ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Protecting DNS Query Communication against DDoS Attacks

Ms. R. Madhuranthaki[1], Ms. S. Umarani, M.E., (Ph.D) [2]

II M.Tech (IT), IT Department, Maharaja Engineering College, Avinashi, India[1]

HOD, IT Department, Maharaja Engineering College, Avinashi, India[2]

Abstract: The server resources are abnormally consumed by the attackers using the denial-if service attacks. Denial-of-Service denies a victim from providing or receiving normal services. Distributed Denial of Service (DDoS) Attacks are generated in a "many to one" dimension. In DDoS attack model large number of compromised host are gathered to send useless service requests, packets at the same time. Attackers select the hidden channel model for their communication. A C&C channel for a botnet needs to be reliable, redundant, noncentralized and easily disguised as legitimate traffic. Domain Name Service (DNS) provides a distributed infrastructure for storing, updating and disseminating data. DNS is targeted as a stealthy botnet command-and-control channel. Malicious DNS activities are hiding at the network level. Exponentially Distributed Query and Piggybacking Query attacks are detected using the markov chain analysis and statistical analysis mechanism. Probability distribution based analysis model is used to detect automatic domain flux attacks. DNS tunneling technique is used for transmitting arbitrary data via DNS protocol. The attack detection system is improved with security and privacy factors. Automated anomaly detection is adapted to the system. Navy bayesian classification technique is integrated to the system. Small query analysis mechanism is integrated with the system.

I. INTRODUCTION

A denial of service attack is characterized by an explicit attempt by an attacker to prevent legitimate users of a service from using the desired resources. Examples of denial of service attacks include:

- attempts to "flood" a network, thereby preventing legitimate network traffic
- attempts to disrupt connections between two machines, thereby preventing access to a service
- attempts to prevent a particular individual from accessing a service
- attempts to disrupt service to a specific system or person.

The distributed format adds the "many to one" dimension that makes these attacks more difficult to prevent. A distributed denial of service attack is composed of four elements. First, it involves a victim, i.e., the target host that has been chosen to receive the brunt of the attack. Second, it involves the presence of the attack daemon agents. These are agent programs that actually conduct the attack on the target victim. Attack daemons are usually deployed in host computers. These daemons affect both the target and the host computers. The task of deploying these attack daemons requires the attacker to gain access and infiltrate the host computers. The third component of a distributed denial of service attack is the control master program. Its task is to coordinate the attack. Finally, there is the real attacker, the mastermind behind the attack. By using a control master program, the real attacker can stay behind the scenes of the attack. The following steps take place during a distributed attack:

1. The real attacker sends an "execute" message to the control master program.
2. The control master program receives the "execute" message and propagates the command to the attack daemons under its control.
3. Upon receiving the attack command, the attack daemons begin the attack on the victim.

Although it seems that the real attacker has little to do but sends out the "execute" command, he/she actually has to plan the execution of a successful distributed denial of service attack. The attacker must infiltrate all the host computers and

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

networks where the daemon attackers are to be deployed. The attacker must study the target's network topology and search for bottlenecks and vulnerabilities that can be exploited during the attack. Because of the use of attack daemons and control master programs, the real attacker is not directly involved during the attack, which makes it difficult to trace who spawned the attack.

II. RELATED WORK

Related work can be grouped in three categories. First, several C&C techniques of botnets have been analyzed in depth [4], [7]. For example, Holz et al. [5] provide insights into botnets with peer-to-peer C&C architecture in a case study on the storm worm. Stock et al. and Calvet et al. [3], [11] analyze the Waledac peer-to-peer botnet in detail. Additionally, we discuss general architectural issues and limitations of DNS botnet C&C. The second group of related work contains approaches to detect botnets in network traffic. This kind of related work can be separated into application protocol specific approaches and protocol independent approaches. As HTTP is used as botnet C&C, some work has been done to develop specifically tailored detection methods for HTTP-based botnet C&C, such as Perdisci et al. [2]. Goebel and Holz [1] present methods in order to detect IRC-based botnets. However, due to their protocol-dependent orientation, none of these approaches are able to detect DNS C&C.

Independent of the application layer protocol, Gu and Strayer [10] propose botnet detection methods based on network flow characteristics. However, protocol-independent approaches will likely fail to detect DNS C&C as they expose neither chat-style characteristics nor necessarily spatial-temporal correlated behavior. For example, Feederbot does neither exhibit periodicity nor synchronized transactions among different bot executions – effectively exploiting the gap of existing detection approaches. Therefore, we provide a detection method specifically tailored to DNS C&C based on rdata and behavioral communication features – successfully filling this gap.

