www.ijecs.in

International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 4 April, 2014 Page No. 5441-5446

Netflow Traffic Analyzer For worm detection- A Survey

Manish Khule[1]***, Megha Singh***[2]***, Deepak Kulhare***[3]

[1]Student,C.S.E,Ciit,Indore,
mannykhule@gmail.com
[2]Assistant Professor, C.S.E, Ciit
[3]Assistant Professor, C.S.E, Ciit, Indore,

Abstract: ***Due to easy access and requirement of the Internet make it more popular for research and information sharing. Because of this feature a malicious codes are also easily exchange. A worm (malicious codes) can disturb network and normal network operation. Internet worms are causes significant worldwide disruption, a huge number of infected hosts generate traffic, which will impact the performance of the internet. Therefore this is one of the areas where researchers are concentrating to find effective detection system, which will presence the worms and reduce the worm's spread. This paper deals with a classified study of most important and commonly used methods for detecting internet worms using Netflow, which can help network managers to monitor suspect Internet worm's activities by analyzing the source data from the router***

Keywords: Internet worms, anomaly detection, network intrusion detection, Netflow**.**

1. Introduction

The Internet is persistently threatened by many, types of attacks such as viruses, and worms. A worm is aself propagating program that infects other hosts based on a known vulnerability in network hosts. In contrast, a virus is a piece of code attached to another executable program, which requires human action to propagate. A major challenge in networking is how to detect new worms and viruses in the early stages of propagation in a computationally efficient manner.

The impact of worms and viruses on the Internet include delays due to congestion, extensive waste of network bandwidth, as well as corruption of user's computers and data. Furthermore, viruses and worms can carry software that]enables attackers to gain access to the personal informa-tion of users. In addition, recent worms are capable of launching distributed denial-of-service (DDoS) attacks against other hosts [1].

During the past 20 years, thousands of different worms have been developed. Some of these worms have caused huge disruption to global networks. The most notable worms include Morris, Code Red and Code Red II, Nimda, Slapper, and Sapphire/Slammer worms, and recently, SoBig.F, MS Blast, and Mydoom. From the first worm that was released in 1988 (the Morris worm), the area of Internet worm detection has been a significant research problem. In order to understand the worm threat, it is necessary tounderstand the various types of worms, payloads, and attackers. Taxonomy of the various possible worms, payloads, and attackers as an initial guide to plausible defenses.

This taxonomy is necessarily incomplete, simply because new tactics, payloads, and attackers may arise. This taxono-my is based on several factors: target discovery, carrier, acti-vation, payloads, and attackers. Target discovery represents the mechanism by which a worm discovers new targets to infect. The carrier is the mechanism the worm uses to trans-mit itself onto the target [5-9]. Activation is the mechanism by which the worm's code begins operating on the target.

Payloads are the various non-propagating routines a worm may use to accomplish the author's goal. Finally, the various possible attackers have different motives and would there-fore utilize different payloads.

In addition, it is important to note that worms needn't be confined to a single type within each category. Some of the most successful worms are multi-modal, employing multiple means of target discovery, carrier, payload, etc, where the combination enables the worm to surpass defenses (no matter how effective) that address only a single type of worm. In this section, summary of previous approaches to worm detection has been done [6-10]. Usually, the detection methods are based on the feature of the Internet worm such as abnormal network traffic, content comparison, process scanning and detecting network connection.

The current detection method for the Internet Worm two general categories: Signature-based Detection and Ano-maly Detection. Signature-based detection is based on defin-ing malicious patterns that the system has to detect. Signa-ture-based detection suffers from the problem that it requires a signature of each attack be known. In contrast, anomaly

detection differs by constructing a profile of normal behaviors or activities on the network, and then looking for activities that do not fit the normal profile. Since not all the abnormal activities in the network are suspicious, anomaly detection has the problem of raising false alarms when it encounters normal traffic.

The Connection-Oriented detection method Ob-serves the number of connection with the target host and checks the connection behavior. Conditions are the core of Connection-Oriented detection method.

The Internet worms diffuse quickly to infect servers, destroy information, embed backdoor, and consume resource from network bandwidth In the trap oriented detection me-thod, the surveillance area can be separated into single host and the several network segments on the Internet. In this method, the accuracy is quite high and it is easy to differen-tiate between the normal and abnormal traffic. Therefore, the nodes have to collect the network flows (information which is produced from router), for finding abnormal traffic.[10].

