Computer viruses and electronic mail

Francesco Gennai, Marina Buzzi

CNR, Istituto di Informatica e Telematica, via Moruzzi 1,
56010 Pisa, Italy
{Francesco.Gennai, Marina.Buzzi}@iit.cnr.it
http://www.iit.cnr.it/

Abstract. Today the Internet is a valuable source of information as well as a powerful communication medium, with undoubted social and economic benefits, however it also poses some security risks. Viruses may hide in email attachments or in apparently innocent applications directly downloadable from the Internet. In this work we give a brief overview of virus types and main defense techniques. Then we present statistical data of virus attacks revealed by an anti-virus SW activated on our e-mail server, and discuss results in terms of virus types and temporal distribution.

1 Introduction

Network communications represent an easy means for the spread of viruses. Internet users are constantly threatened by the spread of new viruses hidden in appealing objects such as jokes, games, chats, and e-mails ostensibly sent by friends. Although Internet services such as e-mail and www represent the main "open doors", floppy and CD disks are still minor "contributors".

The damage provoked by infections can be very costly for an organization's time and resources and becomes critical when it affects sensitive systems and data. A basic rule in computer security is to make frequent backups to avoid any kind of data destruction or corruption.

Various countermeasures can be applied to prevent infections, such as activating an anti-virus SW, or setting filter rules on the e-mail server in order to discard dangerous files present in the message. Another important factor is awareness of the problem: users must become familiar with both risks and elementary defense techniques (i.e. do not open executable files, disable automated macro activation, etc.).

In this work we present data statistics of virus attacks revealed by an anti-virus SW activated on our e-mail server and discuss results in terms of virus types and temporal distribution.

The paper is organized into two parts. The first part includes a brief overview of virus types and main defense techniques; the second shows the results of our experimentation.

2 Virus overview

The term "computer viruses" refers to programs designed to spread themselves by infecting executable files, system areas, hard or floppy disks, etc. without the user's knowledge. In addition to this self-replication activity viruses can perform 'bad' actions (payloads) ranging from mild disturbance (annoying the user with silly messages) to provoking damage or outright disaster (e. g. deleting/corrupting files or performing a Denial of Service (DoS) attack).

A computer is infected when a copy of the virus resides in the machine. Once the virus is loaded into the memory it can run in background and start to replicate itself. If the infected computer is on the network the infection will propagate very quickly to other machines. This process can be interrupted only by detection and elimination of the virus.

The best-known virus types include:

- **Parasitic**. Parasitic viruses (or file viruses) are code fragments that reproduce by attaching themselves to executable files. When the user starts the infected program, the virus is launched first and then, in order to hide its presence, it triggers the original program to be opened. The parasitic virus acquires the same rights as the original program and thus is able to self-replicate, and to release its payload.
- **Worm**. An Internet worm, unlike a virus, does not infect other program files but uses computer networks and takes advantages of SW bugs to replicate itself. A worm scans the network for another system having one specific security hole (such as buffer overflow), and copies itself into the new machine (via smtp, ftp, http, Internet Relay Chat, etc.) and then continues the self-replication process. Thus it has the ability to self-replicate incredibly quickly. Furthermore, the worm can release a payload such as scheduling a Distributed Denial of Service (DDoS) attack toward a target system or network.
- **Trojan Horses**. A Trojan horse is a program which hides malicious code disguised in appealing shapes i.e. it claims to do something "cool" or useful while actually provoking damages. Trojan horses are not designed to replicate automatically.
- **Macro**. Macro virus is one of the most common types of viruses for various reasons: it requires little skill to write; many common applications (word processing, spreadsheet, presentation) make use of macros; and lastly, documents (such as reports, slides, etc.) are frequently exchanged between users. Macro viruses are self-replicating macros that can become active if the user opens, closes or saves an infected document. However recent versions of these macro-enabling applications display an alert concerning macro-virus risks and permit disabling of macro execution.
- **Boot sector**. A boot sector virus infects computers by modifying the contents of the boot sector program, replacing the legitimate contents with the infected version. This type of virus can only infect a machine if it is used to boot the system up. These viruses no longer represent a threat since operating systems now protect the boot sector, and floppy disks are actually unusable for storing modern (and large-size) applications.

