Security framework for malware detection in a decentralised peer to peer network

Ayangbekun, Oluwafemi J.
Department of Information Systems
University of Cape Town
Cape Town, South Africa
Phemmyc@yahoo.com

Amodu, Taiwo O.
Department of Computer Science
Crescent University Abeokuta
Abeokuta, Nigeria
Taiwo311@gmail.com

Abstract
In a network of connected computers which aims to perform several tasks and important operations in real-time detection of malware requires computers to share information and access resources from other connected systems in the network. A decentralized P2P network which enables peers to have equal privileges also operate as both server and a client, and gives connected peers in a network equal access control. The major security challenge to ensure that these computer tasks are done smoothly and efficiently is the threat by malicious software - software used to disrupt computer operation, gather sensitive information, or gain access to private computer systems. Several attempts has been made to combat security challenges caused by the plethora of malware in a network but yet despite the hard work that has been put to achieve this, hackers or malicious software developers still don't relent in inventing new ways to nullify the hard work put towards combating malware attacks in a network. This paper focuses on a security framework for malware detection to ensure that connected peers are well secured by protecting each connected peer from malware attacks, providing security measure which monitors the shared content of the connected peers and to detect malware in real time (RT) which will also flag any detected malware in the network during the course of scanning of the connected peers automatically.

Keywords- ClamAV; Malware; Security; Network; Decentralised P2P network; Crawler; Node, plethora of malware.

I. INTRODUCTION

Contemporarily computers are taking over virtually every task that we do while accounting for a degree of accuracy, efficiency and maximum utilization of data for storage and processing. Malware, short for malicious software, is software used to disrupt computer operation, gather sensitive information, or gain access to private computer systems. The majority of active malware threats are usually worms, Trojans rather than viruses. A security framework is a designed structure indicating an approach, steps to be taken and/or the mechanism that is adopted to secure data and other vital information of a system in other for it not to be compromised by malicious codes or software and as well attacks by unauthorized users by restricting access to confidential information so as to fully secure the system.

Peer-to-peer network is a network where there is no specific designation of nodes in the networks, i.e., no peer is designated to be a server while another peer being a node. In the case of domain network, it is required to designate a node as server constantly and other nodes as clients. A decentralised P2P network is a network platform in which the connected peers in the network have equal privilege to send and receive files within the network, it is a network that is not based on a centralised architecture where by a peer acts as the main server to other peers in the network. A decentralised P2P network offers a unique advantage over a centralised P2P network in the sense that peers in a decentralised network does not depend on another peer or a centralised server before they can send, share and/or receive files and/or information from other peer(s) unlike in the case of a centralised P2P network where the peers depends on a centralised server. The P2P model is different from domain model. Peer means a node with same designation. It does mean that in P2P network there is no concept of naming server and client. All the nodes are given equal importance and that is the reason they are known as peers. In a decentralised P2P network, every computer plays the roles of a client and a server at the same time. The computers can initiate requests to other computers, and at the same time respond to incoming requests from other computers on the network.

A major security threat in a decentralized P2P network is the vulnerability of data through rapid proliferation of malware attacks and malicious intentions of intruders that aim to compromise a system by damaging the system or stealing confidential and vital information. So many researches has been made to fight against malware proliferation in a network, a decentralized P2P to be precise, but yet still, malware continue to exist and thereby becoming the major security threat in securing highly confidential information in a system. Despite the efforts of researchers to curb the plethora of malware, yet malware developers have also been developing codes to nullify the efforts of researchers to capture them and irrespective of how hard a researcher tries to fight against the proliferation of malware attacks, malware developers has not relented in their efforts to further make their codes hidden deep down in our computer systems and making it difficult and almost not impossible for them to be detected by system firewalls and well known antivirus. The research to fight against the spread of malware is continuous since malware developers will not seize to develop new variants to defile previous efforts to capture them. This paper focuses mainly on intrusion detection, and this project addresses a security framework for malware detection in a decentralized P2P network.