Definition 1 An Internet worm is a piece of self-replicating code that does its replication over the Internet, i.e. the target is accessed using a layer 3 or layer 4 protocol, typically TCP or UDP. In order to propagate, the host on which the worm code is executed (called infecting host or infected host) contacts an other host (the target host) over the Internet, replicates its code onto the target and triggers execution (infection) of its code on the target. We will sometimes call the running copy of the worm code on the target a child or child instance of the worm. All hosts that have the same number of infection steps from the initially infected host(s) are called an (infection) gen-eration. 20 3 Worm Traffic While in principle worms that propagate using the ICMP protocol or using raw IP in some fashion (i.e. where the protocol field in the IP header is ig-nored or not used) are possible, we are not aware of any worms that use these means to propagate. A distinction that is sometimes made is between the notion of a worm and a virus. The idea is usually that a worm can propagate without user interaction, while a virus cannot. We do not make this distinc-tion. In a sense we allow the user to be part of the execution environment. In this way our definition includes email worms and other application worms that require a user on the r e-mote host to open an email attachment, for example, in order to trigger worm code execution.

Definition 2 A fast Internet worm is an Internet worm that infects most of the vulnerable (reachable) host population in less than a day. We are aware that this definition is not too precise. Nonetheless we are not aware of a better one.

Definition 3 The initial outbreak (or just outbreak for short) of a fast Internet worm is the time from its first infection over the Internet until it reaches saturation. Saturation is typically reached when around 90% of the vulnerable host population that is reachable has been infected. Again, the term saturation is not too well defined. Intuitively it is reached when the tar-get selection strategy of the worm produces mostly unsuc-cessful selections, since most vulnerable hosts have already been infected.

2. GENERAL WORM M ECHANISMS

Every Internet worm has to have a certain minimal functionality in order to be viable:

- A worm has to be able to identify possible infection targets.
- A worm has to be able to transfer its code to a selected target.
- A worm has to be able to induce a vulnerable target to run the transferred worm code.

Infection Mechanisms

A worm should be able to identify already infected targets and refrain from re-infecting them. Interestingly, the first three requirements are already enough. The fourth merely improves efficiency. Also, if a service on the target system is capable and willing to propagate the worm without having its security compromised, then a worm can do without any kind of system compromise at all. A compromise of the target system to some degree is customary nonetheless, especially when some other purpose, like espionage, sending of spam or attacks on other systems is intended.[1-10] A second rea-son for target system compromise is that many worms use security vulnerabilities to obtain resources on the target sys-tem. The advantage is that in this way the basic execution services of the target system become available to the worm and any functionality its designer wants can be easily im-plemented. The typical worm uses a propagation mechanism that works like this:

1. Select a potential target
2. Attempt to contact the target
3.Compromise the targets security in some way to obtain the resources to transfer and execute a copy of itself.
4. If more infections are desired, goto step 1
5. Do damage on the local machine or do damage somewhere else using the local machine

The last step is optional and can also be done earlier. However, in order for a worm to propagate as fast as possible, it is a sound design choice to not impair the functionality of an infected host until the worm has completed most or all of its intended propagation activity from that host. In addition, the damage may be done later to delay the discovery of the worm or in order to allow coordinated attacks from several infection generations.Note that we also regard data collection activities, such as looking for passwords or credit card numbers, as causing“damage”.

1.Target Selection Mechanism- The target identification and selection mechanism a worm uses, since target selection has by far the largest influence on the actual worm traffic seen in the Internet during an outbreak. The reason is that, while the target address may or may not be assigned to a host and if there is a host, this host may or may not be vul-nerable, the worm code has to select targets and then try to contact them. These connection attempts, also called scan traffic is the most visible sign of a fast Internet worm in its main propagation phase.

2.Random Scanning-Perhaps the most simple target selection strategy is purely random scanning. For this, the target selection code usually includes a Pseudo Random Number Generator (PRNG) or uses an OS service with this functionality. Infection targets are then selected by generating a 32 bit random number and using that as the target IP address. In a more advanced set-ting, ranges that do not contain normal hosts, such as mult i-cast-addresses, can be excluded. Care needs to be taken, that the random target selection is implemented correctly. Interes-tingly, many worm writers seem to get this wrong Mistakes include constant PRNG seeding after propagation, use of inferior PRNGs with non-even value distribution and even PRNGs that cannot generate all output values and hence miss many possible targets.

