ISSN: 0975-833X

Available online at http://www.journalcra.com

International Journal of Current Research
Vol. 3, Issue, 4, pp.046-049, April, 2011

INTERNATIONAL JOURNAL OF CURRENT RESEARCH

RESEARCH ARTICLE

DETECTING AND MONITORING THE BOTNET USING HONEYPOT

*** Ahilandeswari, C. and Palanisamy, V.**

Department of Computer Science and Engineering, Alagappa University , Karaikudi , TN, India.

ARTICLE INFO

Article History:
Received 10th January, 2011
Received in revised form
14th February, 2011
Accepted 11th March, 2011
Published online 17th April 2011

Key word:
Botnet, Honeypot
Shut down, Monitored, Hijacked

ABSTRACT

A "botnet" consists of a network of compromised computers controlled by an attacker. Here we present the design of peer-to-peer botnet with honeypot. Compared with current botnets, the proposed botnet is harder to be shut down, monitored, and hijacked. It provides robust connectivity, individualized encryption and control traffic dispersion, limited botnet exposure by each bot, and easy monitoring and recovery by its botmaster. The honeypot used here is designed in such a way that it doesn't enable the attackers to track the data during the communication process.

INTRODUCTION

A "botnet" consists of a network of compromised computers ("bots") connected to the Internet that is controlled by a remote attacker ("botmaster").These C&C servers receive commands from their botmaster and forward them to the other bots in the network. Fig. 1 shows the basic control communication architecture for a typical C&C botnet.Most of the current research has focused upon the C&Cbotnets.From a botmaster's perspective, the C&C servers are the fundamental weak points in current botnet architectures. First, a botmaster will lose control of it's botnet once the limited number of C&C servers are shut down by defenders. Second, defenders could easily obtain the identities of all C&C servers based on their service traffic to a large number of bots, or simply from one single captured bot (which contains the list of C&C servers). Third, an entire botnet may be exposed once a C&C server in the botnet is hijacked or captured by defenders. Current P2P Botnets and their Weaknesses Considering the above weaknesses inherent to the centralized architecture of current C&C botnets, it is a natural strategy for botmasters to design a peer-to-peer (P2P) control mechanism into their botnets. Botnets such as Slapper, Sinit, Phatbot, and Nugache have implemented different kinds of P2P control architectures. Sinit uses public key cryptography for update authentication. Ugache attempts to thwart detection by implementing an encrypted / obsfucated control channel. Nevertheless, simply migrating available P2P protocols will not generate a sound botnet, and the P2P designs used by several botnets .

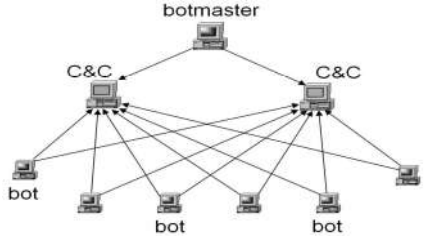


Fig. 1. C&C architecture of a C&C botnet.

Proposed Peer to Peer Botnet. To generate a robust botnet capable of maintaining control of its remaining bots even after a substantial portion of the botnet population has been removed by defenders.To prevent significant exposure of the network topology when some bots are captured by defenders. To easily monitor and obtain the complete information of a botnet by its botmaster. To prevent defenders from detecting bots via their communication traffic patterns. Design of an hybrid botnet with honeypot The botnet communicates via the peer list contained in each bot. A botmaster could easily monitor the entire botnet by issuing a report command. This command instructs all (or partial) bots to report to a compromised machine (which is called a sensor host) that is controlled by the botmaster. The IP address of the sensor host, which is specified in the report command, will change every time a report command is issued to prevent defenders from capturing or blocking the sensor host beforehand.

RELATED WORK

Arce and Levy presented a good analysis of how the Slapper worm built its P2P botnet. Barford and Yegneswaran gave a detailed and systematic dissection of many well-known botnets that have appeared. botnets is mainly focused on monitoring and detection. The comprehensive studies on using honeypots to join botnets in order to monitor botnet activities in the Internet. Bots in the first group are called servent bots since they behave as both clients and servers. The second group contains the remaining bots, including bots with dynamically allocated IP addresses, bots with private IP addresses, bots behind firewalls such that they cannot be connected to the global internet. Only servent bots are candidates in peer lists. All bots, including both client bots and servent bots, actively contact the servent bots in their peer lists to retrieve commands. Deploying honeypots To learn how intruders probe and attempt to gain access to your systems and gain insight into attack methodologies to better protect real production systems.

