ISSN:0975-9646

Classification and Adaptive Novel Class of Botnet Detection

Nijhandhan.M[1], Narmadha.R.P[2]
[1,2]*Department of Computer Science and Engineering*
[1,2]*Sri Shakthi Institute of Engineering and Technology*
L&T Bypass Road, Coimbatore, India

Abstract—malicious software typically resides stealthily on a user's computer and interacts with the user's computing resources. Our goal in this work is to improve the trustworthiness of a host and its system data. By providing a new mechanism that ensures the correct origin or provenance of critical system information and prevents adversaries from utilizing host resources. It defines the data provenance integrity as the security property stating that the source data is generated which cannot be spoofed or tampered with the host resources. The decentralized nature of Peer-to-Peer (P2P) botnets makes them difficult to detect, their distributed nature also exhibits resilience against take-down attempts, and moreover smarter bots are stealthy in their communication patterns and elude the standard discovery techniques which look for anomalous network or communication behaviour. In this project, a novel methodology is used to detect P2P botnet traffic and differentiate it from P2P traffic in a network. A cryptographic protocol for ensuring secure and timely availability of the reputation data of a peer to other peers at extremely low costs. Content auditing is done on receiver side even signature fails and mentions the ratio or impact of fake content. As result, a peer's reputation motivates it to cooperate and desist from malicious activities like denial of service, brute force and many attacks.

Keywords— Botnet, P2P, Network traffic, Cryptographic protocol, Content auditing.

I. INTRODUCTION

PEER-TO-PEER(P2P) networks are self-configuring networks with minimal or no central control. P2P networks are more vulnerable to dissemination of malicious or spurious content, malicious code, viruses, worms, and trojans than the traditional client-server networks, due to their unregulated and unmanaged nature. For example, the infamous VBS. Gnutella worm that infected the Gnutella network, stored trojans in the host machine. A botnet is a collection of internet connected programs communicating with other similar programs in order to perform tasks. The word botnet is a combination of the words robot and network. Botnets is a network which consists of bots and the botnets are created by the botmaster for communication infrastructure to perform malicious activities like Denial-of–Service, Brute force attacks and so on. Botmaster is a person which controls the bots. The difference between botnets and other malwares is the bots communicate through Command-and-Control(C&C). The bots uses C&C to receive the commands and perform malicious activities which devoted by botmaster. Our goal is to improve the trustworthiness of the network level data flow. It provide mechanisms that ensure the correct origin or provenance of critical system data, which prevents adversaries from utilizing host resources

II. SYSTEM ANALYSIS

The existing system includes designing an effective P2P-botnet detection system is faced with several challenges. First, the P2P file-sharing and communication applications, such as Bit torrent, emule, and Skype, are very popular and hence C&C traffic of P2P botnets can easily blend into the background P2P traffic. This challenge is further compounded by the fact that a bot-compromised host may exhibit mixed patterns of both legitimate and botnet P2P traffic (e.g., due to the coexistence of a file-sharing P2P application and a P2P bot on the same host). Second, modern botnets tend to use increasingly stealthy ways to perform malicious activities that are extremely very hard to observed in the network traffic. Third, as the volume of network traffic grows rapidly, the deployed detection system is required to process a huge amount of information efficiently. To date, a few approaches capable of detecting P2P botnets. However, these approaches cannot address all the challenges. In P2P systems, the computational burden of the system can be distributed to peer nodes. Therefore, the users become themselves actors by sharing, contributing, and controlling the resources of the systems. This characteristic makes P2P systems very interesting for the development of decentralized applications

This system also identifies P2P bots within a monitored network by detecting the C&C communication patterns that characterize P2P botnets, regardless of how they perform malicious activities in response to the botmaster's commands. Specifically, it derives statistical fingerprints of the P2P communications generated by P2P hosts and leverages them to distinguish between hosts that are part of legitimate P2P networks (e.g., file sharing networks) and P2P bots. The high scalability of our system stems from the parallelized computation with bounded computational complexity This system analyses the traffic generated by the P2P clients and classifies them into either legitimate P2P clients or P2P bots. It investigate the active time of a P2P client and identify it as a candidate P2P bot if it is persistently active on the underlying host. Further analyse the overlap of peers contacted by two candidate P2P bots to finalize detection. The goal is to improve the trustworthiness of the network level data flow; specifically, we provide mechanisms that ensure the correct origin or provenance of critical system data, which prevents adversaries from utilizing host resources.

