ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

DEFENDING STEALTHY MODE ATTACK BY LIVE DETECTION AND ADOPTABLE LEARNING TECHNIQUE

Mr. N. Aravindhu, G. Vaishnavi, D. Maheswari

Senoir Assistant Professor, CSE, Christ college of Engineering & Technology, Puducherry,India

Student, CSE, Christ college of Engineering & Technology, Puducherry,India

Student, CSE, Christ college of Engineering & Technology, Puducherry,India

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 two-factor 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[3].

The term bot is short for robot. Criminals distribute malicious software (also known as malware) that can turn your computer into a bot (also known as a zombie). When this occurs, your computer can perform automated tasks over the Internet, without you knowing it. Criminals typically use bots to infect large numbers of computers. These computers form a network, or a botnet. Criminals use botnets to send out spam email messages, spread

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

viruses, attack computers and servers, and commit other kinds of crime and fraud. If your computer becomes part of a botnet, your computer might slow down and you might inadvertently be helping criminals.

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.

To evaluate the piggy back query strategy, our data set is a two-month-long network trace obtained from a university and collected with the IP Audit tool. A static approach is to have a botmaster create an ordered list of domain names and pack the list in malware code for bot to look up, which is same to the use of a one-time password pad for authentication. Botnets have been to use subdirectories for direct communication, However, for a DNS-tunneling-based channel,subdirectory approach does not apply, as the botmaster does not run a web server and the

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

communication is based solely on domain name systems. Considering that botnets often use third-level domains instead of subdirectories,Dagon proposed to use the ratio between second-level domains (SLDs) and third-level domains (3LDs) to identify botnet traffic. DNS-based stealthy messaging systems that requires deep packet inspection and statistical analysis. Deep packet inspection examines packet

payload beyond the packet header.Specifically, we quantitatively analyze the probability distributions of

(bot's) DNS-packet content.

.

.3.1 DRAWBACKS IN EXISTING SYSTEM

- Able to identify a bot master only after an attack has been made.
- It cannot prevent or predict an attack so they can't protect it.
- Did not check it in Live.
- Bot Master cannot be caught red handed.

4. PROPOSED SYSTEM

It uses stochastic implementation of markovs chain link analysis algorithm to correlate with history in database. This method is used to store the new attack which is detected lively during process into the database. A discrete Markov chain model can be defined by the tuple <S,A, lambda;> . S corresponds to the state space, A is a matrix representing transition probabilities from one state to another. λ is the initial probability distribution of the states in S . The fundamental property of Markov model is the dependency on the previous state. If the vector s[t] denotes the probability vector for all the states at time 't', then:

$$\hat{s}(t) = \hat{s}(t-1)\, A \qquad \text{——} \quad (1)$$

If there are 'n' states in our Markov chain, then the matrix of transition probabilities A is of size n x n.Markov chains can be applied to web link sequence modeling. In this formulation, a Markov state can correspond to any of the following:

- URI/URL
- HTTP request
- Action (such as a database update, or sending email)

The matrix A can be estimated using many methods. Without loss of generality, the maximum likelihood principle is applied in this paper to estimate A and λ. Each of the matrix A[s,s']can be estimated as follows:

$$A(s,s') = \frac{C(s,s')}{\sum_{s''} C(s,s'')} \qquad \text{——} \quad (2)$$

$$\lambda(s) = \frac{C(s)}{\sum_{s'} C(s')} \qquad \text{——} \quad (3)$$

C(s,s') is the count of the number of times s' follows s in the training data. Although Markov chains have been traditionally used to characterize asymptotic properties of random variables, we utilize the transition matrix to estimate short-term link predictions. An element of the matrix A, say A[s, s'] can be interpreted as the probability of transitioning from state s to s' in one step. Similarly an element of A*A will denote the probability of transitioning from one state to another in two steps, and so on. Given the "link history" of the user L(t-k), L(t-k+1).... L(t-1) , we can represent each link as a vector with a probability 1 at that state for that time (denoted by i(t-k), i(t-k+1)...i(t-1)). The Markov Chain models estimation of the probability of being in a state at time 't' is shown in equation 4.

$$\hat{s}(t) = \hat{i}(t-1)\, A \qquad \text{——} \quad (4)$$

The Markovian assumption can be varied in a variety of ways. In our problem of link prediction, we have the user's history available; however, a probability

