ISSN: 0974-2115
www.jchps.com Journal of Chemical and Pharmaceutical Sciences

# Integrated implementation of modelling, propagation and detection of worm in outbox attacker

**P.Jesu Jayarin[*], V.Kishore[2]**
[*]Dept of CSE, Jeppiaar Engg College, Chennai, India
[2] Project Engineer, Infosys, Chennai, India
***Corresponding author: E-Mail:jjayarin@gmail.com**

**ABSTRACT**

Propagation of the Email based Malware is a serious and one of the critical threats that are faced by the present networking systems. It is very important to predict the amount of damage it can cause and the way the worms are propagating to control the damage rate of the network. The modern day worms are more dangerous due to its new features like self-start and reinfection. The self-start and reinfection are the new features that are powering the Email-Malware. These features says that the Email-Malware can attack the already compromised system once again even it has been compromised. Self-start specifies that the worms can replicate themselves again and again once certain system process like restart happens. Even if the system is already infected these system gets infected again and again due to self-start. Reinfection is the process in the system that is already compromised will be compromised again when it opens a malicious link again. The key aim of the implementation is to find the propagation path of these worms and stop them. The first implementation is to distribute the patch and kill the worms. The enhanced part of the implementation will be to stop the worm in the out box of the attacker by scanning it at outbox itself. And hence stop the worm from being propagated in the network. The implementation uses the SII and AOD algorithm to perform the proposed tasks.

**KEY WORDS:** Network Security, email-malware, self-start, reinfection, malware propagation.

## 1. INTRODUCTION

Email is serving as a very important means of communication in the real world, while it is also a cause of a major threat in the network. The Email-Malware is propagating very fast in the network and compromising all the nodes in the network. A malware is sent to the receiver from a source that looks like the trusted one to the receiver. The victims are made to click the mail containing the malware and once the mail with the malicious link is clicked the node gets compromised and the malware starts attacking the system. The users cannot be stopped from opening such malicious links as they might not be aware or capable in finding which is a fake mail containing a malicious link. The researchers are in the process of finding the propagation of the malware and stopping its propagation rather than depending on the users to find it and avoid it. Various steps had been taken by various researchers but have not yet resulted in finding a good solution in stopping the propagation of the malware and stopping them from attacking the system.

The present work regarding these issues involves finding the propagation path of the malwares and once this propagation path has been found out, then measures are being taken to reduce the impact that these malwares are causing. But the current system considers that a malware can affect a single node only once and can send out a malware from that affected node only once (Gao, 2011), whereas the modern day malwares are more aggressive in the sense that they are propagating more rapidly with few new features namely self-start and reinfection (Chen, 2005; Wen, 2012). The earlier malwares were in the sense that if a node is already compromised and if this compromised node is opening a malicious link then the node will not be affected again, whereas the new features add up additional supportive hand to these malwares in the sense that even if the node has been compromised by a malware and the node again clicks the malicious link, the node is infected again. Hence the modern day malwares are posing a greater threat to the network one such modern day worm is “Here You are” (Fossi, 2010). Because these features were not considered by the previous works of the researchers the propagation of these malwares was not very much accurate. Hence this insight has made us to solve the task of considering such attacks and reduce the impact of these malwares on the network.

The previous works used the Susceptible-Infected-Susceptible(SIS) process for finding the propagation but we are proposing the Susceptible-Infected-Immunized(SII) (Sheng Wen, 2014), model for finding the propagation path of the worms and Attackers-Outbox-Detection(AOD) model to stop the worm from being propagated from the outbox of the attacker. The combinations of these two models have proved to be a very much efficient methodology to stop the propagation of the malware.

The important contributions of the proposed project are:

- A new model is being proposed to find the propagation methodology of the email-malware.
- A new model is being designed to stop these malwares at the out box of the attackers.
- The efficiency of the proposed system is found out by propagating various malwares and finding the percentage of accuracy.

**Problem Statement:** Using email for spreading of the malware was in existence from the very olden times. The Melissa is one such example used to propagate through the email. These malware first checks if the node has been infected or not. If the node has already been infected, these malwares will not perform any task on the system. And if they are not infected then they will affect the system. But the modern malwares with the features like self-start and reinfection are really dangerous.

