Ulrich Flegel, Michael Meier (Eds.)

Detection of Intrusions and Malware & Vulnerability Assessment

GI Special Interest Group SIDAR Workshop, DIMVA 2004
Dortmund, Germany, July 6-7, 2004
Proceedings

Gesellschaft für Informatik 2004

Lecture Notes in Informatics (LNI) - Proceedings
Series of the Gesellschaft für Informatik (GI)

Volume P-46

ISBN 3-88579-375-X
ISSN 1617-5468

Volume Editors
Ulrich Flegel
University of Dortmund,
Computer Science Department, Chair VI, ISSI
D-44221 Dortmund, Germany
ulrich.flegel@udo.edu
Michael Meier
Brandenburg University of Technology Cottbus,
Computer Science Department, Chair Computer Networks
P.O. Box 10 13 44, D-03013 Cottbus, Germany
mm@informatik.tu-cottbus.de

printed by Köllen Druck+Verlag GmbH, Bonn

LIV - The Linux Integrated Viruswall

Teobaldo A. Dantas de Medeiros

GEINF – CEFET/RN
Centro Federal de Educação Tecnológica
Av. Senador Salgado Filho 1559, Tirol, Natal – RN - BRAZIL
59015-000
teobaldo@cefetrn.br

Paulo S. Motta Pires

DCA/UFRN
Universidade Federal do Rio Grande do Norte
Centro de Tecnologia – Natal – RN - BRAZIL
59.072-970
pmotta@dca.ufrn.br

Abstract This paper presents a system developed in Linux aiming the protection of local area networks containing Windows workstations against malicious agents. The developed solution, named LIV - Linux Integrated Viruswall, besides filtering **SMTP**, **HTTP** and **FTP** traffic destined to the protected network, is capable of detecting malicious agents propagation in the local area network using a technique that we call "sharing-trap". Compromised workstations are isolated from the network and their users are notified, stopping the malicious agent's spread. Results collected from a network protected by LIV, containing thousands of Windows workstations, are presented and discussed. This paper includes information about the recent incident caused by MyDoom worm.

1 Introduction

Malicious agents can be defined as computer programs that operate on behalf of a potential intruder, aiding it on the activity of attacking a system or a network [1]. Once limited to damages in the compromised systems, modern malicious agents acquired new characteristics, as the capability of transmitting private information to the program author, the possibility of remotelly control infected machines and the use of a compromised group of computers on a distributed denial of service (DDoS) attack.

During the year of 2001, the economical impact caused by malicious agents was estimated at $13.2 billion [2]. In 2003, only W32/Sobig.F [3], W32/Nachi [4], W32/Blaster [5] and W32/Slammer [6] worms were responsible for economical losses estimated at $3.25 billion. These values show the growing importance of the adoption of actions that reduce damage caused by the malicious agents on systems and on computer networks.

The traditionally proposed model for protection against malicious agents [7] consists of three protection layers: the first of them acting on the Internet gateway, the second protecting the file and mail servers and the third protecting the workstations. In spite of the immunity propitiated by this model, a basic vulnerability persists: The propagation speed of new malicious agents, unknown by the antiviruses that work on three layers of the model, allows that apparently "protected" networks and workstations continue being infected [8].

In this work, we present an Internet gateway solution that aim to protect local area networks against malicious agents. The solution, named LIV, Linux Integrated Viruswall, is endowed with features implemented in other products [9,10], such as **SMTP**, **HTTP** and **FTP-**traffic filtering [11,12,13], and also incorporates new functionalities. Among those new functionalities, we highlight: the use of the sharing-trap to detect malicious agents spreading in the local area network, analysis of the network traffic generated by the workstations to determine the infected ones, isolation of compromised workstations from the network and the use of a proxy server as a communication channel between the protection system and the users of the workstations. Therefore, LIV is not limited to merely preventing the contamination of the protected network. In the case where a malicious agent gets to enter in the network, deceiving the traditional defense mechanisms, LIV will act limitating its propagation on the LAN.

