Glimpse of Bonet: Analysis, Detection and Defense

Sanket N Patel, Tarulata Chauhan
Department Of Computer Engineering
L.J. Institute of Engineering & Technology,Ahmedabad-382210,
Gujarat, India

Abstract-Botnets are a well-recognized and persistent therat to all users of the Internet. Botnets have developed form a subject of curiosity to highly sophisticated instrument for illegally earning money. In parallel, an underground economy has developed which creates hundreds of millions of Euros per year in revenue with spamming. Botnets facilitate an efficient generation and guaranteed delivery of large volumes of spam. Spam bots, or spam-generating bots,use different transmission methods based on the network settings of the infected host. These methods include relaying, proxying, and direct delivery. Hundreds of thousands of new malware samples per month pose an immense challenge for AV companies. Specialized countermeasures against botnets have evolved along with botnet technology, trying to bring them down by targeting the root of every botnet: its command-and-control structure. The owners of some botnets, such as storm worm, torping and conflicker,are employing fluxing techniques to evade detection. This paper provides analysis of Botnet attacks, Botnet detection techniques and Botnet mitigation techniques.

Keyword-Botnet; Spam; Fast Fluxing; Domain Fluxing; C&C

1.Introduction

In recent years, cyber attacks have evolved, becoming more profit-centered and better organized. Emails spams, click frauds, and social phishing are becoming more and more popular. Botnet, which consists of a network of compromised computers connecting to the Internet and controlled by a remote attacker, are for all of these problems. A botmaster can drive numerous compromised computers to attack the target at the same time. Therefore, as an attack platform, a botnet can cause serious damage and is very hard to defend against.

Compare to other attack vectors, the essential component of a botnet is its control and command channel. A C&C is used to update the bot program, distribute commands from the botmasters, and collect victims' private information, such as bank accounts and identifying information. Once a C&C is broken down, the botnet degrades to discrete, unorganized infections, which are easy to be eliminated using host-based cleanup technology.

1.1 The Botnet Lifecycles

(A)Infection: initial installation of botnet malware on target host. this is done by tricking users into running executables to attached to email. By exploiting their browser weakness or exploiting the presence of security holes.

(B)Bootstrapping and Maintenance: each node has to perform a set of actions to detect the presence of other nodes and connect to them, bot controller must be able to counteract when its associated nodes leave the botnets. Such maintenance operation have a fundamental role in ensuring robustness.

(C)Command & Control: botmaster and controller have the necessity of reliably distributing their command(e.g by sending the command start ddos target=192.133.0.10 or send spam mail templates bot software updates) to their controlled nodes. that in turn to send associated results, or current status into back to bot master.

(D)Command Execution: running the received command on each individual bo**t.**

1.2 Command and Control Architecture

A second core problem for botnet attackers is how to communicate with each bot instance. Most attackers would like the ability to rapidly send instructions to bots but also do not want that communication to be detected or the source of the those commands to be revealed.

(A)Centralized: A centralized topology is characterized by a central point that forwards messages between clients. Messages sent in a centralized system tend to have low latency as they only need to transit a few well-known hops. From the perspective of an attacker, centralized systems have two major weaknesses: they can be easier to detect since many clients connect the same point, and the discovery of the central location can compromise the whole system.

(B)P2P: Peer-to-peer (P2P) botnet communication has several important advantages over centralized networks. First, a P2P communication system is much harder to disrupt. This means that the compromise of a single bot does not necessarily mean the loss of the entire botnet. However, the design of P2P systems are more complex and there are typically no guarantees on message delivery or latency.

(C)Unstructured: A botnet communication system could also take the P2P concept to the extreme and be based on the principle that no single bot would know about any more than one other bot. In such, a topology a bot or controller that wanted to send a message would encrypt it and then randomly scan the Internet and pass along the message when it detected

another bot. The design of such a system would be relatively simple and the detection of a single bot would never compromise the full botnet. However, the message latency would be extremely high, with no guarantee of delivery.

1.3 Spam Transmission Methods

Legitimate path for email sending is like this

Client→MTA→MX→Other client now instead of following this legitimate path botnet follow below transmission methods

(A)Open relay: An open relay is an SMTP server that accepts relay requests from any source to any destination, open relay was the most common method used by spammers because it was the default behaviour of any SMTP server.

(B)Open proxy: proxy is basically hide the IP address of client. Now open proxy allow connection between any client and any sever. By using open proxy attacker can hide the IP address of bot who send suspicious mail.