Corresponding author: ahi.cm2010@gmail.com

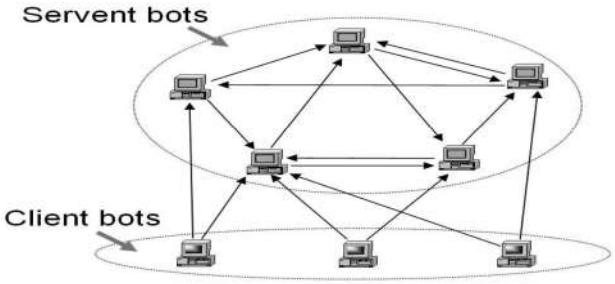


Fig. 2. C&C architecture of the proposed hybrid P2P botnet.

To gather forensic information required to aid in the apprehension or prosecution of intruders.Honeypots came in two flavors Low-interaction and High-interaction. Interaction measures the amount of activity that an intruder may have with honeypot. In addition, honeypots can be used to combat spam. Relationship between Traditional C&C Botnets and the Proposed Botnet Compared to an extension of a C&C botnet. The number of C&C servers (servent bots) is greatly enlarged, and they interconnect with each other. Indeed, the large number of servent bots is the primary reason why the proposed hybrid P2P botnet is very hard to be shut down.

BOTNET COMMAND AND CONTROL

The essential component of a botnet is its C&C communication. Compared to a C&C botnet,The major design challenge is to generate a botnet that is difficult to be shut down, or monitored by defenders or other attackers.
Command Authentication A botmaster generates a pair of public/private keys, (K^+,K^-), and hard codes the public key K^+ into the bot program before releasing and building the botnet. There is no need for key distribution because the public key is hard-coded in bot program. Later, the command messages sent from the botmaster could be digitally signed by the private key K^- to ensure their authentication and integrity. This public-key-based authentication could also be readily deployed by current C&C botnets. Individualized Encryption Key A botmaster may also wish to encrypt her command messages to prevent being eavesdropped by defenders or other attackers. Suppose the peer list on bot A is denoted by L_A. It will not only contain the IP addresses of M servent bots, but also the symmetric keys used by these servent bots. Thus, the peer list on bot A is

$L_A=\{(IP_{i1},K_{i1}),(IP_{i2},K_{i2}).(IP_{iM},K_{iM})\}$, where (IP_{ij},K_{ij}) are the IP address and symmetric key used by servent bot i_j. Individualized Service Port The peer-list-based architecture also enables the proposed botnet to disperse its communication traffic in terms of service port. Since a servent bot needs to accept connections from other bots, it must run a server process listening on a service port. The service port number on servent bot i, denoted by Pi, could be picked by the bot, either randomly or selectively. $LA=\{IP_{i1},K_{i1}, P_{il}), \ldots, (IP_{iM},K_{iM}, P_{iM})\}$ (2)With the new peer list L_A shown above, bot Dispersed network traffic.

BOTNET MONITORING BY ITS BOTMASTER

Another major challenge in botnet design is making sure that a botnet is difficult to monitor by defenders, but at the same time, easily monitored by its botmaster. Monitoring via a Dynamically Changeable Sensor To monitor the proposed hybrid P2P botnet, a botmaster issues a special command, called a report command, to the botnet, thereby instructing every bot to send its information to a specified machine that is compromised and controlled by the botmaster.. Every round of report command issued by a botmaster could potentially utilize a different sensor host. This would prevent defenders from knowing the identity of the sensor host before seeing the actual report command. The sensor is chosen such that it normally provides such a service to avoid exhibiting abnormal network traffic. Use several sensor machines instead of a single sensor. Manually verify the selected sensor machines are not honeypots Wipe out the hard drive on a sensor host immediately after retrieving the report data. .Additional Monitoring Information A botmaster not only wants to know a botnet size and topology in order to conduct efficient attacks. IP Address Type Internet computers are identified by their IP addresses.