A special case is the work of Choi et al. [8], which – though not specifically targeting DNS C&C – addresses group activities in DNS. The authors define differences between DNS query behavior typical to any kind of botnet and legitimate DNS resolution. According to Choi et al., key features for bot-typical behavior include simultaneous DNS queries, quickly changing C&C server addresses as well as transient domains. However, as our analysis of Feederbot discovers, none of these assumptions hold. Feederbot's C&C servers stayed up for the whole monitoring period of nine months and DNS queries are not synchronized between different bots. Instead, we exploit rdata features and persistent communication behavior to detect DNS C&C. The third group of related work covers DNS covert communication. Bernat [9] analyzed DNS as covert storage and communication medium. Born and Gustafson [6] employ character frequency analysis in order to detect DNS tunnels. However, both approaches do not specifically address the detectability of DNS as botnet C&C. In addition, we significantly improve entropy-based features and combine them with behavioral features to target botnets.

III. DNS CONCEPTS

The Domain Name System (DNS) is a hierarchical naming system for computers, services, or any resource connected to the Internet. Clearly, as it helps Internet users locate resources such as web servers, mailing hosts, and other online services, DNS is one of the core and most important components of the Internet. Unfortunately, besides being used for obvious benign purposes, domain names are also popular for malicious use. In a typical Internet attack scenario, whenever an attacker manages to compromise and infect the computer of an end-user, this machine is silently transformed into a bot that listens and reacts to remote commands that are issued by the so-called botmaster. Such collections of compromised, remotely-controlled hosts are common on the Internet, and are often used to launch DoS attacks, steal sensitive user information, and send large numbers of spam messages with the aim of making a financial profit. In another typical Internet attack scenario, attackers set up a phishing website and lure unsuspecting users into entering sensitive

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

information such as online banking credentials and credit card numbers. The phishing website often has the look and feel of the targeted legitimate website and a domain name that sounds similar.

One of the technical problems that attackers face when designing their malicious infrastructures is the question of how to implement a reliable and flexible server infrastructure, and command and control mechanism. Ironically, the attackers are faced with the same engineering challenges that global enterprises face that need to maintain a large, distributed and reliable service infrastructure for their customers. For example, in the case of botnets, that is arguably one of the most serious threats on the Internet today, the attackers need to efficiently manage remote hosts that may easily consists of thousands of compromised end-user machines. Obviously, if the IP address of the command and control server is hard-coded into the bot binary, there exists a single point of failure for the botnet. That is, from the point of view of the attacker, whenever this address is identified and is taken down, the botnet would be lost. Analogously, in other common Internet attacks that target a large number of users, sophisticated hosting infrastructures are typically required that allow the attackers to conduct activities such as collecting the stolen information, distributing their malware, launching social engineering attempts, and hosting other malicious services such as phishing pages.

In order to better deal with the complexity of a large, distributed infrastructure, attackers have been increasingly making use of domain names. By using DNS, they acquire the flexibility to change the IP address of the malicious servers that they manage. Furthermore, they can hide their critical servers behind proxy services so that their malicious server is more difficult to identify and take down. Using domain names gives attackers the flexibility of migrating their malicious servers with ease. That is, the malicious "services" that the attackers offer becomes more "fault-tolerant" with respect to the IP addresses where they are hosted.

IV. ATTACKS USING DNS

From the point of view of a botmaster, a trade-off between C&C communication visibility and the botinherent need to communicate arises. On the one hand, bots must communicate with their C&C instance to receive instructions and transmit data such as stolen credentials. On the other hand, botmasters try to hide the C&C traffic in order to avoid detection. Usually, the design of a botnet's C&C results in messages being obfuscated or encrypted so that it is more difficult to detect and understand the semantics of certain types of C&C traffic.

4.1. DNS as carrier for Botnet C&C

Whereas several application layer protocols have been analyzed by the research community concerning the usage as a basis for botnet command and control, to our knowledge we are the first to openly analyze the Domain Name System protocol as carrier for botnet C&C. DNS, when compared to other application layer protocols provides some advantages. Concerning its usage as botnet C&C, DNS has not been seen so far. Thus, botnets using DNS as C&C benefit from the fact that currently there is no specifically tailored detection mechanism, which in turn, raises the probability for the botnet to remain undetected. Even in environments with heavily restricted Internet access, e.g. by means of firewalling and proxying, DNS is usually one of the few protocols – if not the only one – that is allowed to pass without further ado. Furthermore, whereas for some protocols such as HTTP, there are a number of existing methods to analyze and inspect the network traffic like the one presented by Perdisci et al. DNS is usually served "as is". As another advantage, DNS was designed as a distributed system and as such provides advantages in terms of resilience. We can only speculate as to why Feederbot avoids the pre-configured resolver and directly queries its DNS servers. One reason could be that in this way, the corresponding DNS C&C transactions leave no traces in DNS resolver logs, caches or passive DNS databases. In contrast, the fact that a different than the pre-configured DNS resolver is used, might itself be suspicious enough to catch one's eye – especially in homogeneous environments.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