LIV is constituted of a group of 10 processes, named **ISPAMA** - Integrated System for Protection Against Malicious Agents. **ISPAMA** coordinates the behavior of **CIFS** (Common Internet File System [14]), **SMTP** , **HTTP** and proxy servers. The operation of an antivirus scanner is also controlled by **ISPAMA**. LIV uses the firewall functions of the Linux kernel, via iptables [15], and a database manager for the storage and information exchanging among the various **ISPAMA** processes. The use of a **CIFS** server made possible the creation of a network sharing (sharing-trap), published and made available, without access restrictions, to Windows workstations. The sharing-trap aims to cause the replication of malicious agents to the LIV. Machines that transmit malicious agents to the sharing-trap are considered infected and are isolated from the network. The **SMTP** server acts in the analysis of the e-mail attachments, preventing the entrance of known malicious agents, as well as putting suspect files in quarantine. The **HTTP** server is used for the LIV configuration and, together with the proxy server, acts in the analysis of the downloads made by users of the protected network. The **HTTP** server and the proxy server are also used as a communication channel between the LIV and the users of the network, allowing, for instance, notification of users of the compromised machines in case of infection. The Linux firewall acts in the isolation process of the compromised machines and in the generation of logs related to the workstations' traffic. These logs will be stored later in the LIV database and analyzed by

the **ISPAMA** processes. The log analysis is another method used by the LIV to discover the infected workstations in the network.

After this introduction, we present in the Section 2 the architecture of the LIV. In Section 3, we describe the operation of the **ISPAMA** processes. Section 4 details the operation of **CIFS** server and the sharing-trap. Section 5 discusses the results obtained by LIV in the recent W32/MyDoom [16] incident, reserving to Section 6 the conclusions of the present work.

2 LIV Architecture

The current version of LIV is implemented in Slackware 9.0 Linux distribution [17]. However, there is no known incompatibility with the implementation of LIV in other distributions, since servers and programs that were used in the solution are also available on these other distributions. It is important to emphasize that not all packages used by LIV are present in Slackware 9.0 distribution. Furthermore, some of these packages were substituted by newer versions, or recompiled to support options not available in the distribution version.

At this moment, LIV is protecting a network containing thousands of Windows workstations. The protected network is connected to a single 6 Mbps Internet link. In this particular case, LIV was implemented on a monoprocessed CISC server, with 512Mb of RAM, presenting a quite satisfactory performance. Some results collected from this network configuration are presented on Section 5.

Ten processes run in the LIV server, and these processes are responsible for the implementation of the protection against malicious agents, sharing information amongst themselves by the use of the LIV database. The group of ten processes plus the database is denominated **ISPAMA** - Integrated System for Protection Against Malicious Agents. The specific operation of each **ISPAMA**'s process will be discussed in the Section 3.

Besides the **ISPAMA** processes, several other servers are executed in the LIV machine. **ISPAMA** coordinates the behavior of these servers and, when necessary, activates the functions of the Linux firewall to limit the spread of malicious agents throughout the local area network. A scanner is used to make verifications on e-mail attachments and on downloads.

The applications managed by **ISPAMA** are available for several Linux distributions. These programs are a **CIFS** server, implemented by Samba [18], a **SMTP** server, implemented by Sendmail [19], a **HTTP** server, implemented by Apache [20], and a proxy server, implemented by Squid [21]. It is also necessary to activate the firewall functions of the Linux kernel. Figure 1 shows the general architecture of LIV.

