A Novel Approach to Troubleshoot Security Attacks in Local Area Networks

Y V Srinivasa Murthy
Assistant Professor
Dept. of CSE, ANITS

G Jagadish, K. Mrunalini
Assistant Professor
Dept. of CSE, ANITS

Kakarla Siva
Sr. Consultant
HCL Technologies

P V V Satyanarayana,
V Nikhil Raj Kumar
¾ B. Tech (CSE) ANITS

Summary
Many Software Industries today use huge number of computers to handle their projects in an effective way that is obviously connected in a dedicated Local Area Network, which is geographically less size. Connecting systems through LAN is very useful with respect to File Sharing, Common File Access, and Work Sharing etc.., but lack of standard LAN security. The difficulty for these systems is from the Deadly and Self – Explicative WORMS. Worms are always likely to infect the systems in LAN. Unlike virus, worm spreads to all the systems connected in LAN within a very short period of time thereby making a huge loss to the Industry regarding its economy without the actual intervention of the user. The user doesn't know even that some worm has taken control over the system. In order to provide security to the systems, there is need to detect a worm immediately, and stop the spread of that worm to other systems. Now we are detecting the worms by analyzing the probable paths where they tend to copy themselves. Hence a worm could be identified and deleted instantly. Even if the path the worm copies itself changes and can't be traced, we detect it by capturing the content in packets flowing between the systems in the LAN using JPCAP. Ultimately the Infectious packets i.e., the packets which contain the worm or malicious data can be traced.

Key words:
Local Area Networks, Computer Worm, Security in LAN, SNORT Rules, Intrusion Detection System

1. Introduction

With the increase in Computerization all over the globe, its been a real hard task to provide security for the systems which consists of highly confidential data. In many cases, the systems that are employed in huge number will be connected through a Local Area Network, shown in Fig1 thus to enhance the File Transfer and File Access properties. This can make a task to be completed in a quick time rather than a single person handling it by himself. Now, the question arises that how secure these computers could be. If a small worm happens to attack a terminal system in the network, it immediately gains control over the system without the actual intervention of the user and starts replicating itself and spreads to all other systems in the network. Thus, within no time entire network goes out of control from the users and gets destroyed thus losing all the confidential information. Hence such is the importance for ensuring security to the systems.

In this paper, we tend to find a new solution for ensuring the security for the systems connected in Local Area Network. We also analyze the time that is required for the process to be completed and compare our proposed system with the existing systems with respect to compatibility, time quantum and performance. The main goal of the paper is that it should detect the worms or any malicious information if at all they exist in the any system in the network and immediately stop them from spreading all through the network**.** In this paper, we suggested that the required inputs are existence of any worm in specified location or transfer of any infectious packets and the desired outputs are the worm that is copied to the specified location will be detected and deleted. If an infectious packet is found to be transferring over the network, it will be discarded.

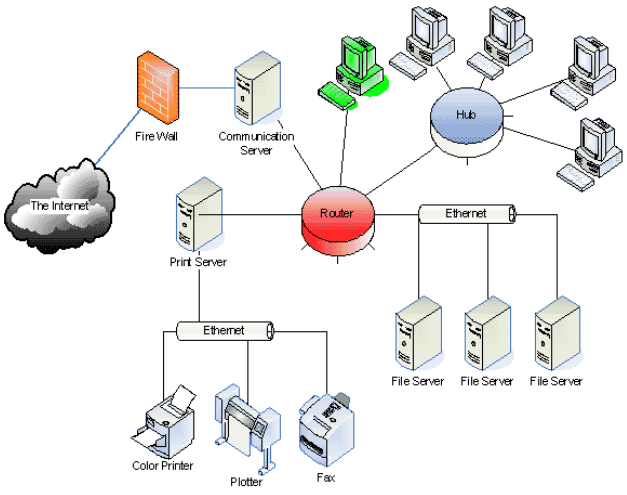

Figure 1: Interconnected four systems with server in Local Area Network

The rest of the paper is organized as follows: Section 2 discusses the literature survey of worms in existing LAN system and their limitations with respect to the security. Section 3 describes about SNORT rules. Section 4 details the proposed model towards worm detection system with snort rules. The performance measurement results of the

Manuscript received September 5, 2011
Manuscript revised September 20, 2011

proposed model are presented in Section 4. Finally we conclude this paper in Section 6.

