Scientific Journal of Impact Factor (SJIF): 4.72

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 4, Issue 3, March -2017

Recovery of files after detection of APT malware using FRR algorithm

Vidya Gholap, Kamini Borkar, Sonali Rajput, Pallavi Patil, Ms. Deepali Chaudhari

[1,2,3,4]*Department of Comp engineering, D. Y. Patil College of engineering*

Abstract — *APT (Advanced Persistent Threat) is a genuine risk to the Internet. With the help of APT malware, attackers can remotely control infected machine and steal the personal information. DNS is well known for malware to find command and control (C&C) servers. The proposed novel system placed at the network departure guide that points toward effectively and efficiently detect APT malware infections based on malicious DNS and traffic analysis. To detect suspicious APT malware C&C domains the system utilizes malicious DNS analysis method, and afterward analyse the traffic of the comparing suspicious IP utilizing anomaly-based and signature based detection innovation. There are separated 14 features in view of big data to describe properties of malware-related DNS. This manufactured a reputation engine to compute a score for an IP address by utilizing these elements vector together***.**

Keywords- *APT, Command and Control, Traffic Analysis, Security. Malware Infections; DNS.*

I. INTRODUCTION

The Advanced Persistent Threat attacks are expanding on the web these days. Unfortunately, they are difficult to detect an APT. It is a continuous or persistent hacking processes and set of stealthy focusing on a particular entity with high-value information, for example, government, military and the monetary business. The aim of an APT assault is to steal the information instead of to make harm the association or system. Once installing so as to hack into the system has been accomplished, APT malware on the contaminated machine by attacker. For instance, APT malware is, Trojan horse or backdoor secondary passage, is customized for firewalls and anti-virus software of the target network. It is not just utilized for remotely controlling the traded off machine in the APT assault, additionally to steal touchy information from extended period of time.

APT malware is altogether different from the worms and bots. The basic role of APT malware is to remotely control the machines and to steal private data, instead of rather than to launch denial-of-service attacks, cause damage or send spam emails. For example, in the case of those worms and bots, the attackers need to use the C&C servers to remotely control thousands of infected host. But APT attackers do not use the same Command & Control server to remotely control so many infected end user machines because it increases the risk of exposure.

For identify malicious domains that involved in APT malware activity is a challenge. The crafted malware in APT attack do not use DGA domains or malicious flux service. The domains for APT malware were registered by the attacker. Compared with these bots and worms the crafted malware requires high degree of stealth. Because of this reason the DNS behavioral features of APT malware are inconspicuous. It is too hard to analyze large volumes of outbound and inbound traffic in a large network, such as an ISP and a large enterprise. Detection of APT malware infections in a big network is another challenging problem.

II. LITERATURE SURVEY

1. detective work Algorithmically Generated Malicious Domain

Authors: Sandeep Yadav, Ashwath K.K. Reddy, and A.L. Narasimha Reddy, SupranamayaRanjanNarus INC.

Description: Recent Botnets like Conficker, Kraken and Torpig have used DNS based“domain fluxing” for command-and-control, where each beast queries for existence of a series of names and additionally the owner should register only 1 such name. throughout this paper, developed a method to find such “domain fluxes” in DNS traffic by probing for patterns inherent to domain names that unit generated algorithmically, in distinction to those generated by humans. especially, we have a tendency to look at distribution of alphanumeric characters additionally as bigrams altogether domains that unit mapped to constant set of IP-addresses. we have a tendency to tend to gift and compare the performance of the many distance metrics, still as KL-distance, Edit distance and Jaccard live. we have a tendency to tend to coach by using a wise information set of domains obtained via a crawl of domains mapped to all or any or any IPv4 address house and modeling dangerous information sets supported behaviors seen to the current purpose and expected. we have a tendency to tend to jointly apply our methodology to packet traces collected at a Tier-1 ISP and show we have a tendency to area unit ready to automatically discover domain fluxing as utilised by Conficker botnet with lowest false positives.