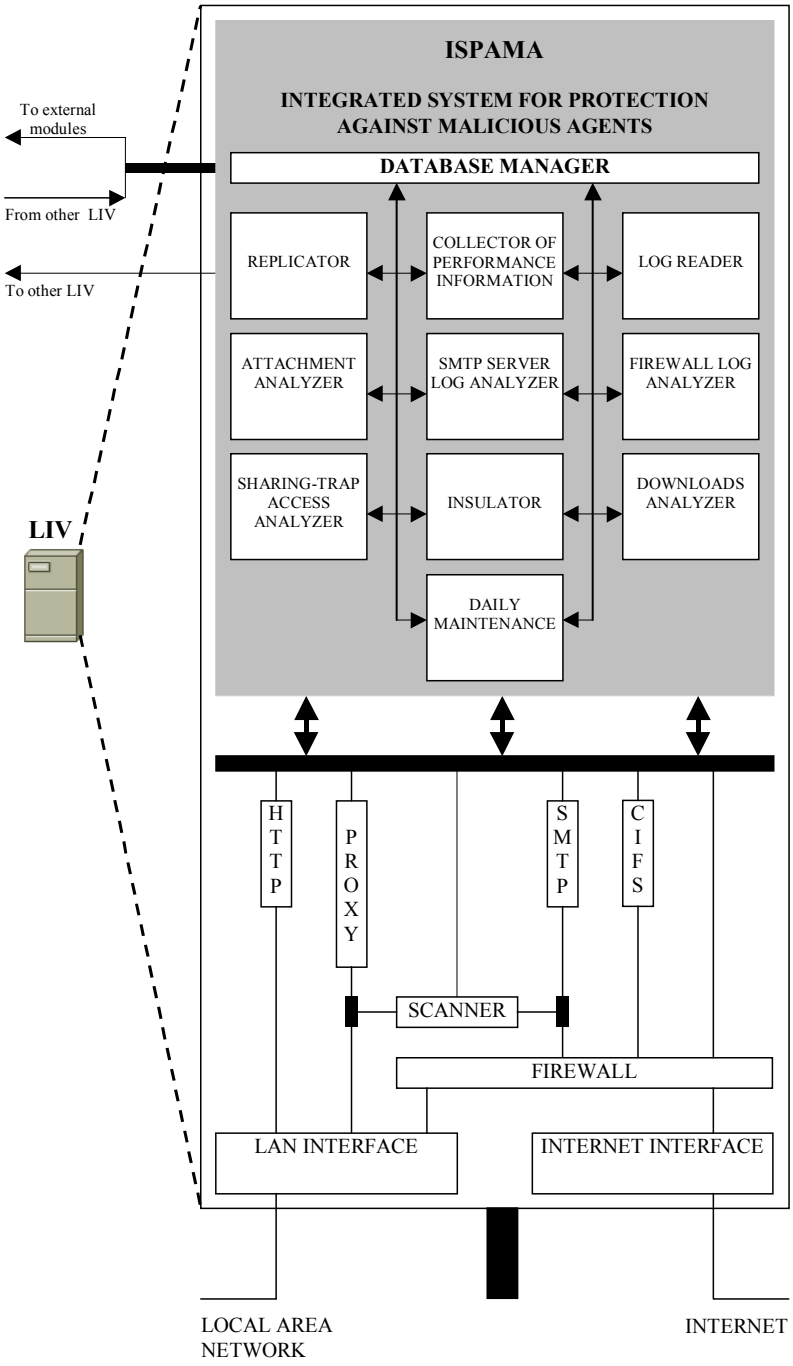

Figure 1: LIV Architecture

As shown on Figure 1, one LIV server can exchange information with another. The use of a larger number of LIV servers make it possible to increase the capacity of limiting malicious agents spread throughout the LAN, in the case where some of them get to enter in the protected network. If a machine is contaminated, LIV will isolate it from the network, preventing that the malicious agents access shares of other workstations or the mail servers of the organization and continue the propagation process on the network. The isolation is implemented in the LIV servers by reconfiguring the Linux firewall. Additionally, LIV allows the use of external insulating modules, that are responsible for the programming of departmental routers of the organization. The external modules instruct routers to filter the packages originated from infected workstations.