2. Background

Systems when connected in a Local Area Network are always prone to get affected by the dangerous worms and other malicious data. If unfortunately a system in Local Area Network happens to be infected by a worm, then within no time the worm being self - replicative, replicates itself and spreads to other systems, crashing the entire network within seconds thus causing huge damage to Industries Economy. Hence this is considered to be a very B serious issue. Hence in our new designed system, we maintain a database of all the worms existing today and the locations where they probably copy themselves into the system so that any worm can be immediately detected and if found, can be deleted. Though if any worm doesn't copy itself to the specified location expected, it can be traced by capturing the data from the packets that flow from system to system in the Network. This data that is captured from the packets flowing through the network is compared with the data that is obtained from the Rules of Snort Intrusion Detection System which contains all the Infectious packet content information. If in case the data from snort rules and data from the packets transferring is found to be matched, the packet is analyzed to contain some worm or malicious data. In this way, we ensure high security for the Lan systems basing on worm patterns using snort rules and locations they copy themselves.

2.1 Worms

A Worm is more dangerous being a self replicating program as compared to a virus. Since it doesn't need any user to do malicious activities, it needs to be handled carefully so as to keep the systems safe and working. Many worms that have been created are only designed to spread, and don't attempt to alter the systems they pass through. Beginning with the very first research into worms at Xerox PARC, there have been attempts to create useful worms. The Nachi family of worms, for example, tried to download and install patches from Microsoft's website to fix vulnerabilities in the host system – by exploiting those *same* vulnerabilities. In practice, although this may have made these systems more secure, it generated considerable network traffic, rebooted the machine in the course of patching it, and did its work without the consent of the computer's owner or user. Some worms, such as XSS worms, have been written for research to determine the factors of how worms spread, such as social activity and change in user behavior. [1] [7]

Worms spread by exploiting vulnerabilities in operating systems. All vendors supply regular security updates, and if these are installed to a machine then the majority of worms are unable to spread to it. Users need to be wary of opening unexpected email and should not run attached files or programs, or visit web sites that are linked to such emails. However, as with the **ILOVEYOU** worm, and with the increased growth and efficiency of phishing attacks, it remains possible to trick the end-user into running a malicious code. [2]

List of some worms :

adtrans	Bagle	Brontok
Blaster	Code Red [3]	Code Red II
Dabber	Doomjuice	ExploreZip
Father Christmas	Hybris	Hydra
ILOVEYOU	Kak worm	Klez
Mabutu	Melissa	Morris
Mydoom	Mylife	Netsky
Nimda	SadmindSasser	
Sircam	Sober	Sobig
SQL slammer	Swen	Supernova worm
Upering	Bolgimo.worm	W32/Fus.worm
W32/IRCbot.worm	Wank	Welchia
Witty	Zotob	Drop.OnGa.BG-Trojan
Autorun.xfd.1	Taterf.B.5	Autorun.xfc.1
Autorun.K	Autorun.qmd	Autorun.rhy

2.2 Worm Description

2.2.1. Autorun..xfd.1-worm

Virus:	Worm/Autorun.xfd.1
Date discovered:	09/02/2009
Type:	Worm
In the wild:	Yes
Reported Infections:	Low to medium
Distribution Potential:	Low to medium
Damage Potential:	Low to medium
Static file:	Yes
File size:	106.295 Bytes
MD5 checksum:	8cec5723623ef9fb5be5ff26a2d1c338
IVDF version:	7.01.01.248 - Mon, 09 Feb 2009 17:48

(GMT+1)

2.2.1. AutoIt.X – Worm

Virus:	Worm/AutoIt.X
Date discovered:	10/04/2008
Type:	Worm
In the wild:	Yes
Reported Infections:	Medium
Distribution Potential:	Medium
Damage Potential:	Medium
Static file:	Yes
File size:	617.473 Bytes
MD5 checksum:	3adfe5101e736d996b27b5d547909477
IVDF version:	7.00.03.144 - Thu, 10 Apr 2008 11:00 (GMT+1)