BOTNET CONSTRUCTION

Its network connectivity is solely determined by the peer list in each bot. Botnets utilize many different infection mechanisms, such as vulnerability exploitation, e-mail viruses, traditional file-based viruses, network share, etc. Basic Construction Procedure A natural way to build peer lists is to construct them as a botnet propagates. Suppose the size of peer list in each bot is configured to be M. As a bot program propagates, the peer list in each bot is constructed according to the following procedure: New infection. Bot A passes its peer list to a vulnerable host B when compromising it. If A is a servent bot, B adds A into its peer list (by randomly replacing one entry if its peer list is full). If A knows that B is a servent bot A adds B into its peer list in the same way.Reinfection If reinfection is possible and bot A reinfects bot B, bot B will then replace R ($R \leq M-1$) randomly selected bots in its peer list with R bots from the peer list provided by A. Again, bots A and B will add each other into their respective peer lists if the other one is a servent bot as explained in the above "new infection" procedure. In the reinfection procedure, a bot does not provide its peer list to those who reinfect it. This is important, because, if not, defenders could recursively infect (and monitor) all servent bots in a botnet based on a captured bot in their honeypot in the following way Fig. 3a shows the degree distribution for servent bots (client bots always have a degreeM, equal to the size of peer list) after the botnet has accumulated 20,000 members. Because the botnet stops growing when it reaches the size of 20,000, the reinfection events rarely happen (only around 600). For this reason, connections to servent bots are extremely unbalanced: more than 80 percent (4,000) of servent bots have degrees less than 30, while each of the 21 initial servent bots have a degree between 14,000 and 17,500 (the last tiny bar at the bottom right corner of the figure close to x-axis value of 10 represents these 21 servent bots). have been infected. Fig. 3b shows the degree distribution for servent bots in this scenario.

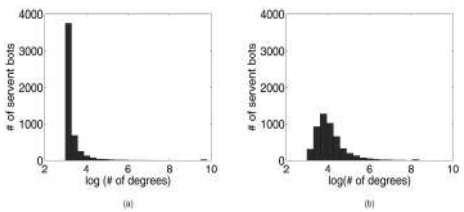


Fig. 3. Servent bot degree distribution (construct botnet via "new infection" and "reinfection" procedure only).

(a) Vulnerable population 500,000.

(b) Vulnerable population 20,000.

When the botnet stops infection process, overall around 210,000 reinfection events happened. Fig. 4. Servent bot degree distribution (constructed via infection and peer-list updating).Botnet Command Initiation Comparing Fig. 2 with Fig. 1, we can see that the proposed P2P botnet does not show how its botmaster contacts the botnet to issue commands. As used in the simulation experiment, a botnet propagates based on a set of initial servent bots. The botmaster sets their service ports beforehand and thus knows their service ports. After the botnet is released, the botmaster could inject commands through these initial servent bots. After the botmaster issues a report command and gets the first report with the information of service ports of all current servent bots, the botmaster can inject commands through an arbitrarily chosen set of servent bots.

BOTNET ROBUSTNESS STUDY

The hybrid botnet has two factors affect the connectivity of a botnet: 1) some bots are removed by defenders and 2) some bots are offline (for example, due to the diurnal phenomenon .These two factors, even though completely different, have the same impact on botnet connectivity when the botnet is used by its botmaster at a specific time. For this reason, we do not distinguish them in the following study.

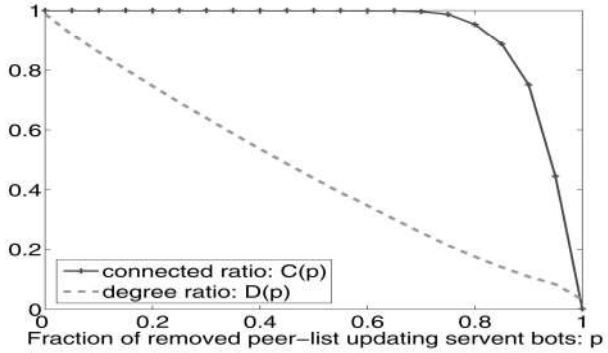


Fig. 4. Botnet robustness study.

Botnet Robustness Based on Two Metric Functions We present two metric functions to measure robustness. Let C(p) denote the connected ratio and D(p) denote the degree ratio after removing top p fraction of mostly connected bots among those peer-list updating servent bots. C(p) and D(p) are defined as C(p)= # of bots in the largest connected graph 3) # of remaining bots D(p)= Average degree of the largest connected graph Average degree of the original botnet. The botnet is the one shown in Fig. 4 that has a vulnerable population of 500,000 and runs the peer-list updating procedure only once when 1,000 servent bots are infected.This result shows the importance of the peer-list updating procedure. Once with different number of servent bots. Even if Fig. 5. Botnet robustness when the peer-list updating procedure runs once with different number of servent bots.Robustness Mathematical Analysis We provide a simple analytical study of the botnet robustness. Assume that each peer list contains M servent bots. It is hard to provide a formula when removing the top p fraction of mostly connected nodes.

