PREVENTION OF WORM AT ROUTER LEVEL FOR PROVIDING SEAMLESS COMMUNICATION IN NETWORK ENVIRONMENT

M. Milton Joe[#1], R.S. Shaji[*2], K. Ashok Kumar[#3]

[#]Assistant Professor, Department of Computer Applications, MAHER – FHS, Meenakshi University,
Chikkarayapuram, Chennai – 69, Tamilnadu, India.
[1]m.miltonjoe@gmail.com
[3]jas_indians@yahoo.co.in
[*] Professor, Department of IT, Noorul Islam University, Nagercoil, Tamilnadu, India.
[2]shajiswaram@yahoo.com

***Abstract* - Worm is the major hurdle, which restricts the comfortable communication in any networks. Worm is a malicious software program that destroys the normal communication in the networking systems. Every system must not be vulnerable to avoid infection by worm in the computing networks. Defending against such worm still plays vital role to the network programmers. Various countermeasures have been taken so far, still they all fail to establish a fair communication in the networks, for worm has the automatic propagation nature to propagate from one host to the other host. Previous countermeasures provide worm detection at the host level only, which takes more time in worm detection. This paper suggests an alternative approach to detect the worm at the earlier stage, which means detecting the worm at the router level and forwarding wormless packets to the hosts that are connected in the networks. We have studied the comprehensive characteristics of worm and its propagating nature. We define a novel mechanism to detect the worm that propagates in an automatic fashion among the systems in the networks at the router itself and forward the clean and wormless packets. Our mechanism leads to several metrics, which proves our methodology works better than previous countermeasures. Our results show how our mechanism works and how all the metrics are obtained, when worm is being detected at the router itself. Thus, the proposed worm detection scheme obviously leads to the secure communication in the networks.**

Keywords - Worm, Networking, Propagation, Router, Packet, Secure Computing, Quality of Service (QoS).

I. INTRODUCTION

Worm is also a software program that acts in quite different manner to destroy the normal flow of communication in the networks [1]. As we know certain worms have made considerable damage in the networks so far. These worms include “Code-Red” worm in 2001 [3], “Slammer” worm in 2003 [4], and “Witty”/“Sasser” worms in 2004 [5].These worms are used to infect a large number of computers and recruit them as bots or zombies, which are networked together to form botnets[6].These botnets can be used to:

(1) Launch massive Distributed Denial-of-Service (DDoS) attacks that disrupt the Internet utilities [7],

(2) Access confidential information that can be misused [8] through large-scale traffic sniffing, key logging, identity theft, etc.,

(3) Destroy data that has a high monetary value [9].

(4) Distribute large-scale unsolicited advertisement emails (as spam) or software (as malware).

There are many types of worms but still these worms are mainly classified into two major categories. Those major types are: 1. Unified worm, 2. Non- unified worm [1]. The unified worm always uses an existing file to propagate from one local host to another local host. Another type that is non- unified worm propagates automatically from one local host to another local host without the intervention of the human in the networks. The second type of worm causes more damage, when the system is vulnerable in the communicating network. All the local hosts must be secured to prevent them from worm infection in the networks, for when networking is formed these worms can automatically propagate within the network to make the communication flow out of control. This type of worm is classified as active worm [1] [2], because of its self propagating nature. This worm must be detected to provide secure communication among networking systems and the flow of control must be maintained in a normal way.

Previous countermeasures detect such active worm at the local host level only. When the worm infects one host, it automatically moves to the other systems that are connected in the networks. This approach may not provide security to the network completely, for when worm is detected and removed from a single networking system within a millisecond it will pass on the other one and keep on infects the entire networking. Obviously a new approach is needed to provide secure communication among networking systems. Our proposed novel scheme will lead to secure communication among the networks, because our detection scheme scans and detects the worm at the router level itself and only forwards wormless packets. Thus the flow of control and secure communication will be maintained with all the metrics. In our approach even though the systems connected in networking are vulnerable will not make considerable damage, since worm is detected at the router itself.