Side effects:

- Downloads malicious files
- Drops malicious files
- Lowers security settings
- Registry modification

Files It copies itself to the following locations:

- %WINDIR%\regsvr.exe
- %SYSDIR%\svchost .exe
- %SYSDIR%\regsvr.exe
- *%drive%*\regsvr.exe

The following files are created:

– %SYSDIR%\setup.ini

– *%drive%*\autorun.inf This is a non malicious text file with the following content:

- *%code that runs malware%*

– %SYSDIR%\28463\svchost.exe Further investigation pointed out that this file is malware, too. *Detected as: TR/Spy.Ardamax.J*

– %WINDIR%\Tasks\At1.job

– %SYSDIR%\28463\svchost.001

2.2.2. Autorun.K – worm

Virus:	Worm/Autorun.K
Date discovered:	26/07/2007
Type:	Worm
In the wild:	Yes
Reported Infections:	Low to medium
Distribution Potential:	Low to medium
Damage Potential:	Medium
Static file:	Yes
File size:	287.044 Bytes
MD5 checksum:	8520bc26d1e51c6a0d7a475def6e69ef
IVDF version:	6.39.00.189 - Thu, 26 Jul 2007 19:19 (GMT+1)

Side effects:

- Downloads malicious files
- Drops malicious files
- Registry modification

Files It copies itself to the following locations:

- %WINDIR%\hinhem.scr
- %SYSDIR%\scvhost.exe
- %WINDIR%\scvhost.exe
- %SYSDIR%\blastclnnn.exe
- *%drive%*\New Folder.exe
- *%drive%*\svchost.exe

The following files are created:

– %SYSDIR%\autorun.ini

– %SYSDIR%\setting.ini

– *%drive%*\autorun.inf This is a non malicious text file with the following content:

- *%code that runs malware%*

– %WINDIR%\Tasks\At1.job

2.2.3. Autorun.rhv-Worm

Virus:	Worm/Autorun.rhv
Date discovered:	27/10/2008
Type:	Worm
In the wild:	Yes
Reported Infections:	Low to medium
Distribution Potential:	Low to medium
Damage Potential:	Medium
Static file:	Yes
File size:	117.248 Bytes
MD5 checksum:	9c93abfe2af88dd661a960b31809912d
IVDF version:	7.01.00.04 - Mon, 27 Oct 2008 17:08 (GMT+1)

Side effects:

- Drops malicious files
- Registry modification
- Third party control

Files It copies itself to the following location:

- *%drive%*\SYSTEM\%SID%\system.exe

The following files are created:

– C:\SYSTEM\%SID%\Desktop.ini

– *%drive%*\autorun.inf This is a non malicious text file with the following content:

- *%code that runs malware%*

2.2.4. *Mytob.HT-worm*

Virus:	Worm/Mytob.HT
Date discovered:	27/06/2005
Type:	Worm
In the wild:	Yes
Reported Infections:	Medium
Distribution Potential:	Medium
Damage Potential:	Medium
Static file:	Yes
File size:	66.937 Bytes
MD5 checksum:	e1fb8181d12248f0b633fcdda38141f9
IVDF version:	6.31.00.110 - Mon, 27 Jun 2005 09:58 (GMT+1)

Side effects:

- Blocks access to certain websites
- Blocks access to security websites
- Drops malicious files
- Uses its own Email engine
- Lowers security settings
- Registry modification
- Third party control

Files It copies itself to the following location:

- %SYSDIR%\ctech.exe

It overwrites a file.