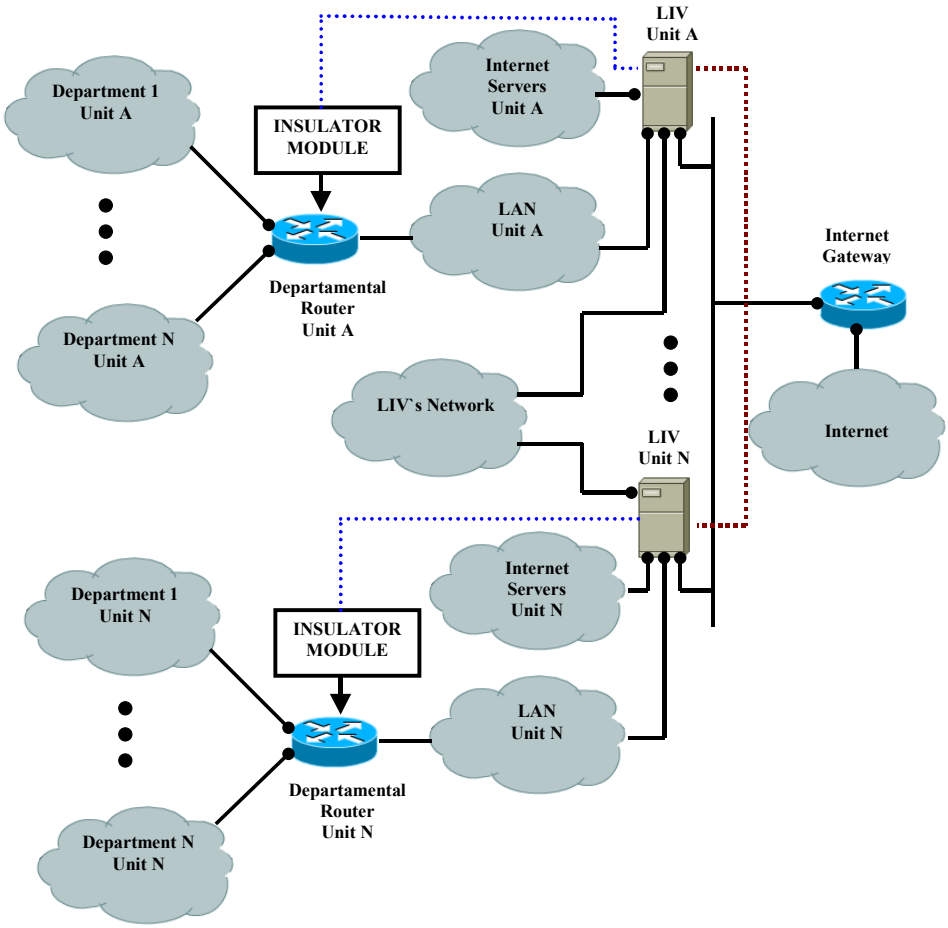

Figure 2: Typical network topology using LIV.

LIV can be used in simple or in complex network topologies. In the simplest network topology, the organization has no Internet servers and only one single LIV is used. In this case, only two network interfaces (LAN and Internet) are needed [22]. Figure 2 shows a more complex situation. In this example, an organization is subdivided in units

and departments. The organization has a single Internet link shared by all units, and each unit has its own Internet servers, including mail servers, and a LIV server. The LIV server is connected to the unit's LAN and to the Internet servers hosted on that unit. Additionally, the unit's LIV is connected to the Internet gateway network and to a communication network constituted of all LIV servers of the organization. This LIV network is used to share information concerning infected workstations and banned e-mail address.

In the next section, we will describe **ISPAMA** processes in detail.

3 ISPAMA Processes

ISPAMA controls all decisions taken by LIV. The **ISPAMA** processes are responsible for reading logs generated by Linux and inserting the records in the LIV database. Based in log analysis, the LIV tries to determine if there are infected workstations in the protected network. Accesses to the sharing-trap and the e-mail attachments verification are also used by LIV to identify infected workstations. Some other processes perform tasks such as replication control, isolation and collection of performance data.

3.1 Log Reader

The log reader stores on the database information generated by the firewall Linux and by the mail server. Only relevant information to the isolation decision is stored. LIV controls the number of log reader processes running according to the amount of records generated by Linux.

3.2 Firewall Log Analyzer

LIV examines the traffic generated by the workstations based on rules defined by its administrator. The rules define the port and the protocol that LIV will monitor in the network, allowing the log reader to configure the firewall so that it will start registering the packets related to the new rules. Besides the port and the protocol, the rules contain other attributes that are periodically anallyzed by the firewall log analyzer. When one of the defined attribute values is exceeded, the workstation that generated these packets will be isolated of the network. Table I summarizes the attributes used in firewall rules.

Attribute	Explanation
Destination Port (*PORT*)	Destination ports of the connections (TCP protocol) or datagrams (UDP protocol) that will be examined by the rule.
Protocol	Protocol used by the packets examined in this rule (UDP or TCP).
Limit of Connections destined to LIV Server (*LCLIV*)	Defines a limit to the number of connections destined to the LIV server that a workstation is allowed to establish using stipulated port/protocol in the interval specified by the rule
Limit of Connections destined to Intranet (*LCIN*)	Defines a limit to the number of connections destined to Intranet addresses that a workstation is allowed to establish using stipulated port/protocol in the interval specified by the rule
Limit of Connections destined to Internet (*LCOUT*)	Defines a limit to the number of connections destined to Internet addresses that a workstation is allowed to establish using stipulated port/protocol in the interval specified by the rule
Limit of Periodical Accesses (*LPA*)	Defines a limit to the number of periodical connections that a workstation is allowed to establish using stipulated port/protocol in the interval specified by the rule
Interval	Time interval used to restrict queries sent to LIV database. Only log records generated in the rule interval are computed when verifying the traffic generated by a workstation

Table I: Firewall rules attributes.

3.3 SMTP Server Log Analyzer

The **SMTP** server log analysis is similar to that described in the previous section for the firewall log analysis. The main difference between them consists on the type of address analyzed. Firewall analysis works with IP addresses of the workstations, and the **SMTP** server analysis works with e-mail addresses. The attributes of the mail server rules also differ from the firewall ones and are presented in Table 2.