2. Botnet Tracking: Exploring a Root-Cause Methodology to Prevent Distributed Denial-of-Service Attacks
Authors: Felix C. Freiling, Thorsten Holz_, and Georg Wicherski
Description: Denial-of-Service (DoS) attacks create a big threat to the web these days particularly if they're distributed, i.e., launched at the same time at an outsized range of systems. Reactive techniques that try {and} find such an attack and throttle down malicious traffic prevail these days however typically need an extra infrastructure to be very effective. The paper we tend to shows that preventive mechanisms will be as effective with a lot of less effort: Presents associate approach to (distributed) DoS attack interference that's supported the observation that coordinated machine-controlled activity by several hosts wants a mechanism to remotely management them. to forestall such attacks, it's so doable to spot, infiltrate and analyze this remote mechanism and to prevent it in an automatic fashion. we tend to show that this technique will be realised within the net by describing however we tend to infiltrated and tracked IRC-based botnets that ar the most DoS technology utilized by attackers these days.

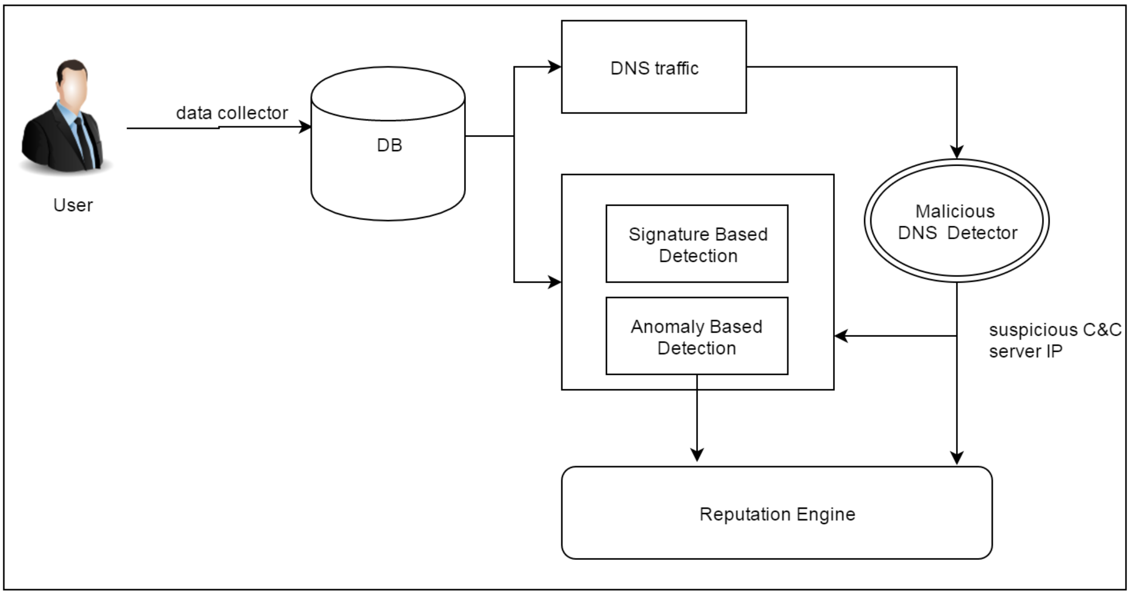
3. A Framework for DNS based detection and mitigation of malware infections on a network
Author: Etienne Stalmans.
Description: Modern botnet trends have cause the utilization of information processing and domain fast-fluxing to avoid detection and increase resilience. These techniques bypass ancient detection systems like blacklists and intrusion detection systems. DNS is one in all the foremost rife protocols on fashionable networks and is crucial for the proper operation of the many network activities, as well as botnet activity. For this reason DNS forms the best candidate for observation, detective work and mitigating botnet activity. during this paper a system placed at the network edge is developed with the potential to observe fast-flux domains victimization DNS queries. Multiple domain options were examined to see which might be best within the classification of domains. this is often achieved employing a C5.0 call tree classifier and Bayesian statistics, with positive samples being tagged as probably malicious and negative samples as legitimate domains. The system detects malicious domain names with a high degree of accuracy, minimising the requirement for blacklists. applied math ways, specifically Naive Bayesian, Bayesian, Total Variation distance and likelihood distribution area unit applied to observe malicious domain names. The detection techniques area unit tested against sample traffic and it's shown that malicious traffic is detected with low false positive rates.