The scenario of the reinfection is shown in the fig1 which says that node A is the source of this network that propagates the worm to the nodes B and C, the node D gets affected by both the nodes B and C multiple times, this shows the principal of reinfection in the sense that even if the B infects node D, node C also infects the node D again and hence the node D contains the worm content more than the parent node. The d1 is the worm from node B and d2 is the worm from node C. The node E will be affected by both the worms sent from node B and node C.

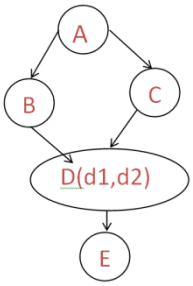


**Figure.1.Network showing how reinfection works**

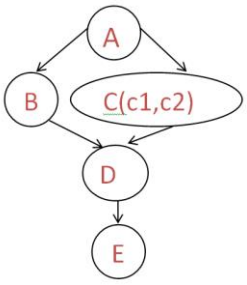


**Figure.2. Network Showing Self-Start**

It states that the node which is already compromised doubles the worm by itself due to some of the system process like restart (Serazzi, 2003). In fig2.the worm in the node C gets multiplied when the node C performs the restart action denoted by c1 and c2, and this increased quantity of the malware will reach the node D and subsequently the node E. Thereby these are the two major problems that are being faced by any node in the network which needs to be given a solution.

The fig3. Shows how many number of nodes are affected by each of the process as the time gets increased.

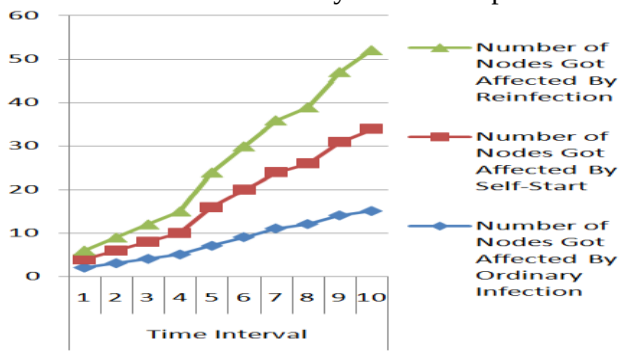


**Figure.3 Graph showing the number of nodes affected in different scenarios**

**Related Work:** A list of previous works have been done to remove the effect of the spreading of worms in the network.

In Gupta (2004) A has proposed the following a number of extensions to the original predator model, persistent predators, including immunizing predators and seeking predators. We report on a set of simulations that explore the effects of predators on small-scale (800 to 1600 node) networks. Our results point out that predators hold significant promise as an alternative to the centralized patch distribution mechanism. The results show that to disinfect systems and distribute patches rapidly across the network without suffering from bottlenecks or causing network congestion predators can be used.

In SteliosSidiroglou (2005), has proposed the following unlike traditional scanning techniques that rely on some form of pattern matching (signatures), his use behavior-based anomaly detection. Under the approach, he .open. all suspicious attachments inside an instrumented virtual machine looking for critical actions, such as writing to the Windows registry, and _ag suspicious messages. The attachment processing can be of oaded to a cluster of ancillary machines (as many as are needed to keep up with a site's email load), thus avoiding the mail server from any computational load.

In Nuno Rodrigues (2012), says that, a generic and systematic model to describe the network dynamics whenever a botnet threat is detected, defining all dimensions, actors, states and actions that need to be taken into account at each moment. They believe that the proposed model can be the basis for developing systematic and integrated strategies, frameworks and tools to predict and fight botnet threats in an efficient way.

**Architecture:** The system consists of two set of implementation one is using SII model shown in the fig4. It specifies that the worm file is sent through the Main Server from the User A to User B and once the worm file has reached the

User B the patch server detects that the worm has been propagated and then it starts sending the patch to the User B. Hence the effect of the worm on the User B will be nullified.