II. RELATED WORK

After receiving considerable damage by the worm in the network communication [3] [4] [5], the network programmers have started to take precautions to stop the damage caused by the worm. The earlier works have been done to detect the worm at the following two ways: 1. Network based detection, 2. Host based detection [1] [2]. The network based model could be evaluated in various ways, such as port matching and addressing matching. Host based detection would be done through monitoring and scanning every host. Obviously it is clear, that wormless packet should be sent to the network and security must be maintained. By observing the previous work of detecting active worm that propagate in an automatic nature in social network is detected successfully at every host level. After identifying the worm spread host the corresponding host is blocked from the network [1], which indicates the poor performance in the network. The wormless packet and valid data will be missed by the blocked host. Scanning every packet at every time leads to more time complexity in the network. To overcome from all these limitations a novel approach is ultimately needed. We present a novel mechanism to overcome these limitations along with some more metrics, which would bring the network communication with wormless packet and security constraint is maintained always. Our architecture shows the detailed description of our novel mechanism and its working methodology and how all the metrics are obtained by the proposed scheme.

III. ARCHITECTURE

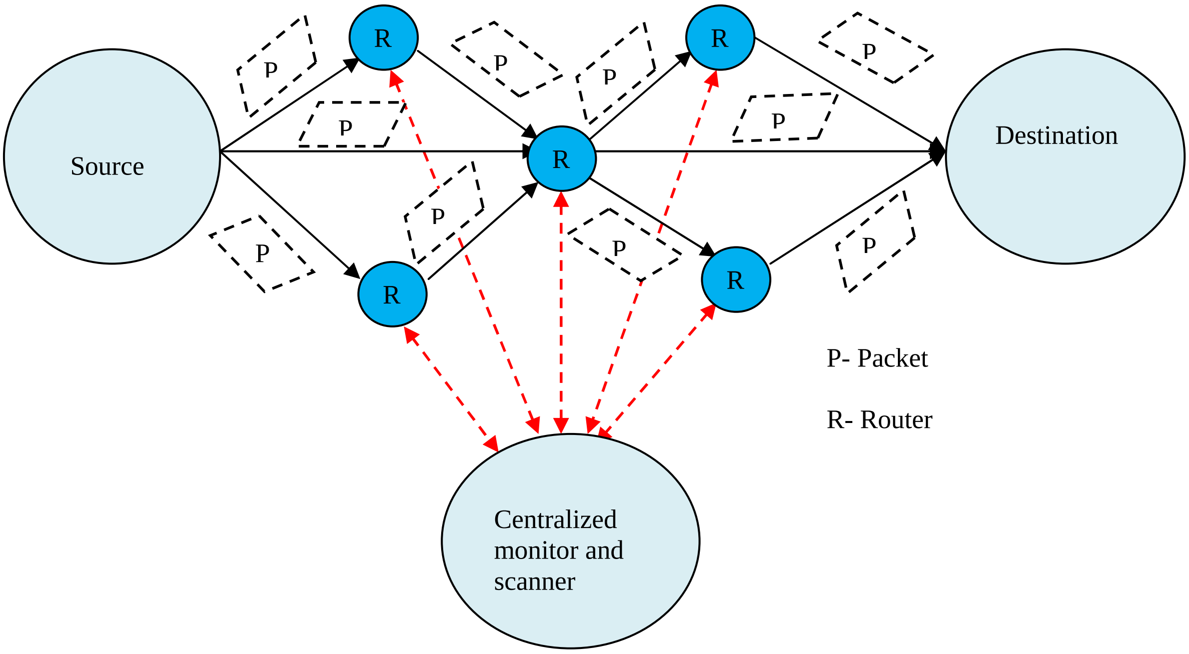


Fig 1 Architecture

Fig 1 illustrates the entire architecture of worm detection at the router level. Any number of hosts can participate in the network and the data should be routed to the correct destination. Data is divided into various chunks and formed as packets. These packets must be routed to the destination from the source. Routers are used to route the packets to the exact destination. As shown in the figure the packets are forwarded to its destination. Router identifies the shortest path and minimum cost path and then routes the packet ahead. The centralized monitor and scanner is used to monitor the packets are forwarded across the router. All the routers are connected with the centralized monitor and scanner. When a packet is forwarded from the source, it moves via the router to its destination. The router checks whether the packet is infected by worm by the matching the values of the packet structure and the routing table information. The packet is sent to the centralized monitor and scanner module, if the packet is infected by worm. The centralized monitor and scanner module checks the worm contained packet and removes the worm by scanning the worm contained bits in the particular packet. The packet is sent back to the router, once the worms are removed from the packet and router does its usual function. That is, the packet is forwarded to the next hop to reach its exact destination. Our novel architecture uses its own packet structure and routing table format, which are specially designed to identify the worm at the router itself. Worm is identified with the values obtained by the packet structure and the routing table.

