RESEARCH INVENTY: International Journal of Engineering and Science
ISBN: 2319-6483, ISSN: 2278-4721, Vol. 1, Issue 12 (December 2012), PP 25-29
www.researchinventy.com

Conficker Botnet Prevention: Crypto-Pan Algorithm

S.Aanjan Kumar
Department of Software Engineering
Mountzion College of Engg and tech
Mobile: 9786501012

Abstract — *This paper measure the potential power of Conficker to estimate its effects on the networks/hosts when it performs malicious operations. Isolating the BotNet traffic from regular traffic makes sharing of the data possible. It uses Address anonymization can be done by Cryptography-based Prefix preserving Anonymization algorithm (Crypto-Pan) to control the attack of Conficker botnet. The above method for anonymizing has many benefits. Crypto-PAn is a cryptography-based sanitization algorithm for network trace owners to anonymize the IP addresses of Conficker botnet and their traces in a prefix-preserving manner.*

Index Terms—*Bot , Botnet, Conficker, Crypto-Pan, Botnet defense*.

I. Introduction:

Botnet is a collection of software agents, or robots, automatically. The term is most commonly associated with malicious software, but it can also refer to a network of computers using distributed computing software. Botnets have become a significant part of the Internet, albeit increasingly hidden[1]. Due to most conventional IRC networks taking measures and blocking access to previously-hosted botnets, controllers must now find their own servers.

Recent days Conficker is the most recent widespread, well known worm/bot. According to several reports it has infected about 7 million to 15 million hosts and the victims are still increasing even now[1]. This analyze Conficker infections at a large scale, about 25 millions victims, and study various interesting aspects about this state-of-the-art malware. The observe that Conficker has some very different victim distribution patterns compared to many previous generation worms/botnets, suggesting that new malware spreading models and defense strategies are likely needed.

In this paper the technique and strategy to find the conficket botnet attack and prevent from its actions.

1.1. Conficker:

Conficker Botnet first appeared in November 2008 and rapidly spread in the world within a short period. It exploits a Net BIOS vulnerability in various Windows operating systems and utilizes many new[1], advanced techniques such as a domain generation algorithm, self-defense mechanisms, updating via Web and P2P, and efficient local propagation.

As a result, it has infected millions of victims in the world and the number is still increasing even now It is clear that the complex nature of Conficker makes it one of the state-of-the-art botnets, and therefore the analysis of Conficker is very important in order to defend against it. A full understanding of Conficker can also help us comprehend current and future malware trends.

However, for a worm/bot that has infected so many victims and has so much potential to damage the Internet, it deserves a much deeper study[2]. Such study is necessary because by analyzing this state of-the-art botnet, e.g., how it differs from previous generation malware and whether such differences represent future trends or not. These deeper investigations could also provide new insights in developing new detection and defense mechanisms for current and future malware.

In table 1.1 shows the conficker botnet ip based attacks more than other botnet[2].

Comp. type	#Comps.	#domains	#IPs
Conficker botnet	1	1.9K	19
Helldark botnet	1	28	5
Mjuyh botnet	1	121	1.2K
Misspelt Domains	5	215	17
Domain Parking	15	630	15
Adult content	4	349	13

Table 1.1: Conficker botnet attacked IP

II. How powerful is conficker botnet

The previous study shows Conficker really fulfills attacks or not, it is possible that a botmaster of Conficker commands victims to carry\ out some malicious actions[3]. Thus, it may be interesting to understand how Conficker victims affect networks or hosts when they perform attacks.

This paper shows power of botnets can be defined by each attack method they provide. For example, if bots generate malicious traffic such as DDoS packets[4], the performance of this attack will be decided by the amount of traffic they can generate.

III. Conficker Botnet :Evolution

This is a natural strategy for botmasters to design a peer-to-peer (P2P) control mechanism into their conficker botnets [5]. In the last several years, conficker botnets such as Slapper , Sinit , Phatbot and Nugache have implemented different kinds of P2P control architectures. In the fig 3.1 shows the evolution of conficker botnet to Hybrid P2P architecture change its node every time change its victims.