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

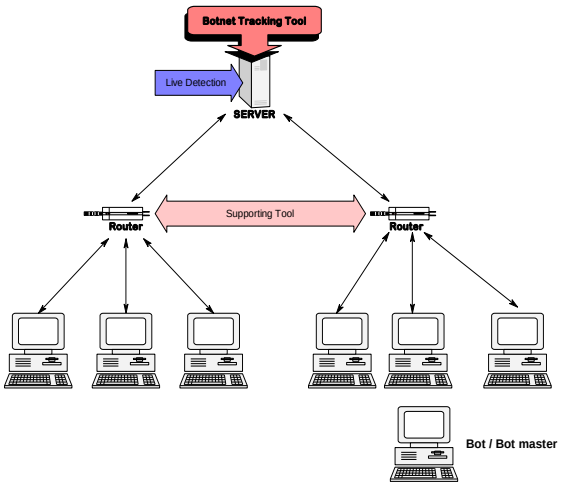
distribution can be created about which of the previous links are "good predictors" of the next link. Therefore we propose vaiancts of the Markov process to accommodate weighting of more than one history state. In the following equations, we can see theat each of the previous links are used to predict the future links and combined in a variety of ways. It is worth noting that rather than compute A*A and higher powers of the transition matrix, these ay be directly estimated using the training data. In practice, the state probablility vector s(t) can be normalized and thresholded inorder to select a list of "probable links/stated" tha the user will choose.

$$\hat{s}(t) = a_0 \hat{i}(t-1)\, A + a_1 \hat{i}(t-2)\, A^2 + a_2 \hat{i}(t-3)\, A^3 \ldots \quad — (5)$$

$$\hat{s}(t) = \mathrm{Max}(a_0 \hat{i}(t-1)\, A,\ a_1 \hat{i}(t-2)\, A^2, a_2 \hat{i}(t-3)\, A^3 \ldots) \quad — (6)$$

4.1 BOTNET TRACKING 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. 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. Sepp Hochreiter's diploma thesis of 1991 formally identified the reason for this failure in the "vanishing gradient problem," which not only affect many-layered feed forward networks, but also recurrent neural networks. The latter are trained by unfolding the into very deep feed forward networks, where a new layer is created for each time step of an input sequence processed by the network. As errors propagate from layer to layer, they shrink exponentially with the number of layers. To overcome this problem, several methods were proposed. One is Jurgen Schmidhuber's multi-level hierarchy of networks (1992) pre-trained one level at a time through unsupervised learning, fine-tuned through back propagation. Here each level learns a compressed representation of the observations that is fed to the next level. Another method is the long short term memory (LSTM) network of 1997 by Hochreiter & Schmidhuber. In 2009, deep multidimensional LSTM networks demonstrated the power of deep learning with many nonlinear layers, by winning three ICDAR 2009 competitions in connected handwriting recognition, without any prior knowledge about the three different languages to be learned. What has attracted the most interest in neural networks is the possibility of learning. Given a specific task to solve, and a class of functions F, learning means using a set of observations to find $f^* \in F$ which solves the task in some optimal sense.



The entails defining a cost function C:F-> IR such that, for the optimal solution $f^*, C(f^*) \leq C(f) \forall f \in F$ - i.e., no

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

solution has a cost less than the cost of the optimal solution(see Mathematical optimization). The cost function C is an important concept in learning, as it is a measure of how far away a particular solution is from an optimal solution to the problem to be solved. Learning algorithm search through the solution space to find a function that has the cost. smallest possible.

4.2 ADVANTAGES OF PROPOSED SYSTEM

- Able to identify bot master before an attack is made.
- Can be in Live Network.
- Tracking tool can identifies the whole chain of network involved in attack.
- Tool created which will isolate the bot master and would not be allowed to be executed at any time.

5.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.

5. ACKNOWLEDGMENTS

Our thanks to the experts who have contributed towards development of the template.

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 Network-Based Botnet Command-and-Control: Emerging Threats and Countermeasures," Proc. Eighth Int'l Conf. Applied Cryptography and Network Security (ACNS).
[8] S. Yadav, A.K.K. Reddy, A.N. Reddy, and S. Ranjan, "Detecting Algorithmically Generated Malicious Domain Names," Proc. 10th Ann. Conf. Internet Measurement (IMC '10).
[9] P. Butler, K. Xu, and D. Yao, "Quantitatively Analyzing Stealthy Communication Channels," Proc. Ninth Int'l Conf. Applied Cryptography and Network Security (ACNS '11).
[10] G. Ollmann, "Botnet Communication Topologies: Understanding the Intricacies of Botnet Command-and Control," https://www.damballa.com/downloads/r_pubs/WP_Botnet Communications_Primer.pdf, 2013.
[11] S. Yadav, A.K.K. Reddy, A.N. Reddy, and S. Ranjan, "Detecting Algorithmically Generated Malicious Domain Names," Proc. 10th Ann. Conf. Internet Measurement (IMC '10), pp. 48-61, 2010.
[12] http://www.microsoft.com/security/resources/botnet-whatis.aspx