The main server and the patch server will be in contact with each other and once the worm fail has been send to the User B the main server identifies this and the Patch Server is informed regarding this and the patch will be sent to the User B.

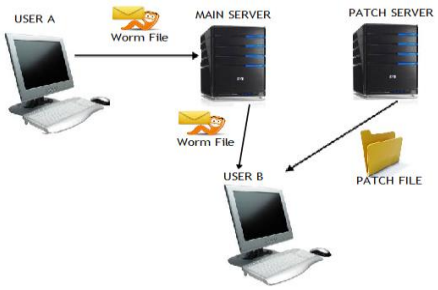


**Figure.4. Destruction of worm using SII model**

Fig5.explains about the AOD model that is used to stop the worm file at the outbox of the attacker itself. The server monitors the out box of the attacker and stops the worms from being propagated.

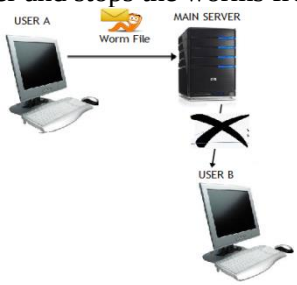


**Figure.5. Destruction of the worm in the outbox of the attacker.**

**System Implementation:** The data flow diagram of the entire process is shown in the fig 6 it completely specifies the entire flow of the proposed system. It says the two different approach present that is first sending the worm to the receiver and then sending the patch file from the patch server and the next thing will be to stop the worm at the outbox itself.

The worm modelling will be the first step of the process and is done to propagate in the network and then see the propagation path so that the patch can be sent through the same path through which the worm had propagated. Fig 6, explains the various phases of the project related to destroying the worm that is getting propagated in the network.

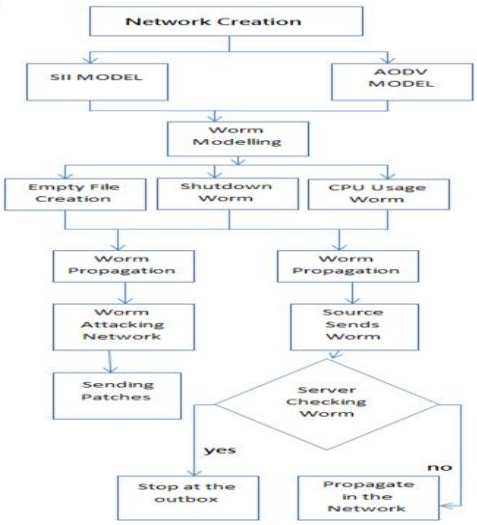


**Figure.6. Data Flow Diagram of Worm Destruction**

## 2. MATERIALS AND METHODS

**The algorithm involved in implementing the process of stopping the worm is as follows:**

```
WormDetect(FileName)
{
        Switch(ch)
        {
                Case 1:         SII(FileName);          Break;
                Case 2:         AOD(FileName);                  Break;
        }
        SII(FileName)
        {
Send the mail from sender to receiver via main server;
If(FileName is a Worm)
{
Call Patch Server to Kill it;
}
}
AOD(FileName)
{
Scan the Outbox of the sender by the Main Server;
If(FileName is a Worm)
Call Patch Server to kill at outbox itself;
}
}
```

**Network Deployment:** The network topology is created to avoid security problems. Network has many number of node details and maintains the connection details also. Nodes are interconnected and exchange data with one another. Network server maintains the node's IP Address, Port details and Status. Node give request to server and get the needed response from server.

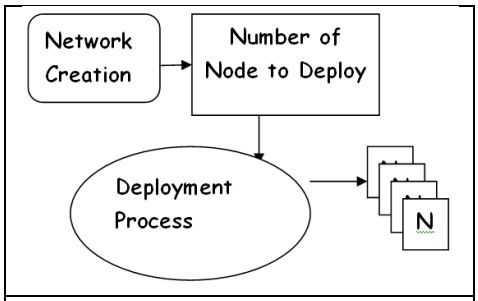


**Figure.7. Network Creation**

**Modelling of Worms:** In this Module we will create the Computer Worm which is malicious code that will perform malicious activities in the User's Computers. In this Project we are creating a New Worm which will create a Folder inside the Folder by developing malicious codes. Once the attackers created the Worm, they will spread the Worm via network to other system.

