INTERNATIONAL RESEARCH JOURNAL IN ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET) E - ISSN: 2454-4752 P - ISSN : 2454-4744
VOL 2 ISSUE 6 (2016) PAGES 1286 - 1289
RECEIVED : 29/10/2016. PUBLISHED: 07/11/2016

November 7, 2016

AN AUTOMATIC EFFECTIVE STEALTH ANALYSIS OF P2P BOTNET

[1]S.K.Mohan Raju, [2]V.Geetha,

[1]PG Scholar, Dept Of Computer Science Engineering,Mailam Engineering College,Villupuram,
[2]Assistant Professor Of Computer Science Engineering,Mailam Engineering College,Villupuram.

Abstract:

This work employees complete stopping of the botnet attack made by botmaster. The attack is made by passing the codeword comments by DNS based stealthy mode command and control channel from one system to another system to hijack the server. Usually we can able to identify the attack only after the attack has been made by the botmaster. But by using Botnet Tracking Tool (BTT) we can keep track of the codeword being used. The attack is prevented by making use of the Botnet Tracking Tool (BTT). We continuously monitor the attack made by the botmaster and the bots. The attack is concurrently checked in the database for the pre-defined codeword and if the attack has been found it would be stopped from further attack. If suppose the new codeword is found during the attack that codeword would be stored in the database future use and then isolates them. It does not allow until a proper authorization is made and clarifies them not as bot master.

Keywords: Network security,codewords, DNS security,botnet detection, botnet tracking tool (BTT),command and control.

1. INTRODUCTION

Network security starts with authentication, usually with a username and a password. This requires one detail authentication the user name and the password— this is also called as one-factor authentication. With the twofactor authentication - the user has used (e.g. a security token or dongle, an ATM card or a mobile phone); and with 3-factor authentication the user also used fingerprint or retinal scan. When it is authenticating, a firewall enforces access policies such as the services which are allows the network users to access the network. The effectiveness of preventing the unauthorized access, this component may fail to check potentially harmful content such as computer worms or Trojans being transmitted over the network. Anti-virus software or an intrusion prevention system (IPS) help detect and inhibit the action of such malware. An anomaly-based intrusion detection system may also monitor the network and traffic for network may be logged for audit purposes and for later high-level analysis. Communication between two hosts using a network may be encrypted to maintain privacy [1]. A general concept including as special case such attributes as reliability, availability, safety, integrity, maintainability, etc Security brings in concerns for confidentiality, in addition to availability and integrity Basic definitions are given first They are then commented upon, and supplemented by additional definitions, which address the threats to dependability and security (faults, errors, failures), their attributes, and the means for their achievement (fault prevention, fault tolerance, fault removal, fault forecasting) The aim is to explicate a set of general concepts, of relevance across a wide range of situations and, therefore, helping communication and cooperation among a number of scientific and technical communities, including ones that are concentrating on particular types of system, of system failures, or of causes of system failures.

2. RELATED WORK

2.1. FINDING MALICIOUS DOMAINS USING PASSIVE DNS ANALYSIS

In this paper, we introduce EXPOSURE, a system that employs large-scale, passive DNS analysis techniques to detect domains that are involved in malicious activity. We use 15 features that we extract from the DNS traffic that allow us to characterize different properties of DNS names and the ways that they are queried. Our experiments with a large, real-world data set consisting of 100 billion DNS requests, and a real-life deployment for two weeks in an ISP show that our approach is scalable and that we are able to automatically identify unknown malicious domains that are misused in a variety of malicious activity (such as for botnet command and control, spamming, and phishing)[4].

2.2 DETECTION OF DNS ANOMALIES USING FLOW DATA ANALYSIS

This paper describes algorithms used to monitor and detect certain types of attacks to the DNS infrastructure using flow data. Our methodology is based on algorithms that do not rely on known signature attack vectors. The effectiveness of our solution is illustrated with real and simulated traffic examples. In one example, we were able to detect a tunneling attack well before the appearance of public reports of it[5].