II. SECURITY FOR P2P NETWORK

Providing Security for Peer-to-Peer network [1]. They demonstrated in their paper the application of Trusted Computing to securing Peer-to-Peer (P2P) networks. They identify a central challenge in providing many of the security services within these networks, namely the absence of stable verifiable peer identities. In their paper, they demonstrate how features of the trusted computing group (TCG) specifications can be employed to enhance the security of P2P environments in other to mitigate pseudo-spoofing attacks, in which malicious parties are able to claim multiple identities, attack P2P networks, and represent a fundamental security threat whereby a peer is creating and handling more than one pseudonym at once. They took this a step further by showing how runs of DAA protocol can be used to build entity authentication at the level of pseudonyms and can be securely linked to the establishment of secure channels with known endpoints.

They further described the specific TCG mechanisms and commands which enabled them to establish security in P2P networks.

III. AUTOMATIC MALWARE DETECTION

Automatic Malware Detection Using Common Segment Analysis and Meta-Features [2]. Presented in their paper an automatic malware detection using common segment analysis. Their paper proposes novel methods, based on machine learning to detect malware in executable files without any need for pre-processing, such as unpacking or disassembling. The methodology they used was the basic method (Mal-ID) that uses common segment analysis in order to detect malware files.

In the detection phase, The Mal-ID basic is a feature extraction process followed by a simple static decision rule. It operates by analyzing short segments extracted from the file examined. Each segment comprises a number of 3-grams depending on the length of the segment (e.g., a segment of length 4 bytes is comprised from two 3-grams that overlap by two bytes). Three features can be derived for each segment: Spread, MFG, and Entropy.

- *Spread:* The Spread feature represents the spread of the signature's 3-grams along the various areas for all the files in a given repository. The Spread feature can be calculated as follows; for each 3-gram, first retrieve the 3-gram-relative-position-withinfile bit-field, and then perform 'And' operations over all the bit-fields and count the resulting number of bits that are equal to 1.
- *MFG:* MFG is the maximum total number of file-groups that contain the segment. The MFG is calculated using the 3-gram-files-association bit-field, in the same manner that spread is calculated.
- *Entropy:* The entropy measures the bytes within a specific segment candidate. The entropy feature is also used to enable identification of compressed areas (such as embedded JPEG images) and long repeating sequences that contain relatively little information.

IV. PROLIFERATION OF MALWARE

Proliferation of malware is simply the means and process that malware undergo to spread across a network by searching for vulnerable nodes to infect and then down to another node, in that manifest until they have circulated throughout the network. Malwares proliferate in so many ways and neither of the ways is to be in favor of the victim because the sole aim of the proliferation of malware is to fulfill the purpose why they are actually created in the first place, which is to spread highly dangerous codes into the user's system while they vary in their various motives of propagating, mostly to damage systems and to steal confidential information for the sole aim of making gains. The most common pathway of malware from criminals to users is through the Internet; primarily by e-mail and the World Wide Web. We are now in a global world whereby the effectiveness and consistency of organizations in terms of delivery of good services and operations now depend on the use of internet for their progress and success.

A report by Microsoft stated in May 2011 that one in every 14 downloads from the Internet may now contain malware code, so basically, it is somewhat impossible to avoid the possibility of being a victim of malware attacks in as much as we surf the internet and majorly download files and any sort of attachments from the internet. Social media and facebook in particular, are seeing a rise in the number of tactics used to spread malware to computers, majorly by clicking on links that will divert the browser of a user to the criminal's page and thereby causing each and every user that clicks on the link to get infected automatically. A user does not really have to download a file of any kind before he/she can be vulnerable to malware attacks. Browsing a malicious page alone can cause havoc on a user's system, let alone by clicking on the link that is developed majorly to cause potential damage(s) on systems. Since there is no guarantee that a user can possibly be inevitable to malware attacks. This project address measures to be taken to avoid the risk of being a victim of malware attacks, precisely in a decentralized P2P network in other not to go beyond the scope of the proposed problem.