In order for spam bots to utilize this service, they need to have IP address of open proxies. This can be achieved by either a network scan or by a downloading from the controlling server.

(C)Proxy Lock: as a special case, some spam bots request the proxy to forward email packets to the MX server of the proxy's domain. This feature is called Proxy lock. In this case, it is the proxy responsibility to look up the MX record of its own domain.This method achieves more than one goal; first,it makes the spam message look more legitimate because it has been relayed by a legitimate and trustworthy mail sever. It also decreases the effort made by spambots to find a relay mail server.

(D)Direct-to-MX: botnet generated spam can be delivered directly to the MX server of the recipient's domain. For legitimate emails and according to SMTP protocol, direct delivery is done bye the MTAs, not by end users or their MUAs. MTAs query the DNS for the MX record of the recipient's domain and then deliver the message to that sever.

Direct delivery is favoured by spamming botnets because it reduces the chance of filtering out spam messages by intermediate relays.

1.4 New techniques in botnet to evade detection

Feily et al. surveyed botnet mechanisms and botnet detection techniques based on different classes they identified: signature based, anomaly-based, DNS-based and mining-based. they also compared and evaluated the advantages and disadvantages of some typical researches from each category. In our survey, we focus on two advance botnet mechanisms:

(A)Fast flux(FF): A mechanism that a set of IP addresses change frequently corresponding to a unique domain name.
(B)Domain flux(DF): A mechanism that a set of domain names are generated automatically and periodically corresponding to a unique IP address.

The rest of this paper is structured as follows. In section II, we focus on different botnet attacks. Following that we introduce how recent approaches were developed to detect botnet in section III. In section IV, we briefly introduce the Defense and takedown techniques. At last, in section V provide the conclusion.

2.Botnet Attacks

(A) Attacking IRC Networks: Botnets are used for attacking IRC networks. The victim is flooded by service request from thousands of Bots and thus victim IRC network is brought down.
(B) Distributed Denial of Services (DDoS): DDoS is a attack on a computer system or network that causes a loss of services/network to users by consuming the bandwidth of the victim network. The resource path is exhausted if the DDoS- attack causes many packets per second (PPS). The DDoS attacks are not limited to Web servers, virtually any service available on the internet can be target of such an attack. Higher level protocols can be misused to increase the load even more effectively by using very specific attacks such as such as running exhausting search queries on bulletin boards or recursive HTTP-floods on the victim's website called spidering.
(C) Key Logging: With the help of a key logger it is very easy for an attacker to retrieve sensitive information. There exists filtering mechanism that aid in stealing secret data.
(D) Sniffing Traffic: Bots can also use a packet sniffer to watch for clear text data passing by compromised machine. The sniffers are used to retrieve sensitive information such as usernames and passwords.
(E) Spamming: Some bots can open a SOCKS v4/v5 proxy—a generic proxy protocol for TCP/IP-based networking applications—on a compromised machine. After having enabled the SOCKS proxy, this machine can then be used for nefarious tasks such as spamming. With the aid of Botnet, an attacker can then send massive amounts of bulk e- mail (spam). Some Bots also harvest e-mail addresses (by opening a SOCKS v4/v5 proxy).

(F) Advertisement Installation: BotNets setup a fake web site with some advertisements. The operator of this website negotiates a deal with some hosting companies that pay for clicks on ads. With the help of Botnet, these clicks can be 'automated' so that instantly a few thousands Bots clicks on the pop-ups, hijacks the start page of a compromise machine so that the 'clicks' are executed each time the victim uses the browser.

(G) Spreading New Malware: This is easy since all Bots implement mechanisms to download and execute a file via HTTP or FTP. Thus, spreading virus via e-mail is very easy using a Botnet.

(H) Manipulating Online Polls or Games: These are very easy to manipulate due to high attention. Since every Bot has a distinct IP address and do the manipulation. Every vote will have the same credibility as a vote cast by a real person. Online games can be manipulated in a similar way.

(I) Mass Identity Theft: By combining above different functions, they are used for large scale identity theft which is one of the fastest growing crimes on the internet. Bogus e- mails that pretend to be legitimate (such as banking e-mail) ask their internet victims to go on line and submit their private information. The fake e-mail are generated and sent by Bots via their spamming mechanism. These Bots can also host multiple fake web sites and as and when one of these fake sites is shut down, another one can pop up. In addition, key logging and sniffing of traffic can also be used for identity theft.