– %SYSDIR%\drivers\etc\hosts

2.2.5. *Taterf.B.5-worm*

Virus:	Worm/Taterf.B.5
Date discovered:	06/02/2009
Type:	Worm
In the wild:	Yes
Reported Infections:	Low to medium
Distribution Potential:	Low to medium
Damage Potential:	Low to medium
Static file:	Yes
File size:	106.581 Bytes
MD5 checksum:	308e6da43f37daec6535b3d98681131f
IVDF version:	7.01.01.238 - Fri, 06 Feb 2009 14:47 (GMT+1)

Side effects:

- Downloads a malicious file
- Drops malicious files
- Lowers security settings
- Registry modification

Files It copies itself to the following locations:

- *%drive%*\yi9.exe
- %SYSDIR%\uret463.exe

The following files are created:

– *%drive%*\autorun.inf This is a non malicious text file with the following content:

- *%code that runs malware%*

– %SYSDIR%\lhgjyit0.dll Further investigation pointed out that this file is malware, too. Detected as: Worm/Taterf.B.7

2.2.6. *Autorun.xfd.1-worm*

Virus:	Worm/Autorun.xfd.1
Date discovered:	09/02/2009
Type:	Worm
In the wild:	Yes
Reported Infections:	Low to medium
Distribution Potential:	Low to medium
Damage Potential:	Low to medium
Static file:	Yes
File size:	106.295 Bytes
MD5 checksum:	8cec5723623ef9fb5be5ff26a2d1c338
IVDF version:	7.01.01.248 - Mon, 09 Feb 2009 17:48 (GMT+1)

Side effects:

- Downloads a malicious file
- Drops malicious files
- Lowers security settings
- Registry modification

Files It copies itself to the following locations:

- %SYSDIR%\urretnd.exe
- *%drive%*\ioockw.bat

It deletes the initially executed copy of itself.

The following files are created:

– *%drive%*\autorun.inf This is a non malicious text file with the following content:

- *%code that runs malware%*

– %SYSDIR%\optyhww0.dll Further investigation pointed out that this file is malware, too. Detected as: TR/Crypt.XPACK.Gen

2.3 Sources of Worm Attacks

The main source these days is through an internet connection. People have broadband connections of internet and are online all the time browsing through different

websites that makes the possibility of getting infected by a worm even higher.

The second reason is through email attachments. A person might send you a legitimate file but if the computer of that person is already infected by a worm than chances are that file is also carrying those worms. Once you download and open that file BAM! You are too infected by a worm.

Another way of getting infected of worm is through file sharing, peer-to-peer or instant messaging. A file is sent to you by a person you know very well. That person may not even have an intention of infecting your computer but his or her computer might be infected which will bring that worm into your computer as well once you accept that file. [4] [6]

2.4. Functions of Worms

A. Uses a compromised machine to spread through instant messaging, mails, sharing etc.

B. It discloses private or sensitive information to the hacker or displays it all over the internet.

C. Changes your settings, wallpapers etc

D. Deletes files and folders of your hard drive without the administrator knowing about it.

E. Causes software instability making the software showing errors whenever opened, hanging of software or closing down without any reason.

F. Your computer becomes really slow making processing really hard. [7]

3. SNORT Tool

A few basic concepts about Snort are given below. It operates in four modes.

A. *Sniffer mode*, which simply reads the packets off of the network and displays them for you in a continuous stream on the console (screen). Here, it acts like tcp dump.

B. *Packet Logger mode*, which logs the packets to disk. Here all the data is logged and post-processed to look for anomalous activity.

C. *Network Intrusion Detection System (NIDS) mode*, the most complex and configurable configuration, which allows Snort to analyze network traffic for matches against a user-defined rule set and performs several actions based upon what it sees.

D. *Inline mode*, which obtains packets from ip tables instead of from libpcap and then causes ip tables to drop or pass packets based on Snort rules that use inline-specific rule types.

Snort uses a simple, lightweight rules description language that is flexible and quite powerful. There are a number of simple guidelines to remember when developing Snort rules. Most Snort rules are written in a single line. This was required in versions prior to 1.8. In current versions of Snort, rules may span multiple lines by adding a backslash \ to the end of the line. Snort rules are divided into two logical sections, the rule header and the rule options. The rule header contains the rule's action, protocol, source and destination IP addresses and net masks, and the source and destination ports information. The rule option section contains alert messages and information on which parts of the packet should be inspected to determine if the rule action should be taken. [8] [9]