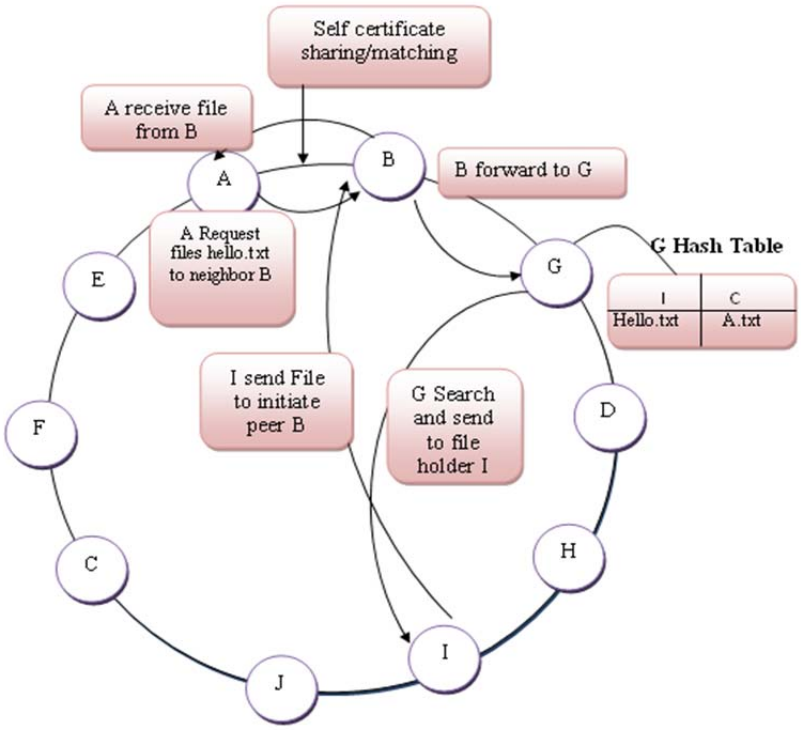

Fig no.2.Architecture

Proposed system is to decrease malicious activity in a P2P system by establishing trust relations among peers in their proximity. No a priori information or a trusted peer is used to leverage trust establishment. Peers do not try to collect trust information from all peers. Fig2 illustrates Each peer develops its own local view of trust about the peers interacted in the past. In this way, good peers form dynamic trust groups in their proximity and can isolate malicious peers.

By implementing P2P file sharing and searching are the conducted experiments to understand impact of our proposed system in mitigating traffic attacks. It enhances and complements the capabilities of existing P2P botnet detection systems, but its not prefect. To develop a more robust defence techniques, where the aforementioned discussion outlines the potential improvements of our system. New approach needs content signature because encryption will make content signature very secure. By training data set to build a machine learning based model becomes it is very challenging to get traffic of P2P botnets before they are detected.

III. Related Work

In this section, we briefly introduce the related work on Botnet, Hash Table Reputation and Data Provenance Integrity.

A. Botnet

A botnet is a collection of compromised hosts (a.k.a. bots) that are remotely controlled by an attacker (the botmaster) through a command and control (C&C) channel. Botnets may structure their C&C channels in different ways. In a centralized architecture, all bots in a botnet contact one (or a few) C&C server(s) owned by the botmaster. However, a fundamental disadvantage of centralized C&C servers is that they represent a single point of failure. In order to overcome this problem, botmasters have recently started to build botnets with a more resilient C&C architecture, using a peer-to-peer (P2P) structure or hybrid P2P/centralized C&C structures. Bots belonging to a P2P botnet form an overlay network in which any of the nodes (i.e., any of the bots) can be used by the botmaster to distribute commands to the other peers or collect information from them.