3. Botnet Detection

3.1 Botnet Fast flux Detection Techniques

Holz et al. claimed that they were the first to develop a metric to detect fast-flux service network (FFSN) empirically. They identified three possible parameters which can be used to distinguish normal network behaviours and FFSNs - the number of IP-domain mappings in all DNS lookups, the number of nameserver records in one single domain lookup, and the number of autonomous system in all IP-domain pairs. Based on these three parameters, they defined a metric - flux-score, which was a result of a linear decision function to predict the FFSN. A higher score indicates a higher fluxing degree, and vice versa. They evaluated their metric by a 10-fold validation using a two-month observation data. Results show that their method can distinguish normal network behaviour from FFSN with a very low false positive rate [1].

There are some limitations in detection methods which focus on detecting domains that are related to IP addresses with short TTL in DNS query results in [1] [2]. In 2009, Zhou et al. overcame these limitations by applying a behaviour-analysis model [3]. To achieve this, they began with characterizing the behaviours of FF domains at some points around these FF domains. Based on the analysis of those behaviours, they presented an analytical model which showed the number of DNS queries required to confirm an FF domain. In addition, they speeded up the detection by two schemes. One scheme is to associate IP addresses with the queries' results from multiple DNS servers; the other one is to correlate queries' results with multiple possible FF domains. They also proved that the detection speed had been speeded up because of those correlation schemes. To avoid single point of failure and improve the scalability, they developed a collaborative intrusion detection architecture LarSID to support the distributed correlation using a peer-to- peer publish-subscribe mechanism for sharing evidence. Their results show that their decentralized model was 16 to 10,000 times faster than previous centralized model [2] with the same correlation schemes [3].

3.2 Botnet Domain Flux Detection technique

Ma et al. applied a supervised machine learning to detect and prevent users from visiting malicious websites based on automated URL classification statistically [4]. It is a lightweight model which investigated the lexical features and host-based properties of malicious domain URLs. The lexical features which they selected are the length of entire URL, and the number of dots in the URL with of a bag-of-words representation. For host-based features, they extracted IP ad- dress properties, WHOIS (registrars) properties, domain name properties (TTL, etc.), and geographic properties (physical location, link speed, etc.). To train and evaluate their features, they applied three classification models - Naive Bayes, Support Vector Machine (SVM), and Logistic Regression on data sets from four different sources (two malicious, two benign ones). They found that WHOIS and lexical features can provide rich information and the combination of all features to form a full feature set can reach the highest accuracy. To compare full feature performance with traditional blacklist method, Ma et al. used a ROC graph to explain how full feature made difference. At last, they show how their classifiers selected automatically from large amount of features and determined the most predictive ones and achieved a modest false positive rate - 95% to 99% [4].

3.3Analysis of network traffic

The network traffic monitoring and analysis approach is useful to identify the existence of botnet in the networks. A collection of the signatures and behaviours of existing botnets was made, to build a common botnet model, which is independent of botnet protocol and

structure. This model can be used to detect botnets. With this model, botnets based on several network traffic anomalies such as high network latency, high volumes of traffic, traffic on unusual ports can be monitored. Even some unusual system behavior that could indicate presence of malicious bots in the network may also be detected. With the common botnet model, defenders can identify the hosts that share both similar communication patterns and similar malicious activity patterns. Although, this technique is powerful for detection of known botnets, it's not effective as we think, in detecting new botnets, especially in monitoring encrypted C&C traffic.

3.4 Host based Detection

Host based detection is simple but effective technology .signature based malware detection and behaviour based detection both fall into this category. signature based malware detection is effective and still widely used. But botmasters may deploy polymorphic technology to defend against signature detection. Luckily, the bots behaviour wont change much. Security defenders could analyze the behaviour of sample bot, such as registry logging, secondary downloading, and service registering, etc, combined with other information gained from reverse engineering, security defenders could make a custom detection toolkit.

4.Botnet Defense

Most of the common defensive techniques, such as firewalls, IDS, or antivirus solutions, act on a local level. The locality is the problem when multiple targets are attacked that are managed by different entities, Two major types of countermeasures are considered. The first is classical countermeasures, which are rather moderate in their implications, but are very limited because of their dependence on the cooperation of other organizations. The second type is more aggressive countermeasures with global consequences which can be conducted by a single organization and are therefore, more suitable for tactical takedown.

4.1 classical countermeasures

(A) C&C Server Takedown If the location of a C&C server has been determined, it can theoretically be shut down or disconnected. This can be made difficult if redundant infrastructures spread multiple instances of the server all over the world, in particular hosting them with different providers. In addition to the main C&C endpoint(s), backup channels have to be identified, if the takedown is to be sustainable.