3. EXISTING SYSTEM

Initially an attack by the bot master is made and the after the attack they have identified that an attack has been made. They have checked experimental evaluation makes use of a two-month-long 4.6-GBcampus network data set and 1 million domain names obtained from alexa.com. They have concluded that the DNS-based stealthy command and-control channel (in particular, the code word mode) can be very powerful for attackers, showing the need for further research by defenders in this direction. The statistical analysis of DNS payload as a countermeasure has practical limitations inhibiting its large scale deployment. in this direction. The statistical analysis of DNS payload as a countermeasure has practical limitations inhibiting its large scale deployment. They have been able to identify it only after the attack has been made. .Botnet command-and-control (C&C) channel used by bots and botmaster to communicate with each other, e.g., for bots to receive attack commands and modify from botmaster, a stolen data. A C&C channel for a botnet needs to be reliable one.Many botmaster used the Internet Relay Chat protocol (IRC) or HTTP servers to send information. Botnet operators continuously explore new stealthy communication mechanisms to evade detection.HTTP-based command and control is difficult to distinguish the legitimate web traffic. We do not allow bots to submit DNS queries to eradicate detection. We only allow bots to either piggyback their queries with legitimate DNS queries fromt the host, or follow a query distribution .Our implementation uses the Python Modular DNS Server (pymds) and a designed plug-in to respond to DNS requests. PyMDS implements the full DNS protocol while allowing the user to implement a programmatic and dynamic backend to create the DNS records returned.Instead of returning records from a static file, PyMDS allowed for the decoding of codewords and the generation of appropriate responses.

4. BOTNT SYSTEM TOOL

Botnet tracking tool is implied to detect the botnet attack lively in the network. This tool is used to review the process which is going on. In this the detection of any attack will be detected. It uses machine adoptable learning technique for prevention of forthcoming attacks. This method is used to say completely about the attack which is checked with the database that it is an attack or not. If it is an attack then it will be stopped from further process. If it is found that it is not an attack then it allows it to do the process. Some of the most successful deep learning methods involve artificial neural networks.

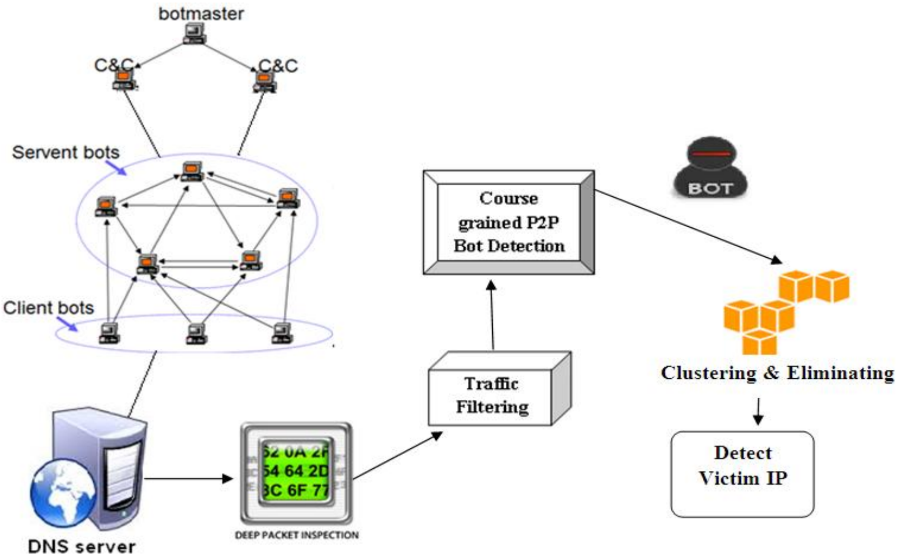

Fig.1.Architecture system

Deep Learning Neural Networks date back at least to the 1980 Neocognitron by Kunihiko Fukushima. It is inspired by the 1959 biological model proposed by Nobel laureate David H. Hubel & Torsten Wiesel, who found two types of cells in the visual primary cortex: simple cells and complex cells. Many artificial neural networks can be viewed as cascading models of cell types inspired by these biological observations. With the advent of the back propagation algorithm, many researchers tried to train supervised deep artificial neural networks from scratch, initially with little success.