Attribute	Explanation
Distinct Originator Addresses (DOA)	Defines how many distinct origin e-mail addresses one workstation can use in the defined rule interval
Recipients Limit (RLM)	Defines the maximum number of distinct recipients in all the e-mail messages sent by a workstation in the rule interval
Messages per Recipient (MPR)	Defines how many messages can be sent by one workstation to the same recipient in the rule interval
External Domain Limit (EDL)	Defines the maximum number of messages sent by a workstation using an external domain in the originator address
Interval	Time interval used to restrict queries sent to LIV database. Only log records generated in the rule interval are computed when verifying the traffic generated by a workstation

Table II: **SMTP** server rules attributes.

If the administrator wishes, some of the rule's attributes can be ignored by LIV in the log analysis processes.

3.4 Insulator

The insulator process is responsible for configuring the Linux firewall whenever a workstation is isolated or reintegrated into the network. The firewall is also configured if the LIV's rules have been changed. The isolation filters the traffic generated by infected workstation, allowing only accesses to essential services, such as name resolution, World Wide Web (**WWW**) to intranet servers and access to proxy ports on LIV servers. Besides interacting with firewall, the insulator alters the proxy server configuration so that it will start blocking the access of infected workstations to the Internet. Whenever an Internet access is denied, the proxy server will inform the workstation user about the infection and the isolation of his machine. The last insulator function is to alter the **SMTP** server configuration so that it will temporarily reject the reception of e-mail sent by addresses that are transmitting malicious agents to the protected network.

3.5 Attachment Analyzer

Each e-mail originated or destined to the protected network is analyzed by LIV. This analysis will initially verify the existence of malicious agents in attachments. If some malicious agent is found, the action taken by LIV will depend on the address of the sender of the message. If the sender's IP address belongs to the intranet, the workstation will be isolated from the network. Otherwise, the sender's e-mail address will be put in a **SMTP** server rejection list. If LIV does not find malicious agent in attachments, it will perform a second analysis, that consists of removing dangerous existing attachments in the message. The LIV's administrator defines which kind of files will be accepted or refused in attachments. Usually, executables, batch files and similar ones should be refused.

3.6 Other ISPAMA Process

The five remaining **ISPAMA** processes are the following: Sharing-trap access analyzer, downloads analyzer, replicator, collector of performance information and daily maintenance. The sharing-trap access analyzer will be discussed in the next section and the operation of the other four processes will be summarized now.

The download analyzer performs a scan operation in files downloaded via the proxy server, being an incumbency of the LIV's administrator to determine which file extensions will be examined and which will not. The replicator process periodically sends information about infected workstations and about the banned e-mails addresses to LIV partners in the protected network. The function of the process collector of performance information is to obtain data about the CPU usage and memory resources

on the LIV server. The collected data are presented graphically in the LIV WEB interface. Finally, the daily maintenance process accomplishes tasks like exclusion of old log records stored in LIV database and the removal of files put in quarantine on server's disk.

4 The Sharing-trap

The sharing-trap is a technique used by LIV to detect workstations compromised by malicious agents that are capable of spreading themselves throughout local area network. Any workstation can access the sharing-trap. There are no access control, therefore LIV will accept access independently of the network credentials informed to the **CIFS** server. Additionally, if a workstation searches for a sharing name inexistent in the **CIFS** server, LIV will map this access to the sharing-trap. Figure 3 illustrates how a Windows workstation sees the sharing-trap in the network.