Legal constraints in some countries prohibit or complicate the takedown of C&C servers, enabling so-called bulletproof hosting requiring law enforcement intervention. In some

countries, authorities and ISPs are reluctant to cooperate with security researchers or other security authorities. This is well-known and taken advantage of by botnet operators.

(B) DNS-based Countermeasures If the C&C infrastructure of the botnet is based on DNS, then a classical countermeasure is deregistration of those domains required by the botnet. This has to be done in cooperation with the respective DNS registrars and was successful in several cases. A requirement for this countermeasure to be sustainable is that the botnet's C&C infrastructure relies solely on DNS mechanisms.

4.2 Proactive Countermeasures

Beside the classical countermeasure, there are also more effective proactive countermeasures.

(A) Response DDoS If the locations of the C&C endpoints are known, a possibility is to launch a counter-DDoS attack to disable these endpoints. This is only possible if there is a single or limit number of C&C servers and would not work in a botnet relying on P2P infrastructure. A requirement for this is the availability of one or more machines for creating the traffic.

The application of a counter-DDoS is possible practically instantly as soon as information about the C&C endpoints is available. However, the sustainability is negligible. The attacked botnet is disabled only as long as the counter-DDoS is executed.

(B)Hack-back Another proactive countermeasure is hacking back, i.e. penetrating the C&C server and taking down the botnet from within. This requires the existence of a flaw in the C&C infrastructure which needs to be found and exploited. A team of highly skilled penetration specialists needs to be involved. In open-source botnets, the C&C protocol can be easily discovered by analyzing the source code. Standard source code auditing tools can be used to find weaknesses in the code. Construction kit botnets are usually sold together with the C&C server, although it is typically in binary form. Therefore, reverse engineering and binary code auditing skills are required. For specialized botnets it can be very difficult to obtain information about the C&C server. It is sometimes possible when using standard components with known vulnerabilities, e.g. specific Web servers. In all cases, analysts are required who are able to think outside box and identify non-obvious relationships between botnet components. Kit-based and specialized botnets require the highest reverse engineering skills. The time required for such a hack-back differs among the different botnet classes. Because of the multitude of available code analysis tools, open-source botnets can often be hacked in a matter of minutes if a sufficient number of vulnerabilities exists, otherwise it is a matter of days depending on the complexity of the code. More time is required for kit-based botnets, since reverse engineering is needed most of the time. Because the server binary is

available, offline and local stress tests can be performed. A minimum of several days can be expected, although the required time is more likely along the order of magnitude of weeks. Hacking of specialized botnets is very difficult. First the protocol has to be reverse engineered and possible weaknesses need to be derived. At least several weeks are needed for this. Once access to the C&C server has been gained, diverse valuable information can be discovered. The installation of a root kit allows the complete control of the server machine and might even result in greater privileges than even the botherders have. However, in most countries it is illegal to gain access to computer systems of others without their knowledge. From an ethical point of view, hacking back is effectively fighting fire with fire.

(D) Infiltration/Manipulation Another proactive countermeasure is the infiltration of a botnet which might lead to the botnet being manipulated and/or disabled from within. This requires an in- depth understanding of the botnet's architecture and C&C protocol. Cross-domain expertise is needed to identify and exploit weaknesses in the C&C protocol or architectures. Related fields in this case are communication protocol design, structured auditing as well as cryptography.

The sustainability of botnet infiltration is typically very high, provided it is not pursued too aggressively. For example, the aggressive monitoring of Storm by researchers was obvious to the botnet operators. If manipulations are made on the C&C server, they can be detected most of the time. To be truly effective, sudden strikes are essential.

(E) BGP Black holing Another possibility is the redirecting of botnet-related traffic, so-called sink holing. The redirected traffic can simply be discarded or analyzed further to gather more information about infected machines. Resources with regard to money, skills and cross-domain knowledge are similar to those of regular C&C server takedowns. The processes can mostly be fully automated. However, the existence of backup channels for C&C processes can be challenging. Once sufficient information about the botnet and its structures is available, the C&C endpoints can be inserted into BGP feeds within a few seconds, although their propagation can take several minutes.

4.3 Honeypot-Based Monitoring

Botnet monitoring helps defenders understand its working patterns, find design vulnerability, and spy on plain command text. honeypot monitoring has been widely used in monitoring botnet activities. Defenders could configure a honeypot to serve as a servant and join the botnet. Because our proposed botnet can delete inactive servants from its peer-list, the honey pot should act as a real servant in order to monitor the botnet as long as possible. in this way, defenders have opportunities to obtain the plain text of commands.