4.2. Segmentation and Encryption

Feederbot's C&C traffic is split into message chunks with a maximum length of 220 bytes per chunk. One message chunk is transmitted in the rdata field of a TXT resource record in the DNS response. In order to evade detection, most of the message chunks are encrypted using the stream cipher RC4. Feederbot uses a variety of different encryption keys. A specific part of the DNS query domain name is used to transmit parameters for key derivation. As an example, one such parametrized key derivation function takes as input a substring of the query domain name qdparam. Structure of a Feederbot DNS C&C Message Chunk This substring qdparam is then RC4-encrypted with the string "feedme" and the result is used to initialize the RC4 decryption of the actual C&C message chunks. The stream cipher is used in a stateful manner, so that if a message chunk gets lost, decryption of subsequent message chunks will fail. In addition, Feederbot's C&C message chunks make use of cyclic redundancy checks to verify the decryption result. The CRC32 checksum preceeds message chunk payload and is not encrypted. Using the results from the dynamic analysis in Sandnet as well as the reverse engineering efforts, we achieved the implementation of a passive decryption utility in order to decrypt Feederbot's DNS C&C messages. Additionally, we implemented a low interaction clone of Feederbot to actively analyze its C&C. Feederbot receives instructions from several DNS C&C servers. Initially, when Feederbot is launched, a request is sent to a very small subset of C&C servers. These servers seem to serve as bootstrapping C&C servers. During our monitoring period of nine months, we have seen only two such DNS servers acting as bootstrapping C&C servers. The response to this initial bootstrapping request message is not encrypted and only base64 encoded. It contains a pointer to at least one other C&C server as well as another domain name. Subsequent communication is encrypted.

V. ISSUES ON DNS COMMUNICATION METHODS

Botnet controllers use stealthy messaging systems to set up large-scale command and control requests. A C&C channel for a botnet needs to be reliable, redundant, noncentralized and legitimate traffic. Domain Name Service (DNS) provides a distributed infrastructure for storing, updating and disseminating data. DNS is targeted as a stealthy botnet command-and-control channel. Malicious DNS activities are hide at the network level. Exponentially Distributed Query and Piggybacking Query attacks are detected using the markov chain analysis and statistical analysis mechanism. Probability distribution based analysis model is used to detect automatic domain flux attacks. DNS tunneling technique is used for transmitting arbitrary data via DNS protocol. The following issues are identified in the current DNS Communication Schemes.

- DNS sensitive data access is not controlled
- Data leaks are not accurately detected
- User intention is required for anomaly detection
- Detection latency is high

VI. PROTECTING DNS QUERY COMMUNICATION

The Domain Name Service based attack detection system is designed to handle command and control message process over DNS query values. The DNS query analysis is performed with statistical analysis. The system analyzes the DNS query values to identify the insertion of DDoS attack detection information. Machine learning approach is used to detect the DDoS attacks. The system is divided into five major modules. They are request observer, statistical analysis, Bayesian analysis, command and control request handler and security and privacy process.

The request observer module is designed to collect service requests from the clients. Statistical analysis module is designed to detect DDoS attacks using statistical methods. Bayesian approach based attack detection mechanism uses the classification process. The command and control handler is designed to manage the DNS query based botnet communication activities. Security and privacy module is designed to provide security and privacy for DSN parameters.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

6.1. Request Observer

Stream requests are collected from various clients. The server assigns session instances to new stream requests. The stream requests are processed by the web server. The responses are redirected to the clients.

6.2. Statistical Analysis

The statistical analysis model is used to detect the Service attacks. The request flow and its similarity are analyzed with frequency and interval values. The user session patterns are learned from the request flow analysis. Attack decisions are made with the support of DNS query format. The requests are assigned with normal or attack labels in the analysis.

6.3. Bayesian Analysis

The Bayesian analysis model is used to detect DNS query based attacks using machine learning approaches. The classification model is used for the request category identification process. The Bayesian classification algorithm is used for the attack detection process. The attacker requests are rejected by the service provider.

6.4. Command and Control Request Handler

The command and control requests are exchanged between the botmaster and bots. The DNS query values are used to exchange the command and control requests. The DNS query responses are verified with the domain name server details. Command and control instructions are dropped with reference to their category.