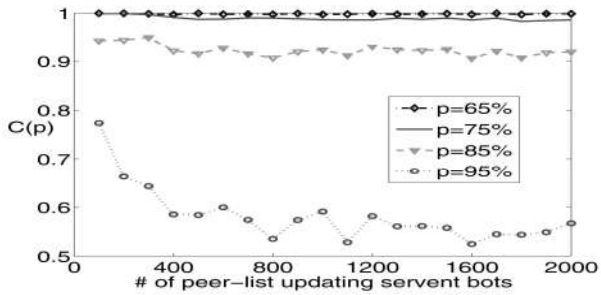


Fig.5. Botnet robustness when the peer-list updating procedure runs

However, we could provide the formula of C(p) when randomly removing p fraction of peer-list updating servent bots. As We simplify the analysis by assuming that each bot in the botnet connects only to peer-list updating servent bots. Then, when we consider removing a fraction of peer-list updating servent bots, more links will be removed compared to the original botnet network. Thus, the probability that a bot is disconnected is p^M. Therefore, any remaining bot has the same probability 1 _ p^M to stay connected

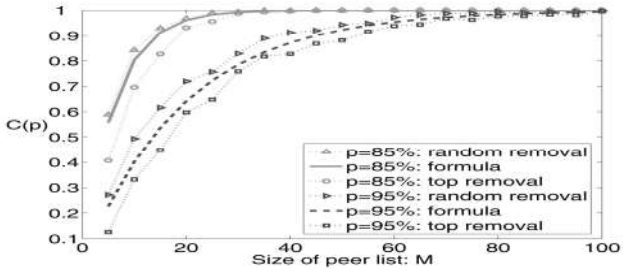


Fig. 6. Comparison of the analytical formula (5) and simulation results.

i.e., the mean value of C(p) is (in case of random removal) C(p) = 1 _ p^M.) Fig. 7 shows the analytical result from comparing with the simulation result C(p) of the random removal, and the simulation result C(p) of the removal of top p fraction of mostly connected peer-list updating servent bots.

Suppose a C&C-based botnet has R C&C servers. When defenders have the complete knowledge of such a botnet, they will always remove these R bots first. Thus, the botnet robustness metric C(p) is

C(p) ={1; <Rbots are removed;
0;>Rbots are removed; (6)

the botnet will be shut down if all R C&C server bots are removed, which makes it much less robust than the proposed P2P botnet.

DEFENSE AGAINST THE PROPOSED HYBRID P2P BOTNET

Botnet Monitoring Based on Honeypot Techniques Honeypot is an effective way to trap and spy on malware and malicious activities. Because compromised machines in a botnet need to cooperate and work together, it is particularly effective to use honeypot techniques in botnet. if a botnet cannot detect and get rid off honeypot bots. The third annihilation method introduced above relies on honeypot techniques. Botnet Monitoring Based on Spying Honeypots If a botnet cannot effectively detect honeypots, defenders could let their honeypots join botnets and monitor botnet activities. Once the commands is understood, defenders are able to 1) quickly find the sensor machines used by a botmaster in report commands

2) know the target in an attack command so that they could implement corresponding counter measures quickly right before the actual attack begins. First, defenders could let their honeypot bots claim to be servent bots in peer-list updating. By doing this, these honeypots will be connected by many bots in the botnet, and hence, defenders are able to monitor a large fraction of the botnet. Second, during peer-list updating, each honeypot bot could get a fresh peer list, which means the number of bots revealed to each honeypot could be doubled.

(a) Botnet growth and a honeypot monitoring as one of the initial servent bot. (b) A honeypot monitoring by joining before/after peer-list updating procedure. A remote code authentication3 cannot enable a botnet or its peer-listupdating sensor to detect these spying honeypots. Upon receiving a botnet report command, a honeypot could have its special program to send back a large amount of identities of fake bots. The information should not be sent from one single IP address. Instead, the honeypot should send out each fake bot's ID and host characteristics with different IP addresses. Simulation and Analysis of Botnet Monitoring by Multiple Honeypots It would be interesting to know how many honeypot defenders should set up in order to have an effective monitoring. By varying the number of honeypots joining a botnet before the peer-list updating procedure. Suppose the peer list size is M, the final botnet has I number of bots, and the number of servent bots used in peer-list updating procedure is K.