4.4 Defense against Domain flux

Domain flux technology has three important features 1)bots send out a lot of Dns resolution requests to a DNS server in a short time 2)most requested DNS have common substrings; and 3)most requested DNS will receive empty A-records because the requested DNS has not registered .Considering these three features, we can detect Domain flux network flow at the gateway of LAN. First, collect DNS request information including the requested DNS, timestamp, end the source IP address from where the request is sent out. Also, the response from the DNS server should be recorded, including the request DNS, destination IP address and responding A-record. Combining the above results, we will know exactly which ip address send out a DNS request at a specific time and the response is also known. In a time window, calculate the longest common substring of all the requested DNS sent out in the while. If the number of requested DNS which have the longest common substring and have an empty A-record exceeds threshold representing a normal web application. We can determine that the host is in domain flux programs. Subsequent DNS requests containing the lognest common substring could be blocked to cut off the communication between bots and command server.

4.5 The index Poisoning

index poisoning attack is done by inserting massive numbers of bogus records into the index of P2P file sharing system[5]. It is used to prevent illegal distribution of copyrighted content in P2P networks. The same technique can also be used to mitigate P2P botnets, because most P2P botnets make use of the indexes in P2P networks to implement their C&C mechanism, such as Stormnet and Peacomm botnet. With the help of honeypots and reverse engineering techniques, defenders are able to analyze behaviors of bot programs and find out the index which is related to the command of botmaster. Because of the limited of index that is used for command distribution, in some sense, the C&C architecture of P2P botnets is similar to that of the traditional centralized botnets. Index-based P2P botnets logically rely on predefined indexes, just like traditional botnets physically rely on central points or communication. And in most of the P2P networks, there is no central authority to manage the index. Therefore, any node no matter benign or malicious is able to insert records into the index. And in the past, there is no algorithm to authenticate the identity of the node and content of the records. That is why P2P botnets are vulnerable to index poisoning attack.

4.6 Fluxing Mitigation

Automatic identify FF that recorded in domain blacklist. Blacklist can be used to stop FF by collaboration from domain name registrar. registrar had authority to shut down a domain this effectively taking down scam. Automatic black list of FF domain can quickly notify registrar about fraudulent domains.ISP can use such a blacklist to protect its cline from FFSN, blacklist derived from DSNBL. Domain blacklist can be used for spam filtering.

5. Conclusions

As a cyber crime platform, botnet is one of the biggest network security threats. In this paper , we discussed all the three phases in lifetime of P2P botnets, and possible direction for P2P botnet detection and mitigation were pointed out. These can help us with depth understanding of details of botnets, and also guide us in the botnets Defense research.

REFERENCES

[1]T. Holz**,** C. Georecki, K. Rieck, and F.C. Freiling, "Measuring and detecting fast flux service networks." In NDSS, 2008.

[2] C. V. Zhou, C. Leckie, S. Karunasekera, and T. Peng, "A Self-healing, Self-protecting, Collaborative Intrusion Detection Architecture to Trace- back Fast-flux Phishing Domains," in Proceedings of the 2nd IEEE Workshop on Autonomic Communication and Network Management, Apr. 2008.

[3] C. V. Zhou, C. Leckie, and S. Karunasekera, "Collaborative detection of fast flux phishing domains." JNW, vol. 4, no. 1, pp. 75–84, 2009.

[4] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: learning to detect malicious web sites from suspicious urls," in Proceed- ings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2009, pp. 1245–1254.

[5] J. Liang, N. Naoumov, and K. W. Ross, "The index poisoning attack in p2p le sharing systems," in Proc. of the IEEE INFOCOM, April 2006.

[6]Areej Al-Bataineh, "Detection and Prevention Methods of Botnet-generated Spam," Universtiy of Texas

[7]C.Czosseck, E. Tyugu, T. wingfield, "On the arms race around botnets-setting up and taking down botnets,"2011 3rd international conference on cyber conflict.

[8]Zhiqi Zhang, Baochen Lu, Peng Liao, Chaoge Liu, Xiang Cui, ”A hierarchical hybrid structure for botnet control and command,” Research center of information security, institute of computing technology , Chinese academy of sciences.

[9]Lei Zhang, Shui Yu, Di Wu, Paul Watters, ”A survey on latest botnet attack and defense,” 2011 international joint conference of IEEE.