V. SYSTEM ARCHITECTURE

The figure below (Fig. 1) shows various phases that are undertaken to ensure a successful transfer of malware free file(s) from a peer to another peer in a connected network. As soon as the system is started and before file(s) could be sent between peers across the network, there is need for a peer (peer A) to search for other peers that are connected to the network and to choose a particular peer to connect to (peer B). After peer B has been selected to connect to, peerA then browses the shared folder(s) that is available for access on peerB and chooses which file to download and before the transfer process can begin, the system will scan the particular selected file for possible malware infection. If the file is clean, the transfer process will continue until peerB receives the file and then peerA will be notified of a successful file transfer. But if otherwise, and a malware is detected, the transfer process will be terminated and then notify other peers in the network about the detected malware and from which peer that is infected and then the process stops or peerA can choose to start over and then initiate connection to another peer that is connected to the network

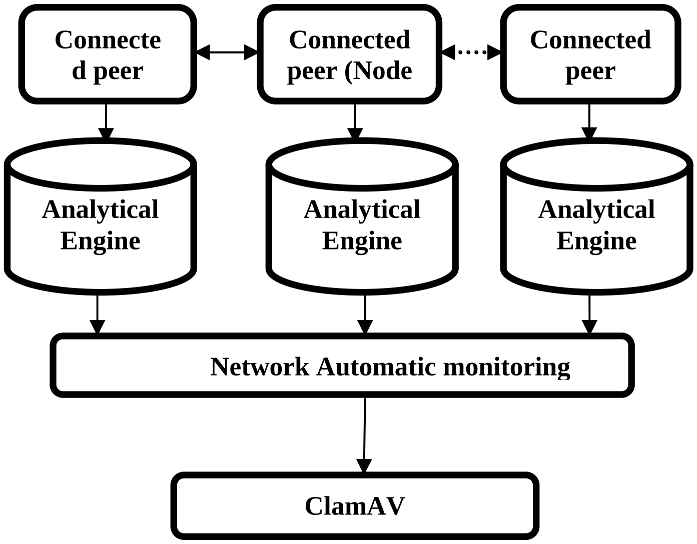


Figure 1. System Architecture

A. *Connected Peers (Node A, B and N)*

In the above architecture (Fig.1), it shows the connected peers in a network which are designated as nodes A, B and N respectively. Node A signifies that a connection is established from a system while node B signifies that a connection is established from a second system in the network up to node N which signifies connection from the last system that is connected to the network which can as well vary due to the number of nodes that is allowed to be connected to the network

B. *Analytical Engine*

The analytical engine entails the workability of a crawler, which makes the application more reliable, is a script that moves through the network of connected peers. It locates the shared folder on each connected peer and scans for malware in runtime (RT). Whenever a malware is detected, in the shared folder of a connected node, it will prompt the system to notify all other connected peers in the network about the particular infection. It can only notify other peers that have the application installed on their system about a known malware and signifying which node is infected in the network (Fig. 1). But if a particular node does not have the application installed, then that node will not be notified of a found malware in the network but this will not hinder the functionality and operation of the crawler as it will still go ahead to craw over all detected shared folders in the network for malware.

C. *Network Automatic Monitoring*

Automatic monitoring across the network involves the phase whereby scanning is done in real time across the network by the crawler. It automatically seeks for malware on every connected peer's share folder in the network automatically. This cannot be done manually as it has been programmed to operate automatically across every found shared folder.

D. *ClamAV Automatic/manual update*

This phase (Fig. 1) provides a service that can be either set to automatic or manual update of the ClamAV which is an open source antivirus engine designed to detect viruses, worms, Trojans and other malicious threats in a system [5]. It provides a high performance mutli-threaded scanning daemon, command line utilities for on demand file scanning, and an intelligent tool for automatic signature updates (ClamAV Team. “About ClamAV”. June 16th, 2014), which can as well be set update manually.

VI. ARCHITECTURAL FLOW

In the flow below (figure 2) shows various phases that are undertaken to ensure a successful transfer of malware free file(s) from a peer to another peer in a connected network. As soon as the system is started and before file(s) could be sent between peers across the network, there is need for a peer (peer A) to search for other peers that are connected to the network and to choose a particular peer to connect to (peer B). After peer B has been selected to connect to, peerA then browses the shared folder(s) that is available for access on peerB and chooses which file to download and before the transfer process can begin, the system will scan the particular selected file for possible malware infection. If the file is clean, the transfer process will continue until peerB receives the file and then peerA will be notified of a successful file transfer. But if otherwise, and a malware is detected, the transfer process will be terminated and then notify other peers in the network about the detected malware and from which peer that is infected and then the process stops or peerA can choose to start over and then initiate connection to another peer that is connected to the network.