3.1. Hybrid P2P conficker Botnet

The design of an advanced conficker botnet, from our understanding, should consider the following practical challenges faced by botmasters: In addition, the design should also consider many network related issues such as

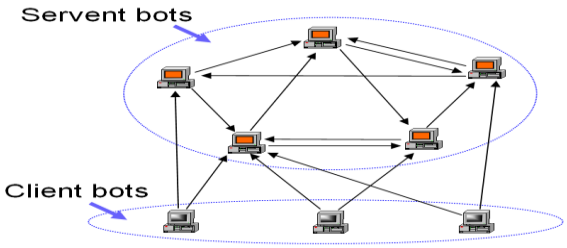


Fig.3.1. Hybrid P2P conficker botnet

dynamic or private IP addresses and the diurnal online/offline property of bots . By considering all the challenges listed above, In this paper to present our research on the possible design of an advanced hybrid P2P conficker botnet[6]. The proposed hybrid P2P conficker botnet has the following features:

1.The Conficker botnet requires no bootstrap procedure.

2. The Conficker botnet communicates via the peer list contained in each bot.

IV. Action performed by conficker botnet:

The Conficker botnets can consist of several ten thousand compromised machines – Conficker botnets pose serious threats. Conficker Botnets can be utilized for many notorious purposes by their masters – for Distributed Denial-of-Service (DDoS), spamming, etc[7]. Since botnets pose such a powerful threat, a combined effort is needed in the research community to develop mechanisms to counter the threat.

4.1. DDoS. A denial-of-service attack (also, DoS attack) is an attack on a computer system or network that causes a loss of service to users, typically the loss of network connectivity and services by consuming the bandwidth of the victim network or overloading the computational resources of the victim system[8].

4.2. Spamming. With the help of a Conficker botnet and thousands of bots, an attacker is able to send massive amounts of bulk email (spam). Some bots also implement a special function to harvest email-addresses [9].

4.3. Click fraud. Conficker Botnets can also be used to gain financial advantages by setting up a fake website with some advertisements. The hosting companies are charged for the number of clicks on the ads.

4.4. Identity theft. Large scale identity theft is one of the fastest growing crimes on the Internet. Conficker Botnets, with an army of thousands of bots, can use packet sniffers to watch for interesting clear-text data passing by a compromised machine and thus can retrieve sensitive information like usernames and passwords[9].

V. Proposed Crypto-Pan:

Conficker botnet controlled using a cryptography-based, prefix-preserving anonymization technique. it allows multiple traces to be sanitized in a consistent way, over time and across locations. That is, the same IP address in different traces is anonymized to the same address, even though the traces might be sanitized separately at different time and/or at different locations[10].

The proposed Crypto-PAn is the software tool based on this technique. To sanitize traces, trace owners provide Crypto-PAn a secret key. Real IP addresses are deterministically mapped to anonymized addresses based on this secret key. This ensures anonymization consistency across multiple traces. The construction of Crypto-PAn preserves the secrecy of the key and the (pseudo)randomness of the mapping from an original IP address to its anonymized counterpart. The security properties inherent in all prefix preserving IP address anonymization schemes. Their interesting observations can be summarized as follows:

The damage caused by an attack is very much specific to the kind of trace. This means that no general statement can be made regarding the safety of releasing traces and each case should be evaluated individually. Compromising addresses at random is a good means of attacking prefix-preserving anonymization. The authors also claim that their scheme is secure to the maximum level possible for a prefix-preserving scheme[11].

5.1 Extension of Crypto-PAn :cryptography method

Crypto-PAn does not create tables but instead mappings are uniquely determined by a passphrase and the symmetric block cipher[12] used by Crypto-PAn. In this paper we uses two advanace cryptography method:

5.1.1. ElGamal key generation:

ElGamal encryption in Crypto-Pan consists of three components: the key generator, the encryption algorithm, and the decryption algorithm. The ElGamal cryptosystem is usually used in a hybrid cryptosystem. I.e., the message itself is encrypted using a symmetric cryptosystem and ElGamal is then used to encrypt the key used for the symmetric cryptosystem[13].