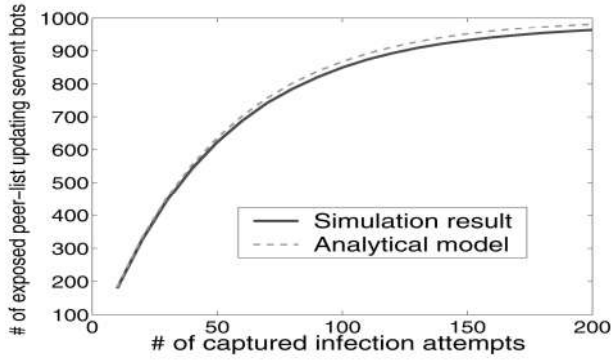
Thus the probability that the peer list in a specific bot contains none of those n honeypots is

$$\left(1-\frac{n}{K}\right)\left(1-\frac{n}{K-1}\right)\left(1-\frac{n}{K-2}\right)\cdots\left(1-\frac{n}{K-M}\right). \quad (7)$$

When K>M, which is the case in our simulations, (7) is approximately equal to

$$\left(1-\frac{n}{K}\right)^{M}. \quad (8)$$

which is the probability that this bot will not be exposed to honeypots.



If a bot infection is composed by several sequential components and a bot passes its peer list to a newly infected host.

CONCLUSION

It provides robust network connectivity, individualized encryption and control traffic dispersion, limited botnet exposure by each captured bot, and easy monitoring and recovery by its botmaster. Implementation of multiple honeypot have certain drawbacks. In order to deploy honeypots efficiently and avoid their exposure to botnets and botmasters, we design a honeypot which doesn't allow the attackers to enter into it.A new approach of it which would work better.

REFERENCES

Dependable and Secure Computing, IEEE Transactions on Issue Date: April-June 2010, Volume: 7 Issue:2 on pages(113-127).Date of current version: 18 May 2010, Sponsored by IEEE Computer Society.

Gu, G., Porras, P., Yegneswaran, V., Fong, M. and Lee, W. 2007. BotHunter: Detecting Malware Infection Through IDS-Driven Dialog Correlation," Proc. 16th USENIX Security Symp.(Security '07).

Information Assurance and Security (IAS), 2010 Sixth International Conference on issue date 23-25 Aug 2010 on pages 62-67 14 October 2010

Phatbot Trojan Analysis, http://www.lurhq.com/phatbot.html, 2008.

Servent, http://en.wikipedia.org/wiki/Servent, 2008.

Sinit P2P Trojan Analysis, http://www.lurhq.com/sinit.html, 2008.

University of Michigan Internet Motion Sensor, http://ims.eecs.umich.edu/, 2008.

Wikipedia:Binomial Distribution, http://en.wikipedia.org/wiki/Binomial_distribution, 2008.

Zou, C. and Cunningham, R. 2006. Honeypot-Aware Advanced Botnet Construction and Maintenance," Proc. Int'l Conf. Dependable Systems and Networks (DSN '06).

Bowden, M. 2010. The enemy with in http://www.theatlantic.com/magazine/archive/2010/06/the-enemy-within/8098/.

Gu, G., Perdisci, R., Zhang, J., Lee, W. et al., 2008. BotMiner: Clustering analysis of network traffic for protocol-and structure-independent botnet detection. In Proceedings of the 17th USENIX Security Symposium (Security08).

Gu, G., Zhang, J.and Lee, W. 2008. BotSniffer: Detecting botnet command and control channels in network traffic. In Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS08). Citeseer.

Holz, T., Steiner, M., Dahl, F., Biersack, E. and Freiling, F. 2008. Measurements and mitigation of peer-to-peer-based botnets: a case study on storm worm. In Proceedings of the 1st USENIX Workshop on Large-Scale Exploits and Emergent Threats, pages 1–9. USENIX Association.

Kola, M. 2008. Botnets: Overview and Case Study. PhD thesis, IBM Research.

Mazzariello, C. 2008. IRC traffic analysis for botnet detection. In Information Assurance and Security, 2008. ISIAS'08. Fourth International Conference on, pages 318–323.

Tangpong, A. and Kesidis, G. 2009.. A controlled environment for botnet traffic generation. http://www.cse.psu.edu/~tangpong/botnet/, April.