3.1. SNORT Plug-Ins:

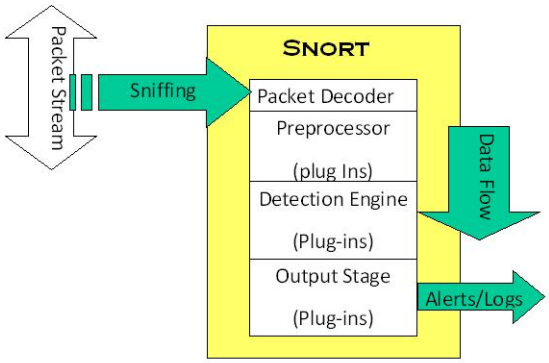

Fig 2: SNORT Architecture

A. Preprocessor

Packets are examined/ manipulated before being handed to the detection engine

B. Detection

Perform single, simple tests on a single aspect/field of the packet

C. Output

Report results from the other plug-ins

3.2. Example

alert tcp $EXTERNAL_NET 27374 -> $HOME_NET any (msg:"BACKDOOR subseven 22"; flags: A+; content: "|0d0a5b52504c5d3030320d0a|"; reference:arachnids,485; reference:url,www.hackfix.org/subseven/; sid:103; classtype:misc-activity; rev:4;)

- **alert** action to take; also **log, pass, activate, dynamic**
- **tcp** protocol; also **udp, icmp, ip**
- **$EXTERNAL_NET** source address; this is a variable – specific IP is ok
- **27374** source port; also **any**, negation (**!21**), range (**1:1024**)
- -> direction; best not to change this, although <> is allowed
- **$HOME_NET** destination address; this is also a variable here
- **any** destination port
- **msg:"BACKDOOR subseven 22";** message to appear in logs
- **flags: A+;** tcp flags; many options, like SA, SA+, !R, SF*
- **content: "|0d0…0a|";** binary data to check in packet; content without | (pipe) characters do simple content matches
- **reference…;** where to go to look for background on this rule
- **sid:103;** rule identifier
- **classtype: misc-activity;** rule type; many others
- **rev:4;** rule revision number
- Other rule options possible, like **offset, depth, nocase.**

4. Implementation

In previous model, there is no detection of infectious packet of specific files. In this proposed work the entire work have been divided into two phases. During the first phase of model, the database of all the worms is stored along with the locations of their possible existence. Now the worm detection code is made to run in order to see whether there are any worms that exist in the locations known. If found any, then those worms would be deleted instantly. But there is a possibility that the worms that are identified are exists but not in the locations that are stored in the database. So, in the second phase, for those worms to be detected, the method of analyzing the packet data in employed. Here the content in the packets are compared with the data content that is obtained from the snort rules. Since snort rules contain the data that helps us to know whether the data in the packet is infectious or not, the comparison with the snort rules data and packet data helps us to detect whether the packet that is getting transferred over the network is malicious packet or not . Hence by the end of two phases, the system is ensured security from worms and infectious packet data. [5]