5.1.2. Rijndael Key Generation

Here created a routine which takes as input a passphrase and creates a 32 byte key. We wanted to reuse as much code as possible, so we did not want to want to add a library and use a hash function. So it used Rijndael, which is already used by the anonymizer, to create a specialized hash function[13]. This method uses intermediate key to CBC encrypts the original buffer once more. The last 32 bytes of this encrypted 256 byte buffer now forms the final key. This is irreversible even if we reveal the other 244 bytes of the buffer because the attacker who knows this final key, does *not* know the intermediate key which depends upon the input. And without this key, the encryption operation cannot be reversed.

Then Crypto-Pan extend to provide an advantage :

1. The damage caused by an attack is very much specific to the kind of trace. This means that no general statement can be made regarding the safety of releasing traces and each case should be evaluated individually.
2. Compromising addresses at random is a good means of attacking prefix-preserving anonymization[14].

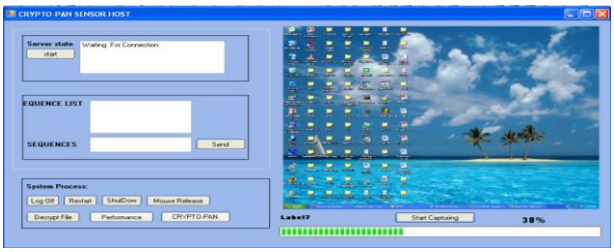
Thus the above stated methods uses Crypto-PAn anonymizer class is a key generator. It takes a key in hex as input, generate keys. Here integrated a key generation algorithm into the software which we then applied to Conficker botnet traces. Here setup the software to work on the binary data of the Conficker botnet logs directly. They could have simply compiled the code with few changes to take a dotted decimal IP address However, this is somewhat inefficient as the Crypto-PAn code works on IP addresses in a binary form. This code depends upon the IP address being at a specific place in the log entry[15].

VI. Control of Conficker using Crypto-Pan: Test Results

In this paper, studied of large-scale Conficker infection data to discover (i) their distribution over networks(ii) difference from previous bots/worms (iii) the effectiveness of current reputation-based malware detection/warning systems, and (iv) some insight to help detect future malware.

Our analysis of Conficker victims and cross-comparison results allowed us to obtain profound insights of Conficker victims[16]. They also guide us to understand the trends of malware infections and to find interesting ideas that can aid the design of future malware detecting systems. Here revealed that current reputation-based malware detecting systems that depend on previously known information are not enough to detect most Conficker victims.

In figure 1.2 shows the test application with sensor host with inbuilt crypto-pan algorithm in user system.



Inbuilt Crypto-Pan Algorithm

Fig.6.1 Test application with inbuilt crypto-pan algorithm to control conficker attack

This result suggests that different kinds of (complementary) detection systems such as an anomaly-based detection system are needed. The measure the performance of Conficker and to believe that it provides more clear understanding the real effects of Conficker on the network or host. Here provide a basis that proves the hypothesis of "A Conficker bot is more likely to infect nearby hosts than randomly chosen hosts" and believe that it calls for more research of detection systems which are based on watching/sharing/correlating neighborhood information.

In this section provides how crypto-pan algorithm prevents the conficker botnet attacks can be stated in graph result figure 1.3 states the attacks lower by crypto-pan regards at time.

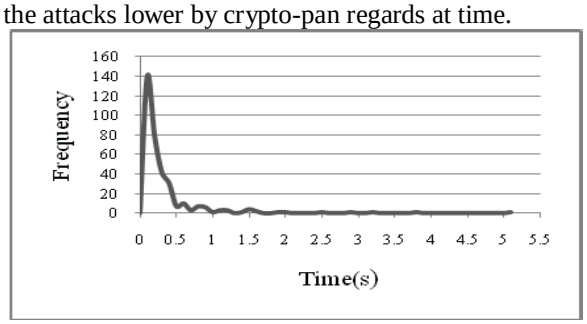


Fig.6.1. Test graph result for crypto-pan prevent conficker in time

However, like most network logs, Crypto-PAn are abundant with sensitive information, the most sensitive fields being incoming and outgoing IP addresses. As such we have analyzed different methods of anonymizing IP addresses. In this paper we discuss four basic types and focus on tools that provide prefix-preserving pseudonymization. Here found Crypto-PAn to be the most superior due to its ability to be parallelized. However, the source code provided a key generator. Here developed a novel key generator that required no additional cryptographic libraries to extend the Crypto-PAn tool. Using this extended tool we created a Crypto-PAn anonymizer that operates on binary IP traces. This have shown that a typical workstation can easily handle the anonymization to Control Conficker Botnet.