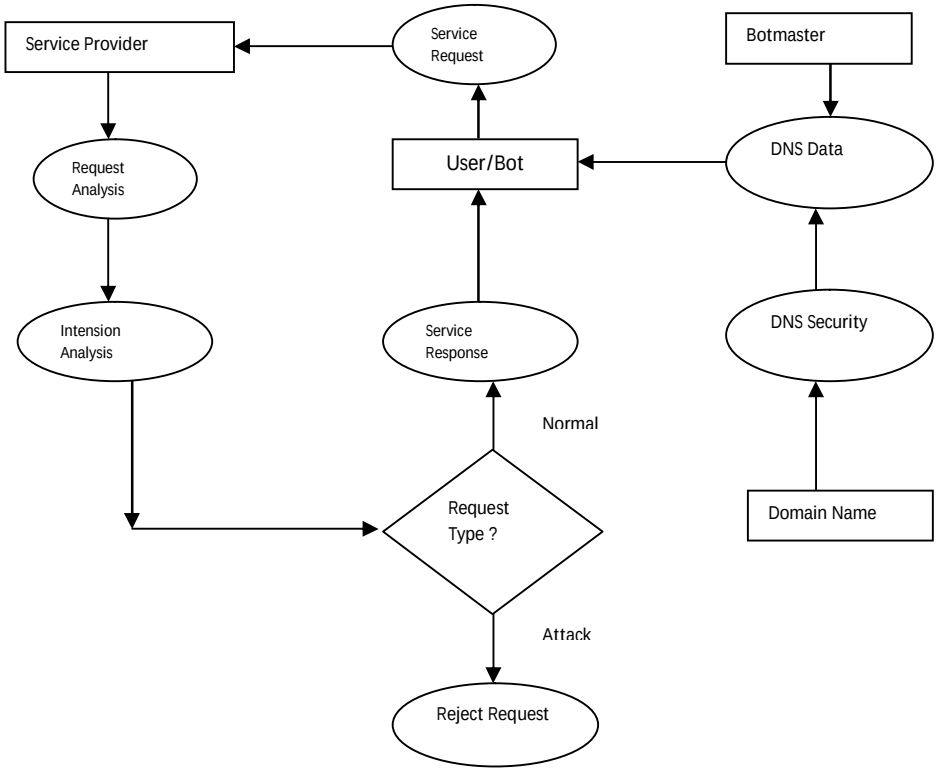

Fig. No: 6.1. Protecting DNS Query Communication

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

6.5. Security and Privacy

The DNS query values are submitted from the clients in the networks. The DNS responses are prepared by the domain name server. The user can add additional parameters to the DNS responses. The RSA algorithm is used to protect the DNS parameter values. Sensitive attribute values are protected with cryptography methods.

VII.CONCLUSION

The DNS query based attack detection scheme is enhanced to provide privacy preserved data traffic analysis. Automated anomaly detection is adapted to the system. Naiva Bayesian classification technique is integrated to the system. Small query analysis mechanism is integrated with the system. The system performs botnet communication detection and control operations. DDoS attack detection mechanism is included in the system. The system improves the detection accuracy with minimum latency. DNS parameter security is also provided by the system.

REFERENCES

[1] J. Goebel and T. Holz, "Rishi: Identify Bot Contaminated Hosts by IRC Nickname Evaluation," in USENIX HotBots, 2007.
[2] R. Perdisci, W. Lee, and N. Feamster, "Behavioral Clustering of HTTP-Based Malware and Signature Generation Using Malicious Network Traces," in NSDI, 2010.
[3] B. Stock, M. Engelberth, F. C. Freiling, and T. Holz, "Walowdac Analysis of a Peer-to-Peer Botnet," in EC2ND, 2009.
[4] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna, "Your Botnet is My Botnet: Analysis of a Botnet Takeover," in CCS, 2009.
[5] T. Holz, M. Steiner, F. Dahl, E. Biersack, and F. Freiling, "Measurements and Mitigation of Peer-to-Peer-based Botnets: A Case Study on Storm Worm," in USENIX LEET, 2008.
[6] K. Born and D. Gustafson, "Detecting DNS Tunnels Using Character Frequency Analysis," 2011.
[7] K. Chiang and L. Lloyd, "A Case Study of the Rustock Rootkit and Spam Bot," in HotBots, 2007.
[8] H. Choi, H. Lee, H. Lee, and H. Kim, "Botnet detection by monitoring group activities in DNS traffic," in CIT, 2007.
[9] D. Bern´at, "Domain name system as a memory and communication medium," in SOFSEM, 2008.
[10] W. T. Strayer, D. E. Lapsley, R. Walsh, and C. Livadas, "Botnet detection based on network behavior," in Botnet Detection, ser. Advances in Information Security, W. Lee, C. Wang, and D. Dagon, Eds. Springer, 2008.
[11] C. R. D. Joan Calvet and P.-M. Bureau, "Malware Authors Don't Learn and That's Good!" in MALWARE, 2009.