A. Packet Structure

Every network protocol must have a packet structure for forwarding data in a specified format. Data is divided into small pieces and formed as packets. These packets of data are sent from source to destination and acknowledgement is received from the destination. The packet data could be sent in any order to the destination. Packets are numbered before they are forwarded from the source. Once the packets reached the destination, they could be rearranged in the corresponding original data order by the packet number. Every protocol has different packet structure for forwarding data from source to destination. Similarly, to detect the worm at the router itself our architecture uses different packet structure format. The following table represents the packet format used to detect the worm at the router itself.

Table 1 Packet Structure

◄──────────32 Bits──────────►

<table>
<tr><td>Version</td><td>IHL</td><td colspan="2">Type-of-service</td><td colspan="2">Total length</td></tr>
<tr><td colspan="4">Identification</td><td>Flags</td><td>Fragment offset</td></tr>
<tr><td colspan="2">Time-to-live</td><td colspan="2">Protocol</td><td colspan="2">Header checksum</td></tr>
<tr><td colspan="6">Source address</td></tr>
<tr><td colspan="6">Destination address</td></tr>
<tr><td colspan="6">Options (+ padding)</td></tr>
<tr><td colspan="3">Data (variable)</td><td colspan="3">Binary value</td></tr>
<tr><td colspan="6">Worm flag</td></tr>
</table>

The table 1 represents the packet structure used for worm detection at the router level. Our packet structure consists of the following information. Our packet structure is designed from the IP packet structure.

Version - indicates the version of this IP datagram.

IP Header Length (IHL) - Indicates the datagram header length in 32-bit words.

Type-of-Service - Specifies how a particular upper-layer protocol would like the current datagram to be handled. Datagram can be assigned various levels of importance using this field.

Total Length - Specifies the length of the entire IP packet, including data and header, in bytes.

Identification - Consists of an integer identifying this datagram. This field is used to help piece together datagram fragments.

Flags - Consists of 3 bits, of which the low-order 2 bits control fragmentation. One bit specifies whether the packet can be fragmented; the second bit specifies whether the packet is the last fragment in a series of fragmented packets.

Time-to-Live - Maintains a counter that gradually decrements down to zero, at which point the datagram is discarded. This keeps packets from looping endlessly.

Protocol - Indicates which upper-layer protocol receives incoming packets after IP processing is complete.

Header Checksum - Helps ensure IP header integrity.

Source Address - Specifies the sending node.

Destination Address - Specifies the receiving node.

Options - Allows IP to support various options, such as security.

Data - Contains upper-layer information.

Binary Value – This indicates the binary value assigned to the data. The binary value could be assigned by the binary value 0 and 1 alternatively.

Worm Flag – This field consists of the total number of binary value 0s and 1s assigned to the data.

B. Routing table

Table 2 Routing Table

Destination ID	Next Hop	Metric	Worm Flag	
H1	H3	3	5	3
H2	H4	2	4	2
H3	H4	3	6	3
H4	H2	1	8	4
H5	H3	0	10	5

Every network protocol has its own routing table for forwarding packets. Similarly our architecture uses the above routing table as shown in the table 2. Routing table is a database stored by the protocol and used while forwarding packets to its destination. Routing table is used to identify the shortest path and minimum cost to reach the destination. The following fields are used in our routing table.

Destination ID – This field is used to identify the destination network, where the packet must be forwarded.

Next hop - This informs the next nearest and shortest route (router), where the packet should be sent further.