VII. Conclusion and future work:

The purpose of this project was to gain a first hand knowledge of the Conficker botnet command and control and how Conficker botnets spread in the network. Our work involved two phases. In the first phase, we studied some of the techniques used for Conficker botnet detection. Some of the emerging trends in Conficker botnet usage were also studied. In the second phase, This investigated some algorithm that can be used Crypto-Pan to anonymize the Conficker botnet data.

In order to there is still exploring the different tools that can be used for anonymizing the Conficker botnet data. There are some anonymization tools available, but the tools have to be customized to work for the Conficker botnet data. As part of future work, Here intend to extend our knowledge of botnets to studying click-fraud attacks. This would also explore bot-nets that use high power P2P systems for communication and hence are difficult to detect.

References

[1] S. Shin and G. Gu. "Conficker and Beyond". In Proceedings of 2010 Annual Computer Security Applications Conference (ACSAC), Dec. 2010.

[2] SRI-International. An analysis of Conficker C. http://mtc.sri.com/ Conficker/addendumC/.

[3] B. Stock, M. E. Jan Goebel, F. C. Freiling, and T. Holz. Walowdac Analysis of a Peer-to-Peer Botnet. In Proceedings of European Conference on Computer Network Defense (EC2ND), Nov. 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 Proceedings of ACM CCS, Nov. 2009.

[5] M. S. Techcenter. Conficker worm. http://technet.microsoft.com/en-us/ security/dd452420.aspx.

[6] Verisign. The Domain Name Industry Brief. http://www.verisign.com/domain-name-services/domain-information-center/domain-name-resources/domain-name-report-sept09.pdf.

[7] D. Watson. Know Your Enemy: Containing Conficker.http://www.honeynet.org/papers/conficker.

[8] Y. Xie, F. Yu, K. Achan, E. Gillum, M. Goldzmidt, and T. Wobber. HowDynamic are IP Addresses? In Proceedings of ACM SIGCOMM, Aug.2007.

[9] Y. Xie, F. Yu, K. Achan, R. Panigraphy, G. Hulte, and I. Osipkov. Spamming Botnets: Signatures and Characteristics. In Proceedings of ACM SIGCOMM, Aug. 2008.

[10] J. Xu, J. Fan, M. H. Ammar, and S. B. Moon, "Prefix-Preserving IP Address Anonymization: Measurement-based Security Evaluation and a New Cryptography-based Scheme," in IEEE International Conference on Network Protocols (ICNP), 2002.

[11] R. Pang and V. Paxson, "A High-Level Programming Environment for Packet Trace Anonymization and Transformation," in ACM SIGCOMM, 2003.

[12] M. Peuhkuri, "A Method to Compress and Anonymize Packet Traces," in ACM SIGCOMM Internet Measurement Workshop(IMW), 2001.

[13] M. Allman, E. Blanton, and W.M. Eddy, "A Scalable System for Sharing Internet Measurements," Passive and Active Measurement (PAM) Workshop, 2002.

[14] C.J. Antonelli, M. Undy, and P. Honeyman, "The Packet Vault: Secure Storage of Network Data," USENIXWorkshop on Design Issues in Anonymity and Unobservability, 2000.

[15] G. Minshall, "TCPdpriv: Program for Eliminating Confidential Information from Traces," Ipsilon Networks, Inc.

[16] G. Kuenning and E. Miller, "Anonymization Techniques for URLs and Filenames," Technical Report UCSC-CRL-03-05,University of California, Santa Cruz, Sep. 2003.