm

### STANDALONE HOST ATTACKS – VIRUSES

"Dr SOLOMON'S All about viruses"  
<http://www.drsolomon.com/vircen/vanalyse/va002.cfm>

"How boot sector viruses infect and spread"  
<http://www.drsolomon.com/products/avtk/tnotes/va003.cfm>

"How file viruses infect and spread"  
<http://www.drsolomon.com/products/avtk/tnotes/tn016.cfm>

"Introduction to macro viruses"  
<http://www.drsolomon.com/vircen/vanalyse/va001.cfm>

"Anti-virus toolkit reference guide"

[http:// www.drsolomon.com/vircen/vanalyse/avtkrefgd.cfm](http://www.drsolomon.com/vircen/vanalyse/avtkrefgd.cfm)

"How to remove a virus using Dr Solomon's Anti-virus toolkit"

[http:// www.drsolomon.com/product/avtk/tnotes/tn008.cfm](http://www.drsolomon.com/product/avtk/tnotes/tn008.cfm)

Fastlane Technology

[http:// www.fastlanetech.com/worm\\_killer\\_reg.htm](http://www.fastlanetech.com/worm_killer_reg.htm)

## NETWORKED HOST ATTACKS – TRANSPORT AND NETWORK LAYER

What is a DoS attack: <http://whatis.com/denialof.htm>

NetworkICE Corporation, Intrusions:

<http://www.8lgm.org/advice/Underground/Hacking/Methods/Technical/Spoofing/Intrusions/default.htm>

Defining Strategies to Protect Against TCP SYN Denial of Service Attacks:

<http://www.cisco.com/warp/public/707/4.html>

Tanenbaum, A.S., 1996, Computer Networks, NJ: Prentice Hall,

CERT/CC, 1998, CERT Advisory CA-96.21 [internet], CERT/CC: PA :

[http://www.cert.org/sdvisories/CA-96.21.tcp\\_syn\\_flooding.html](http://www.cert.org/sdvisories/CA-96.21.tcp_syn_flooding.html)

## NETWORKED HOST ATTACKS – APPLICATION LAYER

[www.cert.org](http://www.cert.org)

[www.chip.de](http://www.chip.de)

[Xforce.iss.net](http://Xforce.iss.net)

[www.hack-net.com](http://www.hack-net.com)

[www.securityfocus.com](http://www.securityfocus.com)

## INFORMATION WARFARE

Committees of the North Atlantic Assembly

<http://www.nato.int/related/naa/docu/1997/ap110stc.htm#II.%20INFORMATION%20WARFARE>

Reliable Software Technologies

[http://www.rstcorp/definitions/information\\_warfare.html](http://www.rstcorp/definitions/information_warfare.html)

Office of International Criminal Justice

<http://oicj.acsp.uic.edu/spearmint/public/pubs/cjarrago/terror02.cfm>

C-Net News

<http://news.cnet.com/news/0-1004-200-339293.html>

C-Net News

<http://news.cnet.com/news/0-1005-200-331929.html>

ConSeal

[http://www.infowar.com/class\\_2/class2\\_q.html-ssi](http://www.infowar.com/class_2/class2_q.html-ssi)

## **Appendix**

### Networked Host Attacks

#### **TCP & ICMP:**

TCP is responsible for breaking up messages into datagrams, and reassembling them properly. However in many applications, we have messages that will always fit in a single datagram. An example is name lookup. When a user attempts to make a connection to another system, he will generally specify the system by name, rather than Internet address. His system has to translate that name to an address before it can do anything. Generally, only a few systems have the database used to translate names to addresses. So the user's system will want to send a query to one of the systems that has the database. This query is going to be very short. It will certainly fit in one datagram. So will the answer. Thus it seems silly to use TCP. Of course TCP does more than just break things up into datagrams. It also makes sure that the data arrives, re-sending datagrams where necessary. But for a question that fits in a single datagram, we don't need all the complexity of TCP to do this. If we don't get an answer after a few seconds, we can just ask again. For applications like this, there are alternatives to TCP.

An alternative protocol is ICMP ("Internet control message protocol"). ICMP is used for error messages, and other messages intended for the TCP/IP software itself, rather than any particular user program. For example, if you attempt to connect to a host, your system may get back an ICMP message saying "host unreachable". ICMP can also be used to find out some information about the network. RFC 792 describes ICMP in detail. ICMP is similar to UDP, in that it handles messages that fit in one datagram. However it is even simpler than UDP. It doesn't even have port numbers in its header. Since all ICMP messages are interpreted by the network software itself, no port numbers are needed to say where a ICMP message is supposed to go.

#### **SPOOFING**

Spoofing is the creation of TCP/IP packets using somebody else's IP address. Routers use the "destination IP" address in order to forward packets through the Internet, but ignore the "source IP" address. That address is only used by the destination machine when it responds back to the source. A common misconception is that "IP spoofing" can be used to hide your IP address while surfing the Internet, chatting on-line, sending e-mail, and so forth. This is generally not true. Forging the source IP address causes the responses to be misdirected, meaning you cannot create a normal network connection. However, IP spoofing is an integral part of many network attacks that do not need to see responses (blind spoofing).