Metric – This field indicates the cost of a route. If multiple routes exist to a given destination network, the metric is used to decide which route is to be taken. The route with the lowest metric is the preferred route.

Worm flag – This field stores the number of 1s and number 0s available in the binary value of the original data. The worm flag, which is available in the packet structure, is updated in the routing table only once initially.

IV. METHODOLOGY

A. Worm Propagation

Worm can be propagated from one location to another location in the following two ways. First type needs an existing file to propagate from one location to another location. The second type is an automatic propagating nature worm. Worm scans for the vulnerable host in the network and automatically propagates to that host in the network and infects it. Once the host is infected, and then the infected host scans for the other vulnerable hosts that are connected in the network and infects them. This process continues until all the hosts are infected. These worms should be detected at the earliest and necessary countermeasures must be taken to provide secure computing in networks. This paper proposes an alternative approach to detect the worms at the router itself and the flow of control, secure computing and quality of service (QoS) are maintained.

B. Worm Detection

Worm could be detected in two ways either host-based detection or network based detection. Host based detection could be done by monitoring and analyzing each host in every period of time. The network based model could be done by monitoring the entire network. Our mechanism goes with the network based model. In our methodology, the original packet data is assigned with binary value respectively. Let's consider the original data is N and its corresponding binary value can be assigned with the binary values 1 and 0 alternatively to each bit of the data.

N* is the binary value of original data assigned with 1, 0 alternatively. The worm flag of the packet structure consists of the number of 1s and number of 0s counted from N*. This worm flag values are also updated in the routing table's worm flag field only once initially, when the packet is about to move from the source. As the packet moves from its source to destination, if a new bit is added with the N such as worm, the N* is also assigned to that bit too and the number of 1s and number of 0s are updated in the worm flag of the packet structure table alone, which is not updated at the routing table's worm flag field. When the packet reaches the router, it compares the worm flag field of the packet structure with the worm flag field of the routing table. If mismatch is found, the router sends the packet to the centralized monitor and scanner module for removing the worm bit. If no mismatch is found the packet is forwarded to the next hop towards its destination. Once the worm contained packet reached the centralized monitor and scanner module, it scans and removes the worm bit that added with the N and sends back the wormless packet to the router. Then the router forwards the packet to the next hop based on its routing table information. Thus the worm contained packet could be detected at the router itself.

C. Pseudo code

```
N=data
N$^*$ = binary value assigned to data
N is forwarded from source to destination
While (Router checks each packet)
{
If (mismatch if found between worm flag)
{
Send the packet to centralized monitor and scanner
}
Else
{
Forward to the next hop
}
While (centralized monitor and scanner scans N$^*$)
{
Remove the worm;
Send the packet back to router;
}
```

V. PERFORMANCE EVALUATION

Worm detection at router level will lead to the following metrics: 1) Detection time (DT), 2) Infection Ratio (IR), 3) Detection Rate (DR), 4) Removal time (RT), 5) Wormless Packet (WP).

A. Detection Time

Our first metrics detection time (DT) defines the time taken to successfully detect the wide spread worm propagation. This assures the speed of the worm detection from its start of the propagation. Also it quantifies the detection speed of our detection scheme. The following mathematical equation represents the speed of our detection scheme. As we know, the routers will change its location dynamically. So the distance of router may vary at time t.

$$DT = d\ (t) \,/\, sp\ (t)$$

Where,

DT = Detection time.

d (t) = Distance.

sp (t) = Speed.

Let’s consider the distance as 50 m at time t and the speed of packet delivery at time t is 128 kbps. Then the detection time could be as follows.

$$DT = 50 / 128$$

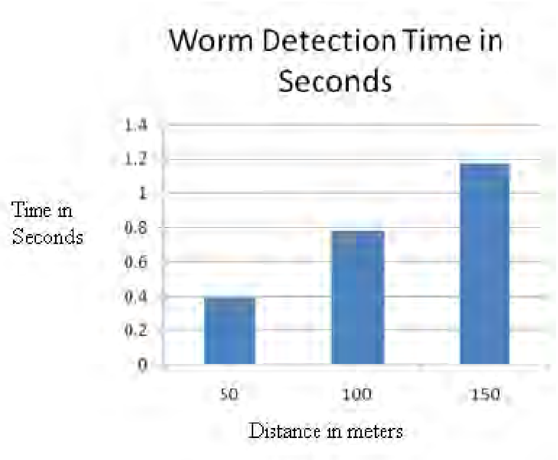
DT = 0.3906 m/bit/s at time t.