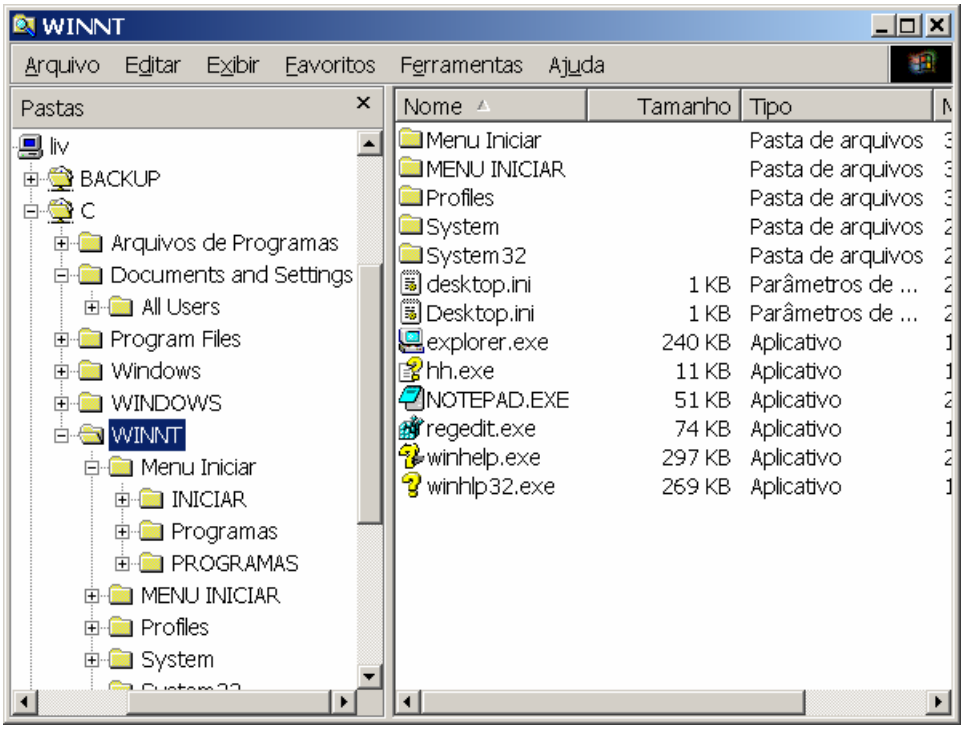

Figure 3 Sharing-trap accessed using a Windows Workstation.

When an access to the sharing-trap is concluded, **CIFS** server runs a LIV process that scans for malicious agents. LIV will isolate the workstation that accomplished the last access to the sharing-trap if some malicious agent is found in the scan. The files of the sharing-trap are restored in the case of some alteration is detected during the analysis

process. Figure 4 shows the key configuration of the Samba server to implement the sharing-trap.

```
1.  [global]
2.      workgroup = EXAMPLE
3.      netbios name = LIV
4.      server string = Linux Integrated Viruswall
5.      interfaces = 192.0.2.0/255.255.252.0
6.      bind interfaces only = Yes
7.      security = DOMAIN
8.      encrypt passwords = Yes
9.      map to guest = Bad Password
10.     password server = EX_DC1, EX_DC2
11.     username map = /etc/samba/private/smbalias
12.     deadtime = 2
13.     wins server = EXAMPLE:192.0.2.10
14.     default service = C
15.     remote announce = 192.0.2.10/EXAMPLE 192.0.2.11/EXAMPLE
16.
17. [C]
18.     comment = LIV Sharing Trap
19.     path = /usr/local/liv/armadilha
20.     admin users = nobody
21.     read only = No
22.     guest ok = Yes
23.     root postexec = /usr/local/liv/cifs %I &
24.     volume = LIV
25.     fstype = FAT
26.     dos filemode = Yes
27.     dos filetimes = Yes
28.     dos filetime resolution = Yes
```

Figure 4. Key configuration of the Samba server to implement the sharing-trap.

Figure 4 shows the case of a LIV server that is member of the Microsoft domain EXAMPLE. Line 9 of the configuration file instructs the Samba server to map invalid user accesses to a guest account. This guest account can access the sharing-trap "C", configured by the lines 18-28. Line 14 redirects accesses to inexistent shares names to the sharing-trap. Line 22 allows guest access to the sharing-trap, and line 20 grants administrative privileges to the guest account. Line 23 states that sharing-trap access analyzer is activated after each share access.

In the next section, we will discuss the results obtained by LIV in two months of operation protecting a network composed of thousands of Windows workstations.

5 Results

The results presented in this section were obtained in a network containing more than 6,000 Windows workstations distributed by approximately 180 remote places in the Brazilian state of Rio Grande do Norte. The network topology is similar to that presented in Figure 2. The data were collected in the period from January 14, when LIV was implanted in the network, until February 26, 2004. During this period, LIV analyzed 691,184 e-mails, removing 6,658 malicious agents found in attachments. The amount of scanned downloads has summed 40,467, on which 38 were infected. Figure 5 presents

incidents involving all known variants of MyDoom, Bagle [23] and NetSky [24] in the protected network with a period of one day for each interval. In all these cases, LIV removed the malicious agents from the message, replacing the attachment with a warning message. After that, the warning message was sent for the sender and for the recipient of the e-mail.