- **Hoaxes**. We also mention hoaxes, which not are actually viruses but simply messages containing false information (usually alarming!) and/or instructions in pseudo-technical language, inducing unskilled people to perform actions which can provoke some damage. The user is often instigated to send this message to everyone in its address book. The result is an Internet chain letter, which provokes a flood of unnecessary emails and wastes bandwidth.

Although the main platform for spreading viruses is Windows, due to its widespread diffusion, possible infection vectors and vulnerabilities are also present in Unix (which registered a noticeable increase of bugs in "open source" SW), Macintosh, and also openVMS systems.

The virus trend in the few past years shows that the macro and boot viruses are decreasing greatly whereas, reflecting the mutation of their habitat conditions (increase and spread of networks), executables and worms represent the new threat.

The CERT/CC annual report for the year 2001 includes worms and DoS in the most common intruder activities and also underlines the increasing threat of attacks against or using DNS and routers [1], [2].

2.1 Infection

The main vehicle for spreading viruses is the Internet: millions of interconnected networks and systems represent a fertile ground for electronic infections. Once a computer is infected, the virus tries to take advantage of system and/or network vulnerabilities and by means of communication protocols and network resource sharing, is able to infect other hosts.

Although operating system and application safeguards are increasing, viruses continue to spread. Many factors contribute to facilitating the propagation of viral infections including the following [2]:

- The Internet was designed to favor simplicity and reliability when communication media were unstable, and to be open (i.e. fully interoperable), but basic protocols did not include native security mechanisms. In addition, the ability to attack a system depends on the 'global' security of the Internet: a DoS attack for instance takes advantage of other weak points in the network.
- Very relevant is the problem of SW bugs which are potential security holes used by hackers to penetrate the systems. The growing request for new applications and the short time-to-market requirements impose very short deadlines, which can reduce software quality and reliability. The number of vulnerabilities reported to CERT is constantly increasing (in 2001, double the number in the previous year).
- Viruses and in general attack tools became more sophisticated, increasing the level of automation, the modularity and the speed of replication (the infection is very active). Polymorphic viruses change dynamically, creating different variations thus making detection more difficult. They can use different encryption schemes and functions, vary the command sequence, etc. In addition the use of protocols such as HTTP or IRC which generate high and heavy traffic require many resources to efficiently analyze data and discern between legitimate and fraudulent traffic. As a consequence it becomes harder to prevent and avoid attacks.

- Active contents. This term refers to technologies such as Java, ActiveX, Javascript, etc. that can activate the dynamic execution of programs or code, client side. These languages offer many advantages such as decreasing the web server load (by moving the execution overhead from server to clients) and simplification of user interfaces. The problem is that malicious code may be hidden in the downloaded SW (i.e. to take advantage of the browser's vulnerabilities). Although digital signature can be associated with active content (i.e. a signed applet) assuring content integrity and origin authentication, there is no guarantee that this SW will not provoke any damage [3].
- User ignorance of security risks and mechanisms, together with increasing automation of applications (designed to facilitate user interaction) play a fundamental role in virus propagation. For instance, a large number of infections are propagated via email: the virus replicates itself by automatically mailing its copies to addresses contained in the user's address book. The user double-clicks on an attachment starting the application associated with its MIME[1] type (Multipurpose Internet Mail Extensions [4], [5]), thus enabling the virus to infect. Worse, some old versions of e-mail clients, in order to make the process friendlier, automatically triggered the application's execution on message reception (default option), thus increasing the virus' power to infect.