Now let’s consider the distance as 150 m at time t and calculate the detection time.

$$DT = 150 / 128$$

DT = 1.1718 m/bit/s at time t.

The following graph shows the worm detection time (DT) at time t. The x axis shows the distance of router in meter at time t and y axis as time in seconds at time t. Our architecture detects the worm very earlier as show in the graph.



Graph 1 Worm detection

B. *Infection Ratio*

Infection Ratio (IR) defines the probability of number of vulnerable computers infected among the total number of vulnerable computers at time t. Our detection scheme process produces less infection ratio, since the worm is being detected at the router itself. Since the worm is detected at the router level, it is possible for the computers to be vulnerable in the network. The infection ratio (IR) can be calculated by the following expression.

IR = V (t) / M (t)

Where,

IR = Infection Ratio

V (t) = Number of vulnerable computers infected at time t.

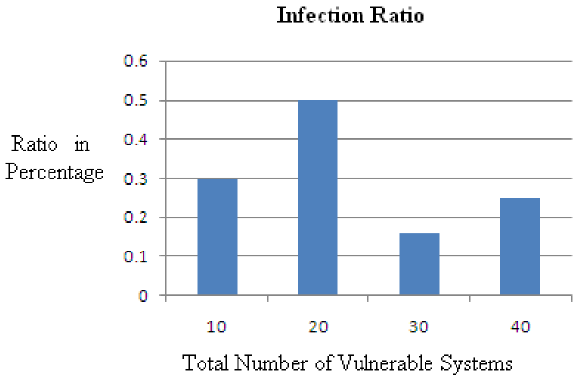
M (t) = Total number of vulnerable computers at time t.

For instance, let’s consider V (t) = 3 and M (t) = 10 and the time is 1 hour. Then the infection ratio (IR) is as follows:

IR = 3 / 10

IR = 0.3

As shown above, the methodology prompts less infection ratio in the networks. Sine worm packet is detected at the router itself, it may be possible for the hosts to be vulnerable and less number of vulnerable computer is infected by the worm. The following graph shows the infection ratio (IR) at time t. The x axis shows the total number of vulnerable systems at time t and y axis as ratio at time t. Our architecture prompts very less infection ratio (IR) as show in the graph.



Graph 2 Infection Ratio

C. *Detection Rate*

Detection Rate (DR) quantifies how accurately the worm is being detected. It is also possible for false alert worm detection. In this paper we use bit manipulation methodology to detect the worm propagation. Our scheme uses binary value to match the bits to find out whether worm attack is happened in the packet or not. Our detection rate is always high since we go for bit manipulation methodology.

D. *Removal time*

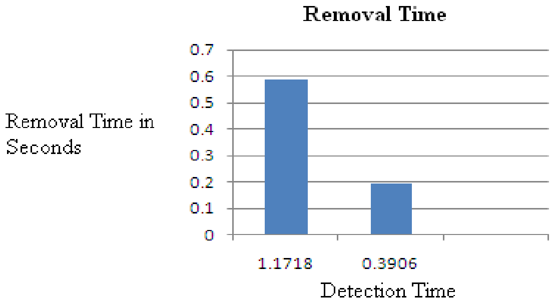
Removal time (RT) defines the time taken to remove the worm from the packet. Here we use binary values for worm identification, which will lead to less removal time (RT). Once the worm packet reached the centralized monitor and scanner module, it scans the N^* for two same binary values appear together instead of alternative values. Once same binary value is found, our scan module scans only those bits and removes the worm from the packet. As we have already stated the detection time as 1.1718, the removal time (RT) could be the half from the detection time (DT).

RT = DT / 2

RT = 1.1718 / 2

RT = 0.5859

The blow graph shows the removal time of our architecture.



Graph 3 Detection Rate

Thus, our architecture scans only at the worm contained bits and removes the worm, which will lead to less worm removal time (RT).