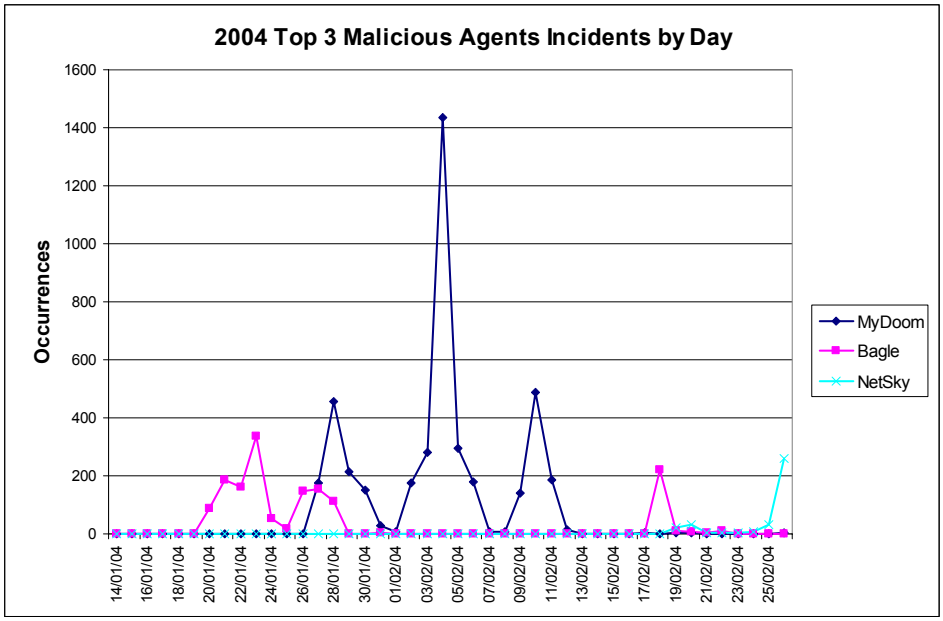

Figure 5. MyDoom, Bagle and NetSky removed from e-mail attachments with a daily period.

Table III presents some data obtained from LIV database about MyDoom.

GENERAL DATA	
MyDooms removed from attachments	4253
Number of infected machines in the protected network	31 (0.52 %)
Propagation peak	04/02/2004
Percentage of infected e-mails in the peak of propagation	6.05 %
SOURCE OF E-MAIL MESSAGES CONTAINING MYDOOM	
LAN (sent by the 31 infected machines before isolation)	3275 (77 %)
Internet	978 (23 %)
→ Average number of e-mails sent from infected machines before the isolation (with an maximum interval of 5 minutes):	105.65
SOURCE OF E-MAIL MESSAGES CONTAINING MYDOOM AND COMING FROM THE INTERNET	
Brazil	881 (90,1%)
Other Countries	97 (9,9%)

Table III. Information concerning MyDoom collected from LIV database.

An inquiry made in some of the 31 stations infected by MyDoom in the protected network had demonstrated that these machines were used in others networks (most of them are notebooks) or that they have also been connected to the Internet via dial-up. In some cases, it was reported that machines were compromised when reading infected messages in Internet webmails, placed outside the LIV protection perimeter. This occurs because some webmails are incompatible with the download protection method used by LIV.

Another significant result of LIV refers to the performed isolation operations. The network where LIV is operating had hundreds of infected machines prior to the LIV implantation. On January 14, when LIV has started its operation, practically all of them were immediately isolated. From January 14 until February 26, LIV accomplished 672 isolation operations. In 376 of these isolations, the network support team of the organization was able to determine with reliability whether the machine was or not infected. In 91.76% of these cases, the machine was really infected. The 8.24% remaining constitute the false positives, when LIV isolates a machine that is not infected. A false positive can occur due to wrong network configuration at workstations, or due to improper LIV parameters setting. Typically, when LIV server is configured so that it will be able to detect and isolate a higher number of infected workstations, it will also cause more false positives.

In the next Section we present the conclusions of this work.

6 Conclusions

We presented a system for protection against malicious agents that acts on the Internet gateway of the network. The solution, named LIV - Linux Integrated Viruswall, is capable of preventing malicious agents entrance in the protected network as well as detecting already infected workstations. Compromised workstations containing malicious agents are isolated from the network and their users are notified. LIV introduces new features in comparison with other solutions. First, LIV uses the sharing-trap technique to detect malicious agents spread through LAN. Additionally, LIV analyzes the network traffic generated by workstations to verify if they are compromised. Another innovative feature is the use of the proxy server as a communication channel with the users of the infected machines, making it possible to inform them when a malicious agent infect their machines even when there is no local antivirus installed.

After 44 days of its implantation on a network composed of more than 6,000 Windows workstations, LIV had already blocked the entrance of approximately 6,700 malicious agents in the protected network. LIV had also detected the infection of at least 345 machines, isolating them from network, and avoiding the malicious agents spread. In the analyzed period, 368 attached files had been put in quarantine. Many of these files contained malicious agents still unknown at the time in which they were analyzed by the LIV's scanner.