2.1 Prevention

A few basic norms can contribute to reducing the risk of infection and/or to limiting damage. Computer viruses usually act with the unwitting cooperation of the user. An effective defense is having the basic skill to deal with the problem as well as the full awareness of our behavior. The use of the following basic norms throughout the organization can contribute to a very effective anti-virus strategy:

- First of all: investing in education, making users security-aware. User education plays a fundamental role in preventing infections. Many viruses are activated by the user's unwitting complicity: i.e. double-clicking e-mail attachments, downloading files from the Internet, etc. People are curious and they are strongly tempted to try new tools or applications. Some guidelines include:
  - Do not trust file extensions. Fake extensions can be attached to executable files in order to trick the user (ie. XXX.VBS.COM or YYY.DOC.EXE).

[1] MIME (Multipurpose Internet Mail Extensions) [RFCs 2045, 2046, 2047, 2048, 2049] was conceived as an extension of the format of e-mail messages defined in RFC822 to permit the inclusion of multimedia data in the body of the message, while retaining its compatibility with the standard format. MIME defines basic data types text, image, audio, video, application, message and multipart. For each data type more subtypes can be defined (i.e. text/plain, text/html, image/tiff, image/gif, etc.). The *Content-type* header field specifies the media type/subtype of data in the body of a message and its canonical form. A multipart MIME message is a structured message composed of several parts containing different data types. Message parts are delimited by means of a "Boundary": a string starting with two --. The last boundary is also closed with two -- in order to end the message.

- When possible, disable all forms of automatic script or program start-ups such as Active and Java scripting;
- Exchange documents in printable formats i.e. PDF or PS, or in formats which do not contain macros such as Rich Text Format (RTF) instead of MS Word files, if they do not need to be modified;
- Do not use self-extracting files that do not show archive content;
- Report any virus alarm to the security administrator who can discern between genuine and false information (hoaxes);
- Set the PC to boot from drive C thus eliminating the effectiveness of boot sector viruses.

 Define an ad hoc security policy for your organization, including network monitoring and traffic control [6], [7]. The use of anti-virus software is only one component of an effective security policy, which in large organizations whose work is based on the net (i.e. e-commerce and e-business companies) requires a thorough and accurate analysis and evaluation, which are beyond the scope of this paper. However, at the very least one should:
  - block the transfer of executable files (i.e. EXE, VBS, etc.) on the Internet gateway and/or e-mail server;
  - configure servers disabling features/services not explicitly required,
  - configure routers/firewalls enabling only traffic to authorized servers and ports;
  - run and maintain updated anti-virus software in order to detect, report and, when possible, disinfect viruses;
  - read security bulletins and rapidly apply SW patches to OS and/or applications;
 Maintaining fully effective backup procedures for systems, applications and data, in order to retrieve safe copies, in case of virus attack.

In addition Intrusion Detection Systems (IDS), which conduct analyses of live network traffic [8] and Honeypots, usually single systems that emulate systems, known services or vulnerabilities, or create jailed environments, can contribute to monitoring network and system activities and detecting attacks [9], [10].

2.3 Damages of virus attack

Virus penetration can occur when the anti-virus is unable to recognize a new virus. It is difficult to evaluate the costs that a company must sustain to recover damages because they depend on many factors such as virus penetration, value of the corrupted data, recovery capacity of the organization, nature of company business, etc. Damages are not only quantifiable in terms of money but can include:

 time and resources necessary for disinfecting systems and restoring data and applications;
 work time lost until systems are restored;
 eventual interruption of network connection and therefore possible loss of revenue;
 decreased credibility for customers;
 violated confidentiality and possible stolen information.

To these consequences must be added the psychological factor, inducing persons afraid of contracting viruses to be removed from newsgroups and mailing lists.

To give an idea of the global economic impact of virus damage we only report data from Computer Economics, which estimated the full impact of Code Red to be 2.6 billion dollars [11].