**Worms Propagation:** Once the attackers created the Worm, they will propagate the Worm via network that is connected to that system, So that the worm will be spread to other Users Computers. While sending via routing technique, the User's has to be present within the contact range. The Attacker can send the worm file via Application that was installed in their Computers. And the computer connected with system is compromised so that the worm will easily transferred to the other system and became infectious system.

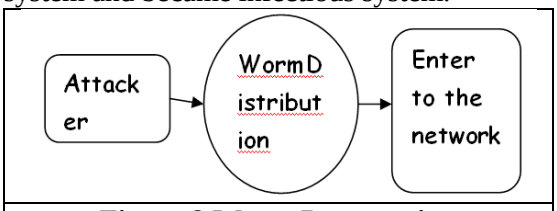


**Figure.8.Worm Propagation**

**Patches Distribution:** Once the Server identifies that a worm file has been sent to the user's Computer, the server will provide the patch files to delete the Worm files. Using an application the patches will be distributed to the User's Computer automatically to clear the Worm.

**Automatic Worms Detection from Outbox:** Once the attack spread the Worm File to other User's Computer the content of the message of the file will be analyzed by the Server to detect whether the file contains that Malicious Behavior or not. If the file contains the malicious behavior, then the Server will detect the file as Worm file. Once the Server detected that the Worm file it will deliver the patches to the User's Computer and deletes the Worm File.

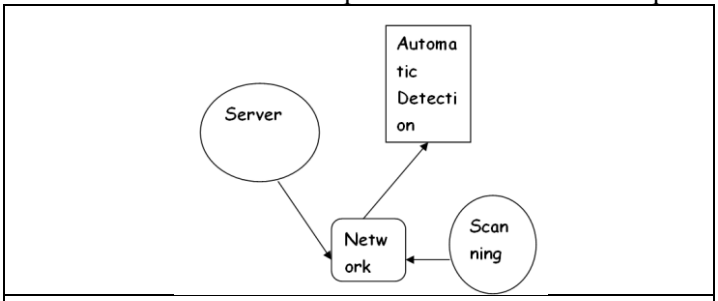


**Figure.9. Detecting worm at the outbox of the attacker**

## 3. RESULT

The efficiency of the project is studied using the number of nodes that is getting affected using the various technics. The graph given below explains the different situation under which the nodes get affected. It shows the comprise rate in a self-start, reinfection, Transfer using SII and followed by AOD methods. The graph is drawn based on the assumption of the number of nodes affected. It depends on the nature of the attack namely self-start and reinfection, both of these types of characteristics of a worm is highly dangerous in nature.

The graph in fig 10 states that the number of nodes getting affected by a worm will increase slowly at its starting point and it will steeply increase after a slow start and start affecting the system in a larger extent. After a particular point of time the number of nodes getting affected due to the worm will get reduced or stagnated as almost all the nodes would have been affected by that time. Hence there will not be any further increase in the amount of nodes affected by the worms after a particular point called the saturation point. By the time the saturation point has reached almost maximum number of nodes will be affected by the propagation of the worm through the email.

The table 1 state the various parameters that were taken into consideration when the system was executed.

**Table.1.Parametes for evaluation**

| Parameters | Description |
|---|---|
| Number of Nodes | 30 |
| Time of execution | 15 mins |
| Connection used | TCP/IP |
| Transfer mode used | FTP |
| Algorithms used | SII/AOD |
| Number of system infected before algorithm used | 27 |
| Number of system infected after implementing algorithm | 2 |