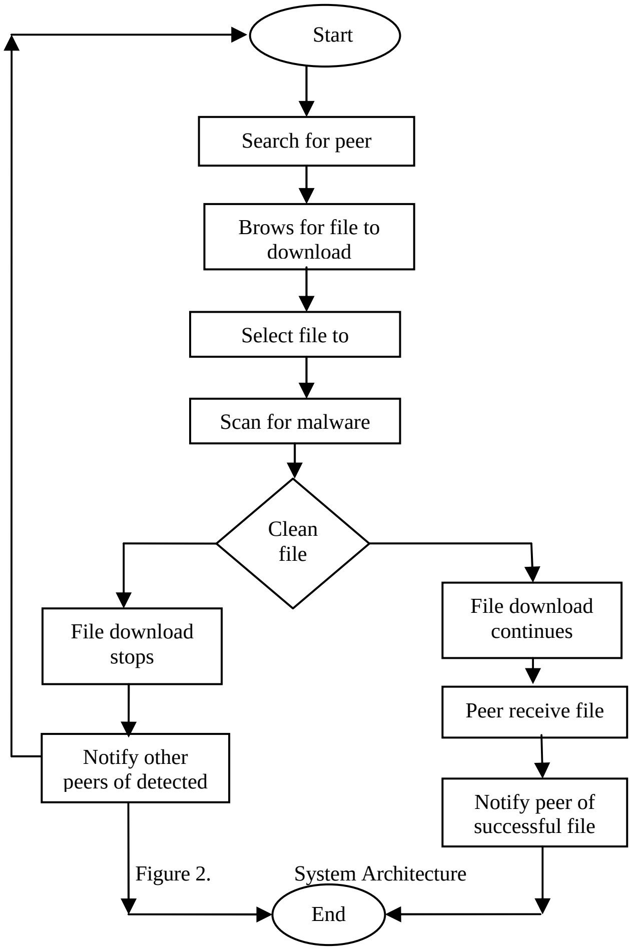


Figure 2. System Architecture

VII. SOFTWARE IMPLEMENTATION

The business logic of system seats on top of the Microsoft.NET platform with CSharp as the programming language. The presentation layer will be Window Presentation Foundation WPF make-up language for platform independent. This produces the homepage pane (Fig. 3) which serves as the entry point to the application representing the malware scanner.

It consists of modules which make up the system. It has two tabs at the top left corner (Share folder and Browse for file). The application can be minimize to the task bar and also re-launched from the task bar by right clicking on the Icon at the task bar as shown in figure 4