5. ANALYSIS

To overcome such drawbacks, Starnberger et al. presented Overbot in [43]. Overbot uses an existing P2P protocol, Kademlia, to provide a stealth command and control channel while guaranteeing the anonymity of the participating nodes and the integrity of the messages exchanged among them. Overbot is the closest work to ours; although we share a similar underlying decentralized structure, there are a number of salient properties that set the two approaches apart. In our approach, the communication bootstrap of a node starts by sending messages using a set of pre-defined nodes–gate nodes (GNs)–that are shipped with the malware. Gate nodes are ordinary bot-infected nodes, and, as such, they perform message routing exactly as any other node. They are just used during the initial bootstrap phase every time a new infected node wants to join the network, but, after that, they do not need to continuously receive communication. On the other hand, sensor nodes in Overbot are a resident part of the botnet: an observer may perform statistical analysis and inferences on the traffic that such nodes generate and receive. Furthermore, Overbot's sensor nodes are equipped with the private key of the botmaster. This means that once a node is compromised, it becomes possible to decrypt all the traffic sent to the botmaster. On the other hand, a compromised node in our approach exposes only its symmetric key, which gives the chance to disclose the traffic sent only by that node.

The parasitic overlay network presented in this paper has all the features required to thwart the current state-of-the-art botnet detection approaches. Message encryption hampers the creation of content-based network signatures, while unknown routing strategies make it difficult to track down IP addresses. In addition, Skype itself makes the network highly resilient to failure and provide a massive user corpus, which gives the chance to rely on a non-negligible number of bots. It is worth noting that speculations on using Skype as a vehicle to build a powerful botnet infrastructure have been around for a while , Fortunately, to the best of our knowledge, such rumors have never evolved into a full-fledged Skype-based botnet in the wild. We have nonetheless shown that such a botnet can be easily designed and implemented. In addition, our simulation and deployment experiments have shown that building a stealthy, resilient and low-cost botnet is indeed possible and practical. Research in botnet detection must thus be refined to deal with the threats posed by such advanced malicious networks that are likely to appear in the near future. Since we believe that the menace posed by the model of botnet presented in this paper will soon emerge, our future works will focus also on the improvement of the host-based detection technique we briefly outlined.

CONCLUSION

Botnet tracking tool experimented by giving attacking code worded messages through the bots network so that server will lively detect the status of the systems that are in communication and those systems also will be under surveillance. Database history will be compared with the coded messages so as to prevent any attacking keywords sent to any secured database. It dynamically updates the current attack takes place by learning the new technique applied.

REFERENCES

[1] http://en.wikipedia.org/wiki/Network_security Ding, W. and Marchionini, G. 1997 A Study on Video Browsing Strategies. Technical Report. University of Maryland at College Park.

[2] http://dl.acm.org/citation.cfm?id=1026492 Tavel, P. 2007 Modeling and Simulation Design. AK Peters Ltd.

[3] http://65.54.113.26/Publication/1436760 Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. J. Mach. Learn. Res. 3 (Mar. 2003), 1289-1305.

[4] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi, “Exposure: Finding Malicious Domains Using Passive DNS Analysis,” Proc.18th Ann. Network and Distributed System Security Symp. (NDSS), Feb. 2011.

[5] A. Karasaridis, K.S. Meier-Hellstern, and D.A. Hoeflin, “Detection of DNS Anomalies Using Flow Data Analysis,” Proc. IEEE GlobeCom, 2006.

[6] C.J. Dietrich, C. Rossow, F.C. Freiling, H. Bos, M. van Steen, and N. Pohlmann, “On Botnets that Use DNS for Command and Control,” Proc. European Conf. Computer Network Defense, Sept. 2011.

[7] E. Kartaltepe, J. Morales, S. Xu, and R. Sandhu, “Social NetworkBased Botnet Command-and-Control: Emerging Threats and Countermeasures,” Proc. Eighth Int’l Conf. Applied Cryptography and Network Security (ACNS).