Therefore, in order to limit damage, it is very important for large companies to detect and eradicate the infection as soon as possible, in order to restore systems and data.

3 The studio

In this paragraph we discuss general features of anti-virus SW and discuss data collected by anti-virus SW running on our e-mail server.

3.1 Anti-virus SW

Today the market offers a large set of sophisticated anti-virus strategies [12]. However, basic anti-virus techniques include [13]:

- **Scanners** that perform detection and disinfection of all known viruses (virus signature based systems). They are easy to use and furnish info about virus and infected files. The main problem is that to remain effective they must be constantly updated.
- **Checksummers** rely on detecting changes: if a virus infects an object then it changes. The main disadvantage is that they only detect infection but are unable to prevent it. In addition, checksummers' output must usually be evaluated by skilled personnel, to distinguish between legitimate and viral changes.
- **Heuristics** anti-virus software applies rules to detect viruses. However heuristic analysis remains effective only if new viruses are a re-make of an existing one (i.e. if it uses the same method of replication), otherwise the rules (and thus the SW) need to be updated. In addition heuristic software can mistakenly label objects as viruses when they are not, thus generating 'false alarms'.

The anti-virus SW (and/or content filtering) should be set up in strategic positions where the Internet traffic naturally flows, such as a point of access of network perimeter:

- **Internet/intranet gateways** or **routers** or **firewalls** or **proxies**. On the connection points between the internal networks and the Internet it is possible to check all incoming and outgoing traffic, as they represent possible virus entry points. Depending on the network's architecture, the use of anti-virus software or filtering rules at this level (i.e. for discarding unauthorized traffic) could be a way to catch or stop the virus before it enters the network and infects the systems. The main problem at this level is that encrypted files are not interpretable in transit but only upon arrival at the final destination and the inspection of traffic content means loading the CPU with an additional overhead that, if consistent, could slow down the data transit.

- **E-mail servers**. The anti-virus SW on the e-mail server permits scanning of every incoming and outgoing message (i.e. each MIME part composing the message) and removal of infected parts. Also in this configuration encrypted files are not interpretable in transit.
- **Servers**. If the organization utilizes centrally stored shared information (i.e. groupware server) the anti-virus SW can run on this server, thus reducing the network traffic in comparison to a client-side distributed solution.
- **User system**. Virus scanning on the client side is very important: if the virus penetrates the Internet gateway and/or firewall and overcomes the server scanner, it could be intercepted on the user system. In addition it must not be forgotten that a user desktop can represent a direct point of access for viruses, if he/she accesses an infected file on CD or floppy disk. However anti-virus SW in a user desktop can create problems such as late program execution when the scanning is active. Another drawback of this solution, also considering the increasing use of mobile computers, and the high number of clients in large organizations, is to keep the anti-virus software updated.

In large and complex networks an anti-virus multi-layered approach (e.g. using the anti-virus in all points of access) can be adopted, in order to compensate for the weak points of one layer with the strength of another. However, given that virus scanning shows limitations in detecting new viruses, this approach results ineffective if each layer preserves the same weakness. Therefore different contra-measures must be applied in order to build protection levels with different features [14].

3.2 The environment

The operative environment is represented by a research organization. Inside our organization we host e-mail services for various domains thus managing about 1000 user accounts globally.

The anti-virus SW was activated in August 2001. Before, we simply set up rules in our e-mail server in order to block potentially dangerous files such as executable or known viruses (based on file name or MIME type). However, the increasing number of new viruses and the need for a more reliable content filtering process, led us to adopt an anti-virus SW, to automate and simplify this task.

Since the effectiveness of anti-virus software depends on frequent updates, its administration is of critical importance. For distributing updates pushing, pulling or mixed techniques can be used. We chose automatic updating, which implies no workload, although it is rarely used since an organization often prefers to maintain full control over any installed software. In addition, our users are free to use an anti-virus or not on their PCs but updates are their responsibility.