B. Hash Table Reputation

Reputation values are used to guide peer selection during streaming. A higher reputation value indicates that the node is more trustworthy in terms of the collective evaluation by peers that have had data exchange with it. There are various ways to use reputation values. For example, a peer that issues downloading requests may receive several responses. It can compare the reputation of the responding peers and choose the one with the highest reputation to download data. This reduces the risk of receiving abnormal service from malicious nodes. Unfortunately, there is so far no mechanism that can completely prevent the attack of peers being compromised. We plan to engage in a study of attacks made via anonymous operations, and develop corrective and preventive methods as a part of trust building and trust management research.

C. Data Provenance Integrity

We define a new security property data-provenance integrity. It states that the source from which a piece of data is generated can be verified. We give the concrete illustration of how data-provenance integrity can be

realized for system-level data namely, keystroke events and outbound network packets in a host based setting. Our work focuses on a host-based approach for ensuring system-level data integrity and demonstrates its application for malware or botnet peer detection. In comparison, network trace analysis typically characterizes malware communication behaviours for detection.

IV. MODULES

In this section, the proposed system consists of five modules: neighbour peer extraction, hash table based searching, trustworthy peer communication, asymmetric cryptography approach and file sending and receiving.

A. Neighbour Peer Extraction

Peer's construct its neighbors by sending connection request with its own certification likewise all peers shares its identity and its self certificate with its neighbor peer's. This certificate is compare by neighbor peer when the peer send file search request to the one of neighbor peer thus avoids unwanted flow of packets which reduce the traffic.

Source peer select the neighbor peer as per highest reputation metric/value. The reputation value is calculated under each peer's total entry in the hash table or its load status.

B. Hash Table Based Searching

In the unstructured P2P networks, peers willingness to share the content they have and forward the queries plays an important role during the content search process. Each and every peer in the network must maintain this table which is used to forward the peer request to the apt peer instead of its neighbour peer. The proposed system uses the distributed hash table where each and every peer has the separate hash table.

The information stored in the hash table is based on Reputation management (tracking peers past activity).It helps to perform the file searching operation efficiently. The self certificate is used for ensuring secure and timely availability of the reputation data of a peer to other peers. Since each peer stores its own reputation locally, for reputations to be reliable and elective, they have to be updated and stored securely to prevent malicious peers from the reputation system.

C. Trustworthy peer communication

Each neighbour peer must accept the source file request only after successful match of self certification. Thus avoid the malicious peers which try to compromise with other peer address. Then the initiate neighbour peer forward the source requests to the next peer with the hidden identity of source peer thus reduce the possibilities of unwanted rebroadcast packets and also malicious activities. So the initiate neighbour peer acts as temp source then forward the source request to its neighbour peer this continues until the file found. The destination peer sends the file to the initiate neighbour and the initiate neighbour peer sends the file to the source thus avoid the length of the packet flow and control the misbehave peers.

D. Asymmetric Cryptographic Approach

Peers generate universally unique identifications locally and store them along with their public key, their current IP address. Peer request like sharing self certificate with neighbour peer or file search request or file send/receiving are done under efficient encryption and decryption process.

E. File Sending and Receiving

Each and every peer has the unique identity, based on this, the peer is identified and the transaction will begin. The self-certification is attached with identity of the peer. Each and every peer has the unique identity, based on this, the peer is identified and the transaction is begined. The self-certification is attached with identity of the peer.

The self certification where the algorithm generates the key and public key, these identities are attached with reputation of the given peer. The sender sends the requested file which is associated with its private key and signature, at receiver side receives the file and decrypts the file.