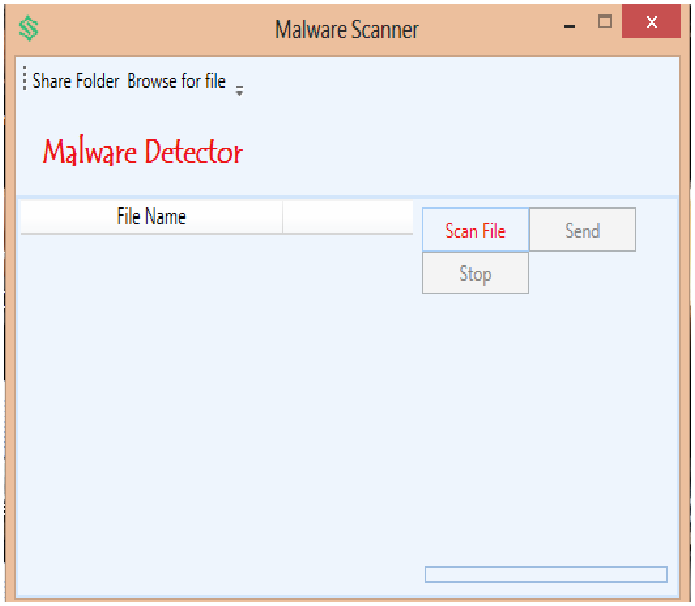


Figure 3. System Homepage

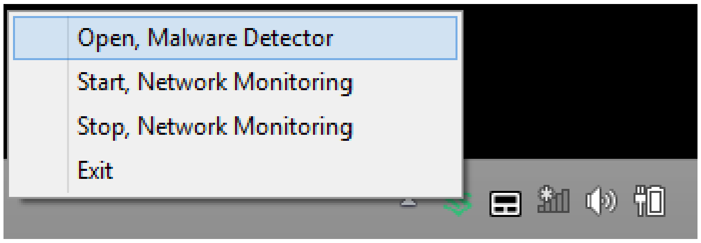


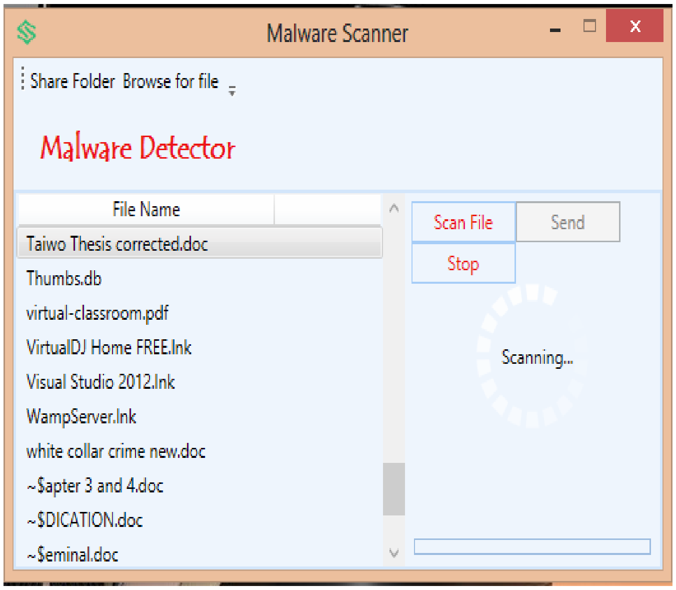
Figure 4. Taskbar Launch

A. Share Folder

The share folder tab (Fig. 3) allows the user to share folders or files to a connected peer in the network. This allows other system within a connected network to share file that is free from virus/malware to the shared folders.

B. Browse for File

The browse for file tab (Fig. 3) allows a user to navigate to files from any directory in the user's system. Therefore when a particular file is selected, the user can send the file to other peer(s) in the network. Before a selected file can be sent across the network, it is necessary for the file to be scanned before the send tab can be enabled as shown in the malware detection pane (Fig. 5).



File Name
Taiwo Thesis corrected.doc
Thumbs.db
virtual-classroom.pdf
VirtualDJ Home FREE.lnk
Visual Studio 2012.lnk
WampServer.lnk
white collar crime new.doc
~$apter 3 and 4.doc
~$DICATION.doc
~$eminal.doc

Figure 5. Scan Process

References

[1] Shane Balfe, Amit D. Lakhani and Kenneth G. Paterson, Information Security Group, “Trusted Computing: Providing Security for Peer-to-Peer Networks” 2005.

[2] Gil Tahan, Lior Rokach, Yuval shahal. Mal-ID: Automatic Malware Detection Using Common Segment Analysis and Meta-Features; Journal of Machine Learning Research 13 (2012) 949-979

[3] “Peer-to-peer definition” (2005). Retrieved March 16, 2005, from http://en.wikipedia.org/wiki/Peer_to_peer.

[4] Prof.P.Pradeep Kumar,Naini Shekar Reddy,R.Sai Krishna,Ch.Kishor Kumar and M.Ramesh. Preventive Measures For Malware In P2P Networks; International Journal of Engineering Research and Applications (IJERA). Vol. 2, Issue 1, Jan-Feb 2012, pp. 391-400

[5] "About ClamAV", www.clamav.net

AUTHORS PROFILE

Ayangbekun, Oluwafemi J. received his Bachelor of Technology (BTech) in Computer Engineering from Ladoke Akintola University of Technology Ogbomoso, Nigeria in 2003. He also obtained his Masters of Science (MSc) in Computer Science from University of Ibadan, Nigeriain in 2007. He is presently a PhD researcher in the department of Information Systems, University of Capetown, South Africa.

Taiwo Amodu received his Bachelor of Science (BSc) in Computer Science with Economics from Crescent University Abeokuta, Nigeria in the year 2014.