Although the server manages only one thousand accounts with relatively low traffic, results are nonetheless significant.

3.3 The collected data

Figure 1 represents the temporal distribution of infected MIME parts vs. total inspected. We configure the system to substitute each infected part with a short message (text/plain part) reporting info such as virus name, file size, MIME type/subtype.

From August 1, 2001 until April 30, 2002 the SW examined 338442 MIME parts (with an average value of 37604 per month). A total of 15255 parts resulted infected by 48 different types of virus, with an average of 1695 per month, corresponding to 4,42 %.

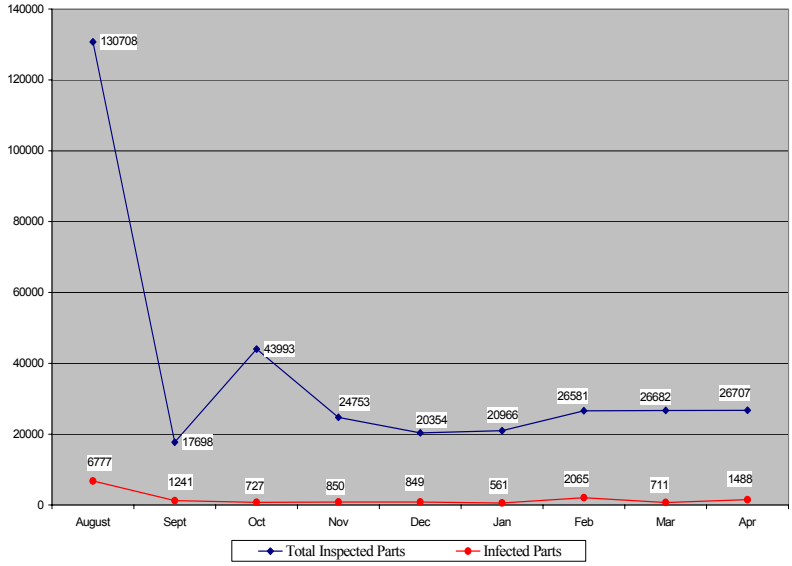

Fig. 1. Total scanned MIME parts vs. infected. A peak is registered in August mainly due to a propagation of one Sircam infection.

Figure 2 shows incoming and outgoing infected parts. Note that the majority of viruses came from outside, but a peak of outgoing infected messages was present in February.

The anti-virus SW block incoming and outgoing infected parts which cross the e-mail server; however, users often have a personal free e-mail account not subject to the organization's control. In this case their PCs can become infected, and the virus subsequently tries to self-replicate via e-mail (i.e. sending an infected message to all addresses in the user's address book), network/resource sharing (copying itself in another vulnerable host), etc.

If the virus uses its own SMTP libraries (or interacts with other "open relay" servers) to send infected messages, it bypasses the anti-virus running on the e-mail server, making it ineffective.