4. An Empirical Reexamination of Global DNS Behavior
Author: HongyuGao, Vinod Yegneswaran, Yan Chen, Phillip Porras, Shalini Ghosh, Jian Jiang, HaixinDuan.
Description: The performance and operational characteristics of the DNS protocol area unit of deep interest to the analysis and network operations community. during this paper, we have a tendency to gift mensuration results from a novel dataset containing quite twenty six billion DNS query-response pairs collected from quite 600 globally distributed algorithmic DNS resolvers. we have a tendency to use this dataset to affirm findings in printed work and see some vital variations that might be attributed each to the evolving nature of DNS traffic and to our differing perspective. for instance, we discover that though characteristics of DNS traffic vary greatly across networks, the resolvers inside a company tend to exhibit similar behavior. we have a tendency to more realize that quite five hundredth of DNS queries issued to root servers don't come productive answers, which the first reason behind search failures at root servers is distorted queries with invalid TLDs. moreover, we have a tendency to propose a unique approach that detects malicious domain teams exploitation temporal correlation in DNS queries. Our approach needs no comprehensive labeled coaching set, which may be tough to make in observe. Instead, it uses a familiar malicious domain as anchor, and identifies the set of antecedently unknown malicious domains that area unit associated with the anchor domain. Experimental results illustrate the viability of this approach, i.e. , we have a tendency to attain a real positive rate of quite ninety six, and every malicious anchor domain leads to a malware domain cluster with quite fifty three antecedently unknown malicious domains on the average.

5. Signature Based Intrusion Detection System Using SNORT
Author: Vinod Kumar, Dr. Om Prakash Sangwan.

Description: Now a day's Intrusion Detection systems plays important role in Network security. because the use of net is growing quickly the chance of attack is additionally increasing in this quantitative relation. individuals square measure victimization signature primarily based IDS's. Snort is usually used signature primarily based IDS as a result of it's open supply software package. World wide it's employed in intrusion detection and bar domain. Basic analysis and security engine (BASE) is additionally wont to see the alerts generated by Snort.

III.SYSTEM ARCHITECTURE



IV. MATHEMATICAL MODEL

Let W is the set of whole of system which consists:
W= {input, process, output}.
Input={D, MDNS, RE, NTA}
Where,

1. D is the set of data collector.
2. MDNS is the set o malicious DNS detector which detects the malicious IP at DNS server traffic.
3. NTA is the network traffic analyzer which detects the network traffic.
4. RE is the reputation engine which calculates the reputation score of an IP address

Process:
1. Data Collector: Data Collector placed at the network edge to record the inbound and outbound traffic produce by the network.
2. Malicious DNS Detector: Malicious DNS Detector is responsible for analyzing the outbound and inbound DNS traffic produced by the network, and detecting suspicious APT malware Command & Control domains. It detects the suspicious APT malware-related domains and provide corresponding suspicious Command & Control server IP addresses for the „network traffic analyzer" of the system.
3. Reputation Engine: The aim of reputation engine is to compute a reputation score for an IP address to judge whether the server or host owning the IP address is infected or not by using malicious DNS and network traffic feature vectors together.
4. Network Traffic Analyzer: It consists of anomaly-based detector and signature-based detector for analyzing the network traffic of suspicious Command & Control server IP addresses. Signature-based detector has defined C&C communication traffic signatures for detection of malware known to the system. And the anomaly-based detector detects anomalous behaviors including statistical anomaly, protocol anomaly, and application anomaly. When the new or

unknown malware was identified by anomaly-based detector will be defined the new signatures. And all the Command & Control communication traffic signatures which have identified will be collected to TM (Targeted Malware) family.