V. EXPERIMENTS

In this section, we present various experiments to detect the botnets in Peer to Peer network the effectiveness of the proposed the self certification and content auditing. In this section, we demonstrate its utility in detecting botnet.

A. Detecting Malicious Action Scheme

Many network services consist of a large set of independent nodes, and these nodes are required to follow some rules to cooperate so as to achieve a given network functionality. To realize such network service, every node must communicate or provide local services with a subset of other nodes which are called neighbours, e.g., upload packets to neighbours, download packets from neighbours and forward packets for neighbours and so on. To guarantee the correct functionality of the network service such that every node can get service with desired performance, nodes must follow the predefined protocols when they participate in the communication with their neighbours.

VI. CONCLUSIONS

This project presents self-certification, an identity management mechanism, reputation model, botnet detection and a cryptographic protocol that facilitates generation of global reputation data in a P2P network, in order to expedite detection of rogues. It describes a general approach for improving the assurance of system data and properties of a host, which has applications in preventing and identifying malware activities. Our host-based system security solutions against malware complement network-traffic-based analysis. The following technical contributions in this project are proposed the model and operations of cryptographic trust verification in a host-based security settings. This demonstrated the trust verification approach in a lightweight framework for ensuring the integrity of outbound packets of a host. It also identifies the performance bottleneck and optimizes its scalability and it provides the high accuracy and scalability ob detecting P2P botnet.

ACKNOWLEDGMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organization. I would like to extend my sincere thanks to all of them.

I am highly indebted to Sri Shakthi institute of engineering and technology and Ms.R.P.Narmadha for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project. I would like to express my gratitude towards my parents & member of Sri Shakthi institute of engineering and technology for their kind co-operation and encouragement which help me in completion of this project.

My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

REFERENCES

[1] D. Dittrich,S. Stover, J. Hernandez, and S. Dietrich, "Analysis of the storm and nugache trojans: P2P is here," in Proc. USENIX, vol. 32. 2007, pp. 18–27.

[2] H. Saidi,P. Porras, , and V. Yegneswaran, "A multi-perspective analysis of the storm (peacomm) worm," Comput. Sci. Lab., SRI Int., Menlo Park, CA, USA, Tech. Rep., 2007.

[3] H. Saidi ,P. Porras, and V. Yegneswaran. (2009). Conficker C Analysis [Online]. Available: http://mtc.sri.com/Conficker/addendumC/index.html

[4] I. Sinclair, C. Nunnery, and B. B. Kang, "The waledac protocol: The how and why," in Proc. 4th Int. Conf. Malicious Unwanted Softw., Oct. 2009, pp.

[5] L. Lemos. (2006). Bot Software Looks to Improve Peerage [Online]. Available: http://www.securityfocus.com/news/11390

[6] Q. Ke ,Y. Zhao, Y. Xie, F. Yu, , and Y. Yu, "Botgraph: Large scale spamming botnet detection," in Proc. 6th USENIX NSDI, 2009, pp. 1–14.

[7] R. Perdisci ,G. Gu, , J. Zhang, and W. Lee, "Botminer: Clustering analysis of network traffic for protocol- and structure-independent botnet detection," in Proc. USENIX Security, 2008, pp. 139–154.

[8] T.-F. Yen and M. K. Reiter, "Are your hosts trading or plotting? Telling P2P file-sharing and bots apart," in Proc. ICDCS, Jun. 2010, pp. 241–252.

[9] S. Nagaraja, P. Mittal, C.-Y. Hong, M. Caesar, and N. Borisov, "BotGrep: Finding P2P bots with structured graph analysis," in Proc. USENIX Security, 2010, pp. 1–16.

[10] J. Zhang, X. Luo, R. Perdisci, G. Gu, W. Lee, and N. Feamster, "Boosting the scalability of botnet detection using adaptive traffic sampling," in Proc. 6th ACM Symp. Inf., Comput. Commun. Security, 2011, pp. 124–134.