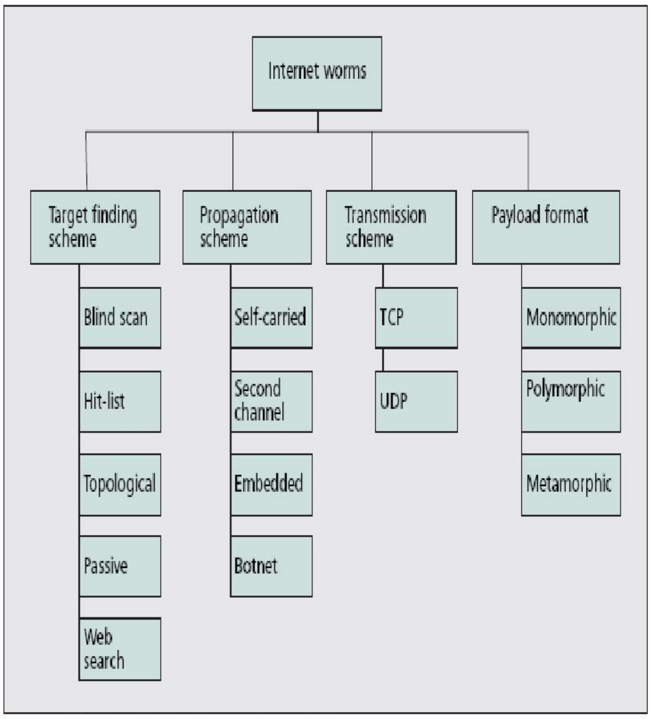

Figure 1. *Categorization of worm characteristics.*

3. LOCAL PREFERENTIAL SCANNING

The Pure random scanning works reasonably well, but one disadvantage is that it does not take advantage of the better network connectivity to hosts in the same LAN or otherwise in close proximity. Local-preferential scanning is very similar to random scanning, but it dedicates a portion of the scan activity to addresses in the same subnet the attacking host is in. Typical implementations have preferences for the /24 sub-net and the /16 subnet of the attacking host. One way to im-plement this type of strategy is to randomly scan in more often in the local /16 subnet, but to scan the local /24 subnet fully. The latter can be done in a simple, linear fashion, al-though this may trigger IDS and/or IPS system sensors.[10] Local-preferential scanning has several advantages. One is that the probability of actually finding hosts with addresses close to the attackers IP address is usually far higher than for randomly selected addresses. After all, the local subnet con-tains at least one host already, namely the infected host. This means that it is not an unused subnet. The second advantage is that the traffic over the Internet access and backbone net-works is reduced. Pure random scanners run the risk of over-loading the Internet access connection and thereby hindering their own propagation. A further advantage is that the net-work latency to hosts in close proximity is lower, leading to faster scanning and infection performance.

4. HITLIST SCANNING

A completely different approach to random scanning is hit list scanning. To implement this strategy, the worm-designer precomputes a list of vulnerable targets. This list is then included in the worm when it is deployed. The worm then not only propagates its own code, but also parts of the hitlist to be used by the respective child instance. Propagation schemes with some degree of redundancy are possible. For example, each so far unused target address could be propa-gated to two or several child instances of the worm, so that if a child instance cannot work through its list fragment com-pletely, some other child instance may still be successful. With this type of redundancy the individual copies should be worked through in different orders to maximize propagation speed. The use of a hitlist scanner for the full vulnerable population for a specific exploit only makes sense if this population is relatively small. Otherwise the Worm Traf-fic transfer of the hitlist will slow down the worm consider a-bly. A second concern is that the hitlist needs to be obtained in a way that does not arouse suspicion. Otherwise the worm could find a situation were the potential targets have already been warned before its initial propagation. A typical use of hitlist scanning is to have a relatively small hitlist of very attractive targets, e.g. hosts with high bandwidth or host that are geographically well placed. The worm then does its ini-tial propagation with a hitlist strategy and then changes over to another strategy after one or a few infection generations, e.g. random scanning.

5. TOLOGICAL SCANNING

Topological scanning bears some resemblance to hitlist scanning. However, the information about potential targets is not precomputed, but instead extracted from the data available on the local host. Possible sources of IP addresses are ARP caches, contact lists of P2P applications, open Internet connections, browser caches, address books of any kind and oth-er sources. Host names and URLs can also be used since they can be converted to IP addresses by DNS lookup. It should be noted that worms that do DNS lookup will generally be quite slow and likely not qualify as fast Internet worms ac-cording to our definition. One primary example of topologi-cal worms are email worms. Although they are not Internet worms by our definition, they represent a very important class of application layer worms. Another class of application layer worms are IM (Instant Messaging) worms, that have also been observed in the wild. P2P file sharing layer worms, but so far no P2P worms have been observed as to our knowledge.Port CharacteristicsScan traffic of a fast Internet worm has some limitations on how source and target ports can be selected. These are differ-ent for TCP and UDP scan traffic. We will now discuss the different possibilities.