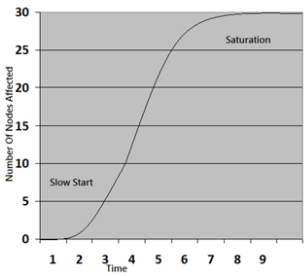


**Figure.10. Graph Showing Worm Propagation Rate**

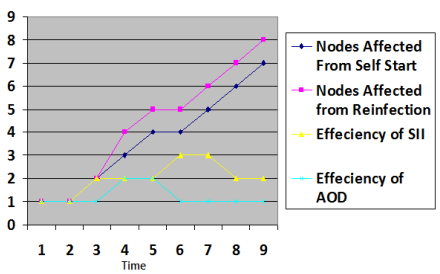


**Figure.11.Graph showing Efficiency of eachmodels**

The graph states the performance metrics of the project using the various propagation technics. The reinfection is assumed to be the most important and vulnerable characteristics of a modern day worm. Hence the number of nodes affected due to reinfection is shown very high. The self-start is assumed to be little less harmful

than that of the reinfection. The SII model proves to help the network from getting affected from the worm propagation from the assumption of the graph. The AOD model has almost reduced the worm propagation rate to nil. It allows almost no worm to propagate in the network at all, as the worms are stopped at the out box of the attacker itself.

## 4. CONCLUSION

The proposed methodology has proved a very good rate of detection and killing of the worm present in the network. Theses worms that are being propagated in the network are very easily identified and stopped from being propagated in the network.

## REFERENCES

Calzarossa M and Gelenbe E, Performance Tools and Applications to Networked Systems, Revised Tutorial Lectures, Springer-Verlag, 2004.

Cert, advisory ca-1999-04, Melissa Macro Virus, 2009.

Cert, Advisory ca-2000-04, Love Letter Worm, 2000.

Cert, Advisory ca-2001-22, w32/sircam Malicious Code, 2001.

Cert, Incident Note in-2003-03, w32/sobig.f Worm, 2003.

Chen Z and Ji C, Spatial-Temporal Modeling of Malware Propagation in Networks, IEEE Trans, Neural Networks, 16(5), 2005, 1291-1303.

Fossi M and Blackbird J, Symantec Internet Security Threat Report, technical report Symantec Corporation, 2010.

Gao C, Liu J, and Zhong N, Network Immunization and Virus Propagation in Email Networks: Experimental Evaluation and Analysis, Knowledge and Information Systems, 27, 2011, 253-279.

Gupta A, DuVarney D.C, Using predators to combat worms and viruses, a simulation-based study, Computer Security Applications Conference, 2004, l0-16

Nuno G, Rodrigues, Antonio Nogueira and Paulo Salvador, Fighting Botnets - A Systematic Approach, EMERGING, The Fourth International Conference on Emerging Network Intelligence, 2012.

Rozenberg B, Gudes E and Elovici Y, SISR, A New Model for Epidemic Spreading of Electronic Threats, Proc. 12th Int'l Conf.Information Security, 2009, 242-249.

Serazzi G and Zanero S, Computer Virus Propagation Models, Proc. 11th IEEE/ACM Int'l Conf. Modeling, Analysis and Simulations of Computer and Telecomm. Systems (MASCOTS '03), 2003, 1-10.

Sheng Wen, Wei Zhou, Jun Zhang, Yang Xiang, Wanlei Zhou, Weijia Jia and Cliff C. Zou, Modeling and Analysis on the Propagation Dynamics of Modern Email Malware, IEEE Transactions On Dependable And Secure Computing, 11(4), 2014.

Stelios Sidiroglou, John Ioannidis, Angelos D. Keromytis, and Salvatore J. Stolfo, An Email Worm Vaccine Architecture, First International Conference, ISPEC 2005, Singapore, 2005, 11-14.

Wen S, Zhou W, Wang Y, Zhou W and Xiang Y, Locating Defense Positions for Thwarting the Propagation of Topological Worms, IEEE Comm. Letters, 16(4), 2012, 560-563.

Wen S, Zhou W, Zhang J, Xiang Y, Zhou W and Jia W, Modeling Propagation Dynamics of Social Network Worms, IEEE Trans, Parallel and Distributed Systems, 24(8), 2013, 1633-1643.

Xiong J, Act, Attachment Chain Tracing Scheme for Email Virus Detection and Control, Proc. ACM Workshop Rapid Malcode (WORM '04), 2004, 11-22, 2004.