The features incorporated to LIV, together with the results obtained, demonstrate that it is possible to significantly increase the security of a computer network against malicious agents by using a regular computer with no special hardware, acting in the gateway of the protected network.

References

[1] Zelser, Lenny: The Evolution of Malicious Agents. Online publication, 2000. Available at http://www.zeltser.com/agents, may 2004 .

[2] Computer Economics: Malicious Code Attacks Had $13.2 Billion Economic Impact in 2001. Online publication, 2002. Available at http://www.computereconomics.com/article.cfm?id=133, may 2004.

[3] CERT: CERT® Incident Note IN-2003-03. W32/Sobig.F Worm. Online publication, 2003. Available at http://www.cert.org/incident_notes/IN-2003-03.html, may 2004.

[4] Symantec: W32.Welchia.Worm. Online publication, 2003. Available at http://securityresponse.symantec.com/avcenter/venc/data/w32.welchia.worm.html, may 2004.

[5] CERT: CERT® Advisory CA-2003-20 W32/Blaster worm. Online publication, 2003. Available at http://www.cert.org/advisories/CA-2003-20.html, may 2004.

[6] CERT: CERT® Advisory CA-2003-04 MS-SQL Server Worm. Online publication, 2003. Available at http://www.cert.org/advisories/CA-2003-04.html, may 2004.

[7] Trend Micro: Virus Protection Across The Enterprise. Online publication, 2003. Available at http://www.trendmicro.com/en/products/gateway/gatelock5000/evaluate/whitepaper.htm, may 2004

[8] CERT: CERT® Incident Note IN-2003-01. Malicious Code Propagation and Antivirus Software Updates. Online publication, 2003. Available at http://www.cert.org/incident_notes/IN-2003-01.html, may 2004.

[9] AMaViS Team: AMaViS - A Mail Virus Scanner. Online publication, 2004. Available at http://www.amavis.org/, may 2004.

[10] Trend Micro: InterScan VirusWall - Features. Online publication, 2004. Available at http://www.trendmicro.com/en/products/gateway/isvw/evaluate/features.htm, may 2004.

[11] Postel, J: Simple Mail Transfer Protocol. Online publication, 1982. Available at http://www.ietf.org/rfc/rfc0821.txt, may 2004.

[12] Fielding, R. et al.: Hypertext Transfer Protocol - HTTP/1.1. Online publication, 1999, Available at http://www.ietf.org/rfc/rfc2616.txt, may 2004.

[13] Postel, J et al.: File Transfer Protocol. Online publication., 1985. Available at http://www.ietf.org/rfc/rfc959.txt, may 2004.

[14] Hertel, C: IMPLEMENTING CIFS - The Common Internet File System, Prentice Hall Professional Technical Reference (PTR), 2003.

[15] Russel, Rusty et al.: THE NETFILTER/IPTABLES PROJECT. Online publication, 2004. Available at http://www.netfilter.org/, may 2004.

[16] CERT: CERT® Incident Note IN-2004-01 - W32/Novarg.A Virus. Online publication, 2003. Available at http://www.cert.org/incident_notes/IN-2004-01.html, may 2004.

[17] Slackware Linux, inc.: The Slackware Linux Project. Online publication, 2003. Available at http://www.slackware.com, may 2004.

[18] Samba Team: The Samba Web Pages. Online publication, 2004. Available at http://www.samba.org, may 2004.

[19] Sendmail: Sendmail Home Page. Online publication, 2004. Available at http://www.sendmail.org, may 2004.
[20] Apache Software Foundation: The Apache Httpd Server Project. Online publication, 2004. Available at http://httpd.apache.org, may 2004.
[21] Team Squid: Squid Web Proxy Cache. Online publication, 2004. Available at http://www.squid-cache.org, may 2004.
[22] Dantas de Medeiros, Teobaldo A.; Motta Pires, Paulo S. : LIV: LINUX INTEGRATED VIRUSWALL – UMA ESTRATÉGIA PARA A PROTEÇÃO DE ESTAÇÕES DE TRABALHO CONTRA CÓDIGOS MALICIOSOS EXECUTÁVEIS: Proc. of IV Simpósio de Segurança em Informática – ITA/CTA, São José dos Campos – SP – Brazil, 2002; p 38-43 (in portuguese).
[23] Network Associates: W32/Bagle.b@MM. Online publication, 2004. Available at http://vil.nai.com/vil/content/v_101030.htm, may 2004.
[24] Computer Associates: Virus Information Center - Win32.Netsky.B. Online publication, 2004. Available at http://www3.ca.com/virusinfo/virus.aspx?ID=38332, may 2004.