5.1 TCP: Source Port

Scan traffic of a fast Internet worm has some limitations on In ordinary TCP traffic, the source port for the connection initiating host, i.e. the host that sends out the initial SYN packet, is chosen at random by the network stack from a port range unlikely to be used as server ports. Each concurrent connection gets its own source port, so that answering traffic can be identified by the port it is sent to. It is possible to drop this requirement and match answering packets by remote IP address and port. This is, for example, done in servers that accept multiple connections on a single port, such as web servers. For a worm, it would be possible to use a static source port and match theanswering traffic by remote IP address. However, this causes additional effort and does not have any real benefit. It also prevents the worm from using the normal network stack, since the normal, OS integrated network stack cannot do this type of matching.

5.2 TCP: Destination Port

The primary limitation for destination port selection in a worm is the exploit used. If an exploit works only on a specif-ic port, then all attack traffic has to be addressed to that port. In addition, the connection initiating SYN packet in a TCP connection is unable to transport data. Even if a port independent exploit was possible, the initial SYN would have to be sent to a port where the remote system sends an answer. With variable ports, the worm would need to do a port scan in order to find such an open port. This scan would slow the worm down significantly. In addition we are not aware of any TCP exploits that can be used on a larger range of target ports. For these reasons a worm using one or more TCP based exploits will likely target one or a small number of TCP ports on the target system

5.3 UDP: Source Port

Since UDP is connectionless, UDP based exploits can be and usually are single-packet exploits. This means the attacking host sends a single packet to the target host and is then either contacted back by the successfully executed exploit code or has to do a second polling step. For both options, the UDP source port is immaterial and can be chosen in an arbitrary fashion.

5.4 UDP: Destination Port

As in the case of TCP, the target port for an UDP exploit depends on the actual nature of the exploit. If the vulnerability is present in a service running on a specific port, the same rationale as for TCP destination ports applies and the target port will be fixed. Unlike TCP, UDP permits transfer of data in the first packet sent. This allows exploit code to be sent torandom destination ports, since establishing a connection is not needed. In order for this to work, the vulnerability needs to be in a service that processes all UDP payloads, such as a firewall or a proxy.

shows internet worm detetion algoriths.these algorithms are allready implemnted.in this paper we are work on netflow method.

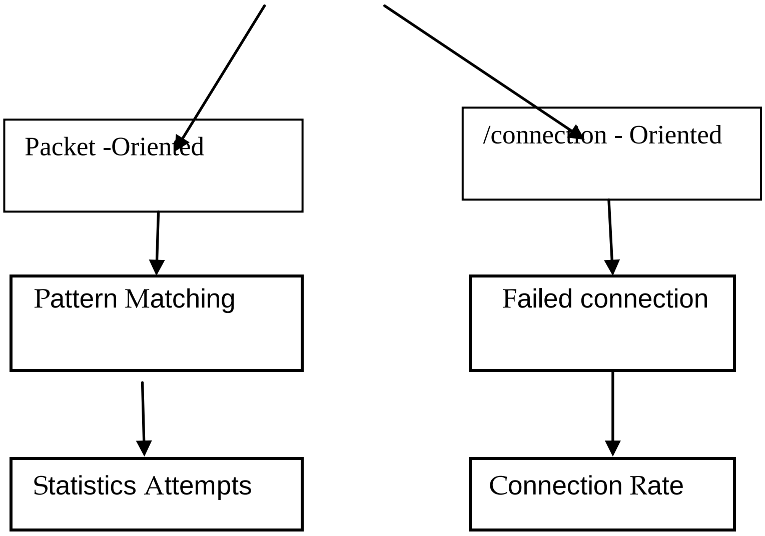

Fig.2 Detection Algorithm

Statistics: The packet flow is statistically analyzed. This analysis can be based on all packets or only on a selection. Connection-oriented: Connection-oriented detection methods interpret packets as part of a connection and base their analysis on the connections as a whole.

Failed attempts: The number of failed connection attempts is counted and compared to a fixed or dynamic threshold. Exceeding this threshold indicates an attack.