V. PROPOSED ALGORITHEM

1. AES Algorithm:

The AES-256 algorithm is composed of three main parts: Cipher, Inverse Cipher and Key Expansion. Cipher converts data to an unintelligible form called cipher text while Inverse Cipher converts data back into its original form called plaintext. Key Expansion generates a Key Schedule that is used in Cipher and Inverse Cipher procedure. Cipher and Inverse Cipher are composed of specific number of rounds For both its Cipher and Inverse Cipher, the AES algorithm uses a round function that is composed of four different byte-oriented transformations:
1) Byte substitution using a substitution table (S-box)
2) Shifting rows of the State array by different offsets
3) Mixing the data within each column of the State array
4) Adding a Round Key to the State
The Cipher transformations can be inverted and then implemented in reverse order to produce a straightforward Inverse Cipher for the AES algorithm. The individual transformations used in the Inverse Cipher.
1) Inverse Shift Rows
2) Inverse Sub Bytes
3) Inverse Mix Columns
4) Add Round Key

The AES inverse cipher core consists of a key expansion module, a key reversal buffer, an initial permutation module, a round permutation module and a final permutation module. The key reversal buffer first store keys for all rounds and the presents them in reverse order to the rounds. The round permutation module will loop maternally to perform 14 iterations (for 256 bit keys).

VI.CONCLUSION

In this, a proposed a system IDnS placed at the network egress points to detect malware infections inside the network combined with DNS traffic analysis. Extracted new features and built a reputation engine based on big data, which includes approximately 400 million DNS queries. The system processes advantages of high efficiency and accuracy. The experimental results show that this security approach is feasible for improving the sustainability of the system and is good at detecting APT malware infections. It is a useful intrusion system that can help to fight against cyber-crime such as theft of information from infected host

ACKNOWLEDGMENT

We might want to thank the analysts and also distributers for making their assets accessible. We additionally appreciative to commentator for their significant recommendations furthermore thank the school powers for giving the obliged base and backing.

REFERENCES

[1] F. C. Freiling, T. Holz, and G. Wicherski, ``Botnet tracking: Exploring a root-cause methodology to prevent distributed denial-of-service attacks," *Lect. Notes Comput. Sci.*, vol. 10, no. 2, pp. 319_335, 2005.

[2] A. Karasaridis, B. Rexroad, and D. Hoe_in, ``Wide-scale botnet detection and characterization," in *Proc. 1st Conf. 1st Workshop Hot Topics Understand. Botnets*, 2007, p. 7.

[3] J. Jung, M. Konte, and N. Feamster, ``Dynamics of online scam hosting infrastructure," in *Proc. 10th Int. Conf. Passive Active Netw.Meas.*, 2009, pp. 219_228.

[4] H. Porras, H. Saïdi, and V. Yegneswaran, ``A foray into Con_cker's logic and rendezvous points," in *Proc. USENIX Conf. Large-Scale Exploits Emergent Threats, Botnets, Spyware, Worms, More*, 2009, p. 7.

[5] S. Yadav, A. K. K. Reddy, A. L. N. Reddy, and S. Ranjan, ``Detecting algorithmically generated malicious domain names," in *Proc. ACMSIGCOMM Conf. Internet Meas.*, 2010, pp. 48_61.

[6] J. Wolf. (2008). *Technical Details of Srizbi's Domain Generation Algorithm*.[Online]. Available: http://tinyurl.com/6mdasc

[7] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi, ``EXPOSURE: Finding malicious domains using passive DNS analysis," in *Proc. NDSS*, 2011.

[8] E. Stalmans and B. Irwin, ``A framework for DNS based detection and mitigation of malware infections on a network," in *Proc. Inf. Secur. South Africa (ISSA)*, Aug. 2011, pp. 1_8.

[9] M. Antonakakis, R. Perdisci, D. Dagon,W. Lee, and N. Feamster, ``Building a dynamic reputation system for DNS," in *Proc. 19th USENIX Secur. Symp.*, 2010, pp. 273_290.

[10] LASTLINE. (2015). *Using Passive DNS Analysis to Automatically Detect Malicious Domains*. [Online]. Available: https://www.lastline. com/papers/dns.pdf