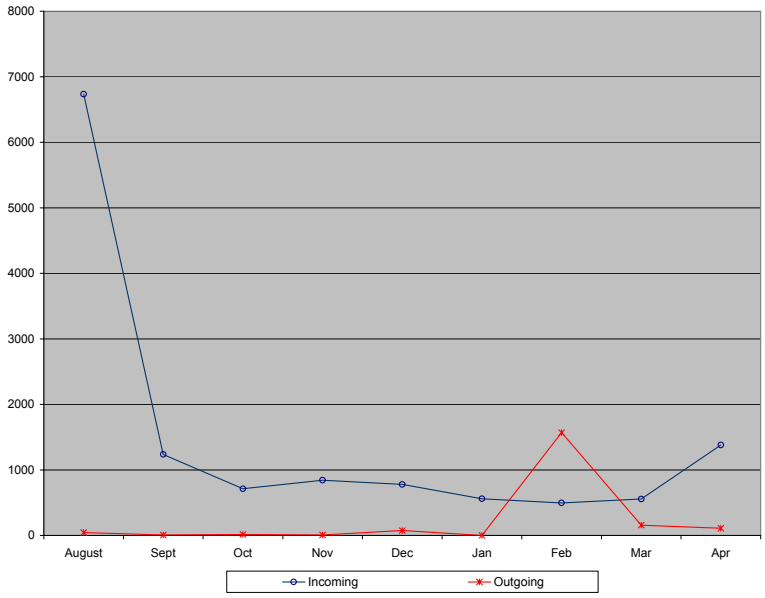

Fig. 2 - Infected MIME parts: incoming and outgoing. One peak of outgoing infected parts is present in February.

New generation viruses (such as Sircam and Nimda) which spread by multiple media (e-mail, open network sharing, web sites, etc.) make their infection faster and their eradication more difficult. Their propagation can persist for long periods of time, although with lower intensity.

Interesting data is provided by the types/subtypes of infected MIME parts as shown in Figure 3. Note that August values are not available, because the logging of MIME types/subtypes was enabled in September.

Registered infected parts include application/mixed, application/octet-stream, audio/x-wav, audio/x-midi, image/gif, application/x-msdownload, text/html, application/ms-word, application/rtf, and application/applefile.

Figure 4 shows the percentage of infected MIME types/subtypes. Actually the inspection of text/html parts was active only in October; otherwise, the percentile of infected html files would have been higher than the value reported in the figure.

Viruses are frequently hidden in application/mixed contents, which group MIME parts with different types/subtypes and application/octet-stream, usually containing an executable file.

Further an executable file may be *uuencoded* and put in a text/plain part thus to be effective the anti-virus SW needs to thoroughly inspect each MIME part.

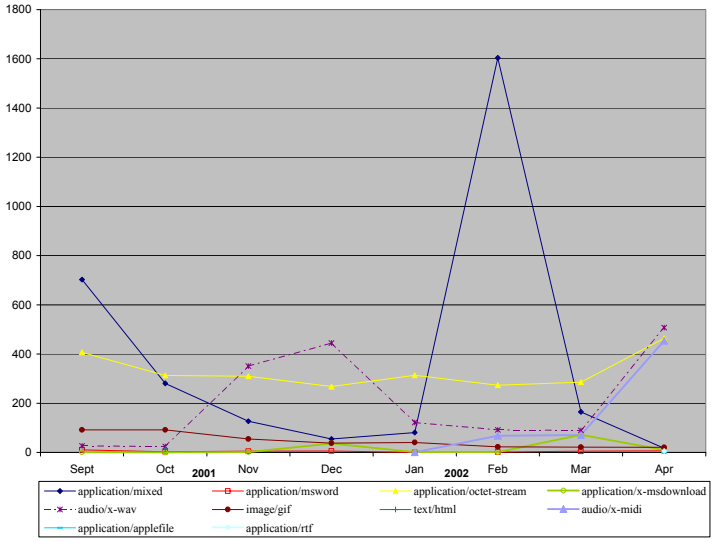

Fig.3 - Location of viruses in MIME parts. Application/mixed shows two peaks in September and February, corresponding to the Sircam attacks; audio/x-wav peaks in November/December and in April due to Badtrans and Klez-G infections respectively (see Fig. 5). Application/octect-stream does not peak but is constantly present.

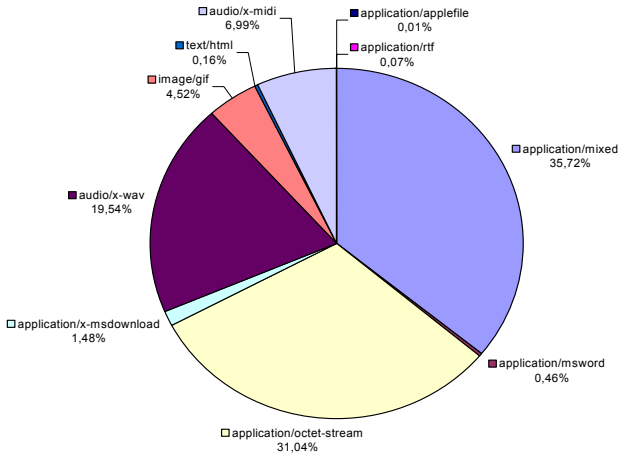

Fig. 4 - Percentage of infected MIME parts. Application/mixed, application/octet-stream, and audio/x-wav registered higher values.