E. *Wormless packets*

The final metrics of this paper proves wormless packets (WP) delivery at the destination in the networks. The worm contained packet at the destination restricts the normal flow and communication in the networks. Obviously the quality of service (QoS) can be maintained by sending wormless packets in the communication networks. In this paper we detect the worm at the router itself, which will somewhat maintain the quality of service in the network. The worm contained packet is detected at the router itself and the worm is scanned using our bit manipulation concept and worm is completely removed by the centralized monitor and scanner module. This way of process always sends the wormless packets to its destination. In this paper, our detection scheme will provide quality of service (QoS) by forwarding wormless packets in the networks.

VI. PARAMETERS AND MEANINGS

Parameters Used	**Meanings**
N	Original Data
N^*	Binary value assigned to the original data
DT	Detection Time
d (t)	Distance at time t
sp (t)	Speed at time t
IR	Infection Ratio
V (t)	Number of vulnerable computers infected at time t.
M (t)	Total number of vulnerable computers at time t
DR	Detection Rate
RT	Removal Time
WP	Wormless Packets

VII. CONCLUSION

In this paper, we have evaluated worms and its types. Also we have studied the comprehensive characteristics worms and its propagating nature. Active worm has an automatic propagation nature from one computer to the other computer. These worms restrict the normal flow of communication in the networks. Secure computing or communication and quality of service (QoS) should be maintained in the networks. In this paper, we have executed a new approach to maintain the secure computing and quality of services (QoS). We have used bits manipulation concept to obtain the secure communication and quality of services (QoS).Our methodology provides secure computing and quality of services (QoS) in the network. Our further study would be studying the other types of smart worm in the networks.

REFERENCES

[1] M. Milton Joe, R.S. Shaji, F. Ramesh Dhanaseelan,"Detection of M-worm to provide secure computing in social networks", Elixir online Journal September 2012.

[2] Wei Yu, Xun Wang, Prasad Calyam, Dong Xuan and Wei Zhao, "Modeling and Detection of Camouflaging Worm", IEEE Transactions on Dependable and Secure Computing, Vol.8, No.3, May-June 2011.

[3] D. Moore, C. Shannon, and J. Brown, "Code-Red: A Case Study on the Spread and Victims of an Internet Worm," Proc. Second Internet Measurement Workshop (IMW), Nov. 2002.

[4] D. Moore, V. Paxson, and S. Savage, "Inside the Slammer Worm," Proc. IEEE Magazine of Security and Privacy, July 2003.

[5] CERT, CERT/CC Advisories, http://www.cert.org/advisories/,2010.12 IEEE transactions on dependable and secure computing, vol. 8, no. 3, may/June 2011.

[6] P.R. Roberts, Zotob Arrest Breaks Credit Card Fraud Ring, http://www.eweek.com/article2/0,1895,1854162,00.asp, 2010.

[7] W32/MyDoom.B Virus, http://www.us-cert.gov/cas/techalerts/ TA04-028A.html, 2010.

[8] W32.Sircam.Worm@mm,http://www.symantec.com/avcenter/venc/data/w32.sircam.worm@mm.html,2010

[9] Worm.ExploreZip,http://www.symantec.com/avcenter/venc/data/worm.explore.zip.html, 2010.

[10] D. Moore, V. Paxson, and S. Savage, "Inside the slammer worm," in IEEE Magazine of Security and Privacy, July 2003.

[11] Z. S. Chen, L.X. Gao, and K. Kwiat, "Modeling the spread of active worms," in Proceedings of the IEEE Conference on Computer Communications (INFOCOM), San Francisco, CA, March 2003.

[12] X. Wang, W. Yu, A. Champion, X. Fu, and D. Xuan, "Detecting worms via mining dynamic program execution," in Proceedings of IEEE International Conference on Security and Privacy in Communication Networks (SECURECOMM), Nice, France, September 2007.

[13] C. C. Zou, D. Towsley, and W. Gong, "Modeling and simulation study of the propagation and defense of internet email worm," IEEE Transactions on Dependable and Secure Computing, vol. 4, no.2, 2007.