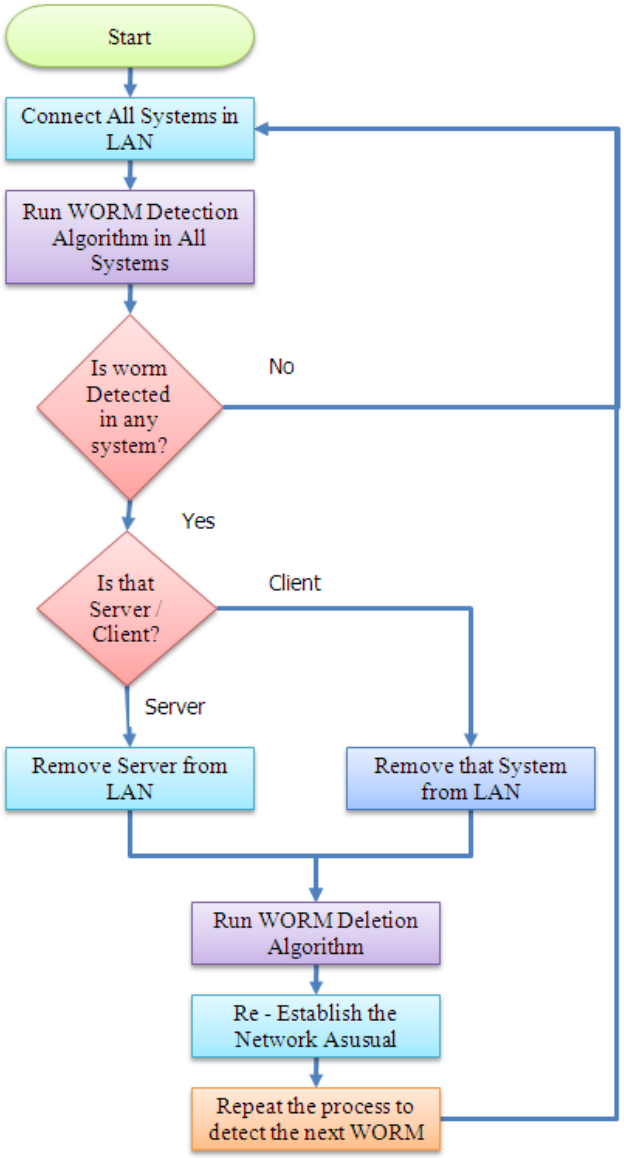

Figure 3: Flow Diagram for Secured Model

4. Performance Criteria

In any proposal, it's important to analyze the Performance with respect to Scalability and Time factor.

4.1. Scalability

As far as our project is concerned, our technique is not limited to certain systems. Hence, we can say that our **Scalability is Unlimited**. The only reason for this is we have developed a code that is made to run in a single

system. Code is no way connected to multiple systems although there is prior requirement that the systems must be connected in a Local Area Network. Hence it can be 3, 30,300 or 'n' number of systems that can be utilized. Regardless of number of systems used, the performance is analyzed basing on single system since the code is associated with a single system

4.2. Time Factor

In our project, there are 2 main phases.
In the first phase, we have developed a database storing the information of worms that exist today. We made sure that the system is free from those worms. For this, the code is made to run and it takes around 2 sec for the execution and make sure that no worms are present.

No. of Locations traversed and searched for worm = 65 (can be increased on updating)

Time taken for traversing these locations = 2 sec

Time taken for each location = 2/65 = 0.0307 sec

In the second phase, we analyzed the content flowing in the network packets and compared those contents with the content present in the Snort Intrusion Detection System rules in order to detect whether the packet is Infectious or not. Using snort there are nearly 47 rule files to be matched with the packet content. At a time, we take 10 packets in order to test the performance.

No. of Snort rule files considered = 47

No. of packets considered for each run = 10

Since it's a combination of two data retrievals, it's not possible to extract the exact time for each retrieval, either snort or packet data (30 sec approx). There is a chance to reduce the time factor that is obtained now by reducing the number of packets considered at a time or taking main snort rules like ftp, tftp , deleted rules etc . However the time taken depends on the Hardware configuration of the system on which the code is made to run

Table 1: Sample Test Cases of Detected WORMS

S.NO.	Condition	Input	Expected Output	Obtained Output	Result
1	Worm Detection by Database Verification (when no worm is present)	Run code connecting the Java with Access Database	No file found	No file found	SUCCESSFUL
2	Worm Detection by Database Verification (when antiv.exe worm is present)	Run code connecting the Java with Access Database	File Deleted antiv.exe Path – C:\Windows\antiv.exe	File Deleted antiv.exe Path – C:\Windows\antiv.exe	SUCCESSFUL
3	Content Retrieval From Snort Rules	Run code by placing the rules in correct path	Content from each snort rules should be retrieved line by line	Content from each snort rules is retrieved line by line	SUCCESSFUL
4	Content Retrieval from packets	Run code after enabling packet flow in LAN	Content from each packet is to be retrieved	Content from each packet is retrieved	SUCCESSFUL
5	Matching of Snort and packet content (when no Match is found)	Run the code that compares both the strings	No match found. No packet content should be found matched with the snort rule content	No match found. No packet content is found matched with the snort rule content	SUCCESSFUL