Last, Figure 5 and Figure 6 show temporal distribution and percentage of well-known viruses respectively.

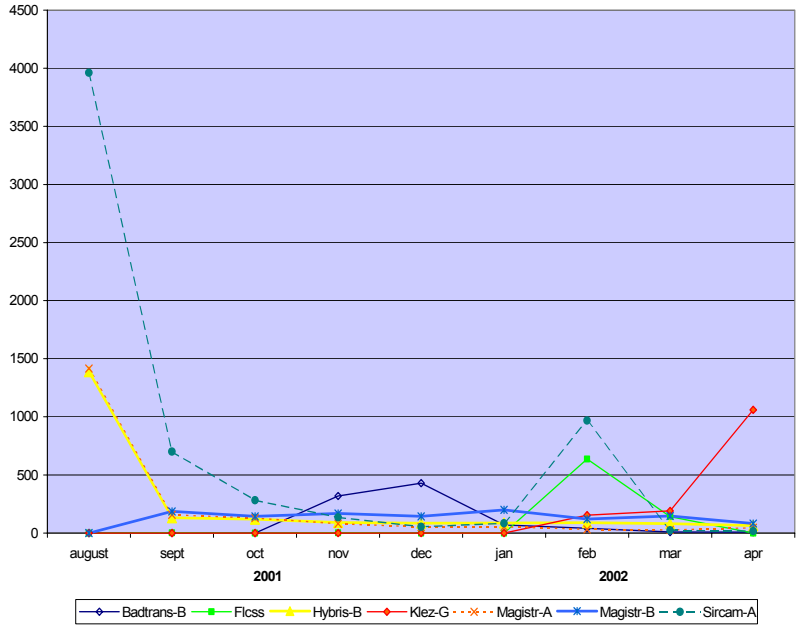

Fig. 5 - Temporal distribution of well-known viruses. We can observe that Sircam peaks in August and in February. The Badtrans infection provoked lower peaks in November and December. The most recent peak, due to Klez-G, is registered in April.

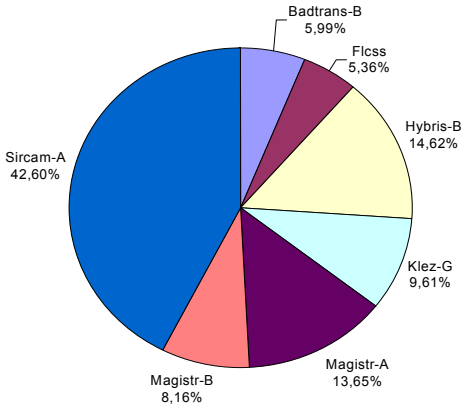

Fig. 6 - Percentage of well-known viruses in the monitored period. Sircam-A is the main presence, followed by Magister (A and B versions).

Recent attacks have used sophisticated techniques which combine worm-like propagation with Denial of Service activities, thus producing fast-spreading infections that cause enormous damage. Red Code, for instance, infected more than 250,000 systems in just 9 hours on July 19, 2001 [2]. The strong penetration of these types of attacks shows that anti-virus software is not effective and must be integrated with other technologies such as anti-intrusion and survivability techniques. In particular, survivability, i.e. the ability of systems to preserve essential services in the presence of an adverse environment (i.e. in case of attacks or failures) in unbounded environments, such as the Internet, is an open research topic [15].