Connection rate: The number of connection attempts (successful and unsuccessful ones) is counted and compared to a threshold. Exceeding this threshold indicates an attack.

Packet-oriented connection-oriented : A lot of computer worms and viruses have been rapidly spreading all over the world in the last years.

6. METHODS FOR DETECTING INTERNET WORMS USING NETFLOW

A Netflow has been defined in many ways. The 7 tuple key, where a flow is defined as a unidirectional se-quence of packets with the following 7 values:

1. Source IP
2. Destination IP
3. Source port for UDP or TCP
4. Destination port for UDP or TCP
5. IP protocol
6. Ingress Interface
7. IP type of service

Using this Netflow information different systems are design for detecting internet worm. We found out of these the following systems are more effective:-
1. Defending against Internet Worm- Infestation
2. Flow WorM system

Defending against Internet Worm-Infestation:

To deal with Internet worms, a pro-active respond-ing scheme consisting of detecting, blocking and notifying operations. The main goal of this scheme is to keep the net-work as healthy as possible during the flooding period of Internet worms. Internet worms that generate extreme high volume of probing packets and pose threats to normal net-work operations .The block diagram is shown in fig-ure 3

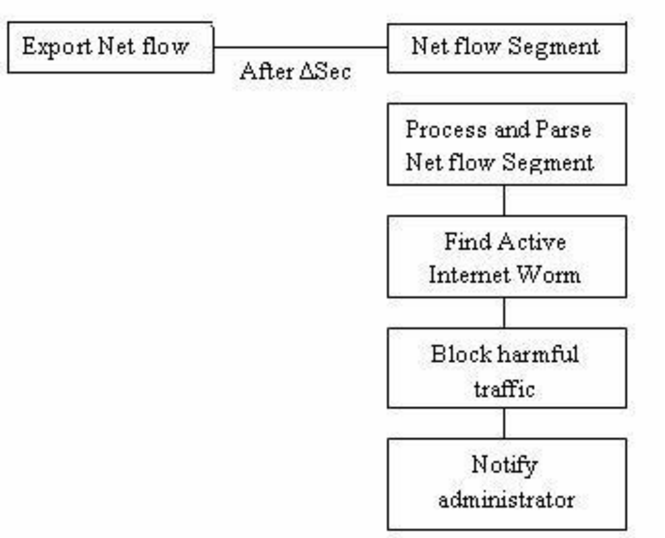

Figure 3.Internet Worms Defending System

1. A network flow is defined as a unidirectional sequence of packets between the source and destination endpoints. contains IP addresses of endpoints, service port numbers, IP protocol type, and the input interface identifier. Router ex-port NetFlow information. Every t seconds, the NetFlow data is collected as a NetFlow segment.[8]

2. Detecting operation identify wormlike behaviors, which are harmful to the network (e.g. sending a flood of packets or high frequent connection attempts). Netflow solution tech-niques is used to monitor and analyze the network traffic ,the NetFlow solution is chosen because it provides a simple and convenient way to obtain network flow information without the needs to add additional passive monitoring devices to the network.

3. After some time period e.g. 10 minutes, proceed and analyze the data to figure out what happened to the network.

4. Blocking operation first try to keep the network as healthy as possible. Then it is to prevent other hosts from being in-fluenced by infected hosts.[5-8] The third one is to make an effort to obey the policy that we should punish bad ones not all other innocent ones.

To keep a healthy network, the access control func-tion on routers and switches are used not only to pr otect the network devices themselves but also to block harmful traffic onto them. There are two cases to be considered in the block-ing operation:

• Hosts inside the network: When the virulent host is inside the network, it is best to isolate the infected host only with affecting other hosts.

• Hosts outside the network: When there are just a few Number of virulent hosts outside the network, then simply blocks traffic from these hosts on the border router.

5. Notifying operation is designed to inform system adminis-trators or persons relative to the security incidents. It can be implemented by sending emails, short messages to pagers or mobile phones.

FloWorM System :

The previous system detects worm, notify the net-work administrator to do further analysis or by coordinating the NMS to automatically process self-defense mechanism.But the problem with the system is that it generate false alarm. FloWorM system [6] can greatly reduce false alarm and can efficiently detect the Internet worm activity. The Functioning of system is shown in figure below:

In FloWorM System NetFlow data is collected from router because traffic information will be recorded on the routers. Then use data from the NetFlow to analyze the flow and generalize four typical abnormal network behaviors, which are briefly discussed belowThe IP address communicates with specific destination ports in order to infect other computer host, the Internet worm will try to increases the number of IP addresses that communicate with other hosts.