5. Further Works

In our paper, we successfully deleted the worms stored in database and detected the infectious packets. In future, the infectious packets that are found flowing through the network can be discarded, thus providing total security to the systems. The database that contains worm information can be updated as and when new locations of worms are known so that better security is assured. If any better Intrusion Detection system is introduced, that can be employed in this project which would yield better results

6. Conclusion

The main motto behind our paper is to protect systems that are connected through LAN from getting affected by any worm. It is necessary to detect the worm once it affects any system instantly, thereby ensuring protection to other systems by stopping the migration of worm to other systems. In the first phase, we maintained the worm information in an Access Database and successfully deleted those worms whose information is present in the Database. For those that are not present in the database, we dynamically retrieved the contents of the packets flowing in the network. We checked whether that content is

Infectious or not using the Snort Intrusion Detection System rules. By this, we developed a mechanism of detecting the infectious packets flowing in the Network. Since every organization employs many computers connected through LAN and since security for the systems is the biggest concern today, we wish that our work would definitely help them to the maximum extent.

Acknowledgments

We owe our tributes to Dr S.C. Satapathy, Head of Department, ANITS for all his support in technical aspects and for his guidance. And also we are thankful to B Tirimula Rao, Associate Professor, ANITS for his support.

References

[1] Sellke, S. H.; Shroff, N. B.; Bagchi, S. (2008) *"Modeling and Automated Containment of Worms"* IEEE Transactions on Dependable and Secure Computing **5** (2): pp 71 – 86

[2] *"ILOVEYOU"* WHoWhatWhereWhenWhy.com. Retrieved 2008-05-26.

[3] Moore, David; Colleen Shannon (2001). "*The Spread of the Code-Red Worm (CRv2)*". CAIDA Analysis

[4] Markoff, John (2009-01-22). "*Worm Infects Millions of Computers Worldwide*". New York Times. Retrieved 2009-04-23.

[5] J. Wu, S. Vangala, L. Gao and K. Kwiat. *"An Effective Architecture and Algorithm for Detecting worms with various scan Techniques*" in NDSS Symposium, 2004

[6] F. Buchholz, T. Daniels, J. Early, R. Gopalakrishna, R. Gorman, B. Kuperman,S. Nystrom, A. Schroll, and A. Smith. *"Digging for worms, fishing for answers"*. In Proceedings of the 18th Annual Computer Security Applications Conference, 2002.

[7] Thomas Chen, Jean-Marc Robert (2004). *"The Evolution of Viruses and Worms"*. Retrieved 2009-02-16.Northcutt, S. (2002) *Network Intrusion Detection*, New Riders Publishers.

[8] Sourcefire Inc., Roesch, M. and Green, C. (2006) *"SNORT users manual – SNORT SNORT Release: 2.6.0"*, Available at: http://www. SNORT.org

About Authors

Mr. Y V Srinivasa Murthy received his Masters' in Computer Science & Technology from GITAM University. He is currently working as Assistant Professor in Anil Neerukonda Institute of Technology & Sciences, Visakhapatnam. He has an excellent command on programming and presented many papers internationally. He is the member of CSI, ISTE and many other reputed professional bodies.

Mr. G. Jagadish received his Masters' in Computer Networks from Andhra University. He is currently working as Assistant Professor in Anil Neerukonda Institute of Technology & Sciences, Visakhapatnam.

Ms. K Mrunalini received her Masters' in Computer Science & Technology from GITAM University. She is currently working as Assistant Professor in Anil Neerukonda Institute of Technology & Sciences, Visakhapatnam.

Mr. Kakarla Siva working as Senior Consultant in HCL Technologies, Hyderabad. He has excellent on windows, VMware and Citrix Technologies. Having 8 years of experience in IT Industry.

Mr. Polisetti Satyanarayana, pursuing his B. Tech (CSE) in Anil Neerukonda Institute of Technology and Sciences. He is the topper in his batch. He has interest in programming and he got first prize in the coding event conducted in local Engineering College. He is very enthusiastic in solving emerging problems.

.

Mr. Nikhil Raj Kumar, currently pursuing his B. Tech (CSE) in Anil Neerukonda Institute of Technology and Sciences. He is very interested in programming & solving the logics. He won first prize in the coding event conducted at local Engineering College.