In September 2001, we received an alert that a system in our network was scanning the Internet for potential vulnerabilities: in fact, the Code Red Worm had infected one of our systems (running MS IIS Servers). At this time we were using a transparent web caching architecture, thus all web traffic generated from our network was transparently intercepted and redirected (via router) toward a cache system. To detect the IP address of the infected system (hidden by the cache server), we needed to analyze the cache log files (containing the HTTP requests with the virus footprint). Actually, in order to respect Italian privacy law, we had anonymized log files (masking IP addresses of stations), so we first needed to change the cache server configuration to later detect the address of the infected system and patch it [16].

This incident confirmed the previous observation (§2.1): on the Internet the system's exposure to attacks depends on the state of security of the rest of the network [2].

4 Conclusion

Hackers and cyber-criminals who cause viral infection can be prosecuted under criminal law for their destructive behavior and sentenced [17]. Depending on national or international legislation, the inadvertent transmission of viruses, e.g. through an infected e-mail attachment, or by means of unprotected servers, may also be subject to civil prosecution.

In spite of this, the creation and spread of viruses is growing at an alarming rate and their degree of automation and sophistication is rapidly increasing. As technologies improve vulnerabilities are also on the rise, introducing security issues for handhelds and mobile phones.

Today, there is no way to implement a totally secure policy but it is necessary to create a strategy and use combined technologies to fend off intruder attacks and combat the virus plague.

Costs sustained by the organization when defining and setting up a security policy represent a valuable investment for the future.

References

1. CERT Coordination Center. 2001 Annual Report. http://www.cert.org/annual_rpts/cert_rpt_01.html
2. CERT Coordination Center. Overview of attack trends. http://www.cert.org/archive/pdf/attack_trends.pdf
3. Carr, K.: Active content: friend or foe. http://www.sophos.com/virusinfo/whitepapers/activecontent.html (2002)
4. Freed, N., Borenstein, N.: RFC 2045: Multipurpose Internet Mail Extensions. Part One: Format of Internet Message Bodies (1996)
5. Freed, N., Borenstein, N.:RFC 2046: Multipurpose Internet Mail Extensions. Part Two: Media Types. (1996)
6. Fraser, B.: RFC2196: Site Security Handbook. (1997)
7. CERT Security Improvement Module - http://www.cert.org/security-improvement/
8. Kandula, S, Singh, S. and Sanghi, D.: Argus: A Distributed Network Intrusion Detection System. Proceeding of SANE 2002, 27-31 May, Maastricht, pp.333-350. http://www.nluug.nl/events/sane2002/papers.html (2002)
9. The honeynet project. http://www.project.honeynet.org/
10. Donkers, A: Honey, I caught a worm. Building yourself a honeypot, some pratical issue. Proceeding of SANE 2002, 27-31 May, Maastricht, pp.304-318. http://www.nluug.nl/events/sane2002/papers.html (2002)
11. Press Releases - Malicious Code Attacks Had $13.2 Billion Economic Impact in 2001 http://www.computereconomics.com/
12. CERT Coordination Center. Computer Virus Resources http://www.cert.org/other_sources/viruses.html
13. Hruska, J.: Computer virus prevention: a primer. http://www.sophos.com/virusinfo/whitepapers/prevention.html (2000)
14. FitzGerald, N.: Free Anti-Virus Techniques. VB2002 Conference, 26-27 Sept, New Orleans (USA). http://www.virusbtn.com/VB2002/abstracts/free_techniques.html (2002)
15. CERT. Survivable network systems: an emerging Discipline. http://www.cert.org/research/97tr013.pdf
16. MS Security Bulletins. http://www.microsoft.com/security/
17. De Villiers, M.: Computer Viruses and The Law. VB2002 Conference, 26-27 Sept, New Orleans. http://www.virusbtn.com/VB2002/abstracts/the_law.html (2002)