• Failure in the network connection is due lack of knowledge of target victim. The port service on the victim host is not prevalence.

• Connecting unassigned IP address use a particular algorithm to produce the target IP addresses.

• The network packet which is produced by the Internet worm will follow the order or content that will launch attack. System work in two stages using different modules:

Stage 1:

The tracking module monitors abnormal flow. Ab-normal flows are marked with the tag while related information are sent to another tracker to keep the record. The suspect tag flow will be sent to Analyzer module. The tracking module work in two steps:
1. Spreading and Scanning
2. Repetitious Pattern

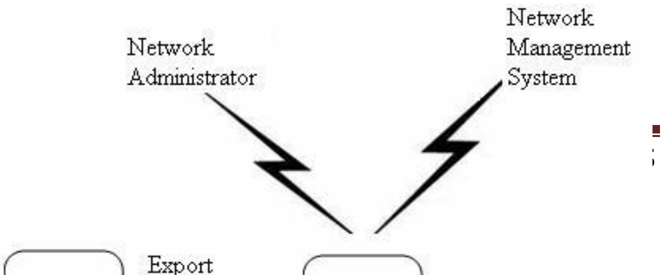

Figure 2.FloWorM System Design

In Spreading and Scanning stage Tracker detect the network worm. For detection of worm it finds out information like:

- Connection with a specific protocol
- DstPort
- No of Connection
- Number of Fail connection

Using this information the tracker find some limit value. If specified connection values cross the limit value such connection is detected as abnormal connection and mark with TAG_SPREAD andTAG_SCAN and send to next tracker. In repetitious pattern ,tracker collects the suspicious flow which is marked with TAG_SPREAD and TAG_SCAN by the Spreading and Scanning for further recognition and detection.

Stage 2:

In 2nd stage Analyzer is used identified the category of the network worm and decide the network worm behavior. TheSignature basedIDS system is used to compare packets b e-tween the normality and the abnormality. Usually, the net-work worm has some specific attack techniques and the in-fected behavior. The infected behavior [2] are in four phases: sending attack packet, connecting to backdoor, exchanging the connection information with the specific host, and sending data by using specific communication port. Suppose A is infected host with the worm and B is infesting host by A. Firstly, A tries to connect to B by using port. Analyzer has to capture this kind of scanning behavior. When the connection is successful, A will try to exploit B and launch the attack. Scanning is the detection method in this system. After scan-ning such connection record is maintain in the system. If coming connection matches with the entry alarm has been produced to inform administrator.

7. CONCLUSION

This paper presents a classified study of internet worm detection system. Defending against Internet Worm-Infestation system is easy to implement but because short holding time false alarm is the major drawback of the system. Another method which described in the paper is FloWorm which provides a high accuracy and it reduces detection rate. In the future, it is also possible to divide the worm connection into two different categories like the connection which impersonate server and the connection which impersonate client.

References

[1] Al-Hammadi, Y.; Leckie, C, ”Anomaly detection for Internet worms”,

[2] Integrated Network Management, 2005, 9th IFIP/IEEE International Symposium on 15-19 May 2005, vol. 2, pp.133-146.

[3] Ellis R. D., Aiken G. J., Attwood S . K., and Tenaglia S ., “A Behavioral Approach to Worm Detec tion”, WORM’04, Washington, DC, USA, 29th October 2004,vol.13, pp.71-79

[4] Schechter S., Jung J., Berger W. A., “Fast Detection of Scanning Worm Infections”, 7th International Symposium on Recent Advance in Intru-sionmDetection (RAID), September 2004, vol.19, pp-17-57.

[5] “Morris (Computer Worm),” retrieved July 2007, http://en. wikipe-dia.org/wiki/Morris_worm

[6] “F-Secure Virus Descriptions: Nimda,” retrieved July 2007, http://www.f-secure.com/v-descs/nimda.shtml, 2001

[7] “CERT” Advisory CA-2001-26 Nimda Worm,” retrieved July 2007, http://www.cert.org/advisories/CA-2001-26.html, 2001.

[8] “F-Secure Computer Virus Information Pages: Slammer,” vol.May, 2005, http://www.f-secure.com/v-descs/mssqlm.shtml

[9]“Sasser Worm Analysis — LURHQ,” May 2005, http://www. lurhq.com/sasser.html

[10] “Secunia — Virus Information — Sasser.G,” May 2013, http:// secunia.com